AN INTERLOCKING SPECIFICATION LANGUAGE

L.-H. Eriksson

Industrilogik L4i AB, Sweden

Maria Fahlén

The National Swedish Rail Administration (Banverket), Sweden

SYNOPSIS

Formal methods – the use of exact mathematical methods to describe and analyse systems – have reached a level of maturity where they can be used in practice to formally describe functional safety requirements on signalling systems as well as to verify the compliance of actual systems with these requirements.

Interest in formal methods is growing among suppliers and administrations, partly due to standardisation work within CENELEC. However, an important factor for the success of formal methods is the availability of generally accepted description languages, methods, and tools. In the work done so far, there has been little common ground apart from basic principles.

We propose the development of an Interlocking Specification Language (ISL). ISL is envisioned as a general purpose specification language augmented with a set of concepts and principles specifically designed to simplify the development of specifications of signalling systems.

We will give an overview of the role of formal methods in quality assurance of signalling systems and of how it will be affected by the ISL.

1. FORMAL METHODS

By *formal methods,* we understand the use of mathematically precise methods to describe and analyse systems that perform computations in a wide sense. Railway signalling systems – both traditional electromechanical and modern computer-based – are in this category. Formal methods are based on the language and reasoning methods of mathematics (typically the part of mathematics known as discrete mathematics and mathematical logic).

Traditionally, software engineering has not made much use of mathematics for design, as opposed to other engineering disciplines. The use of formal methods in the development of programmed systems is analogous to the use of mathematical models and design methods in other areas of engineering. In many cases, substantial quality improvement can be obtained by the introduction of formal methods.

Interest in this kind of methods and techniques has been growing during the past years, partly due to standardisation work within CENELEC. Several suppliers of signalling systems are studying how product development can benefit from the use of formal methods. Formal methods are also interesting as potential means to precisely and unambiguously describe the safety principles of a rail administration for the purpose of communicating such principles and for training.

Any use of formal methods must begin with the formal specification. A *formal specification* is a set of mathematical statements (formulae) that describes requirements on the behaviour of the system. The specifications are written in a *formal specification language*, a mathematical language where the meaning of statements is precisely defined. Since a formal language is precise and unambiguous, this method permits specifications to be written with a high degree of precision and without ambiguity.

Of course, these advantages do not come free. A prerequisite is that the formal language has the vocabulary necessary to express the requirements. This typically means that formal models must be available for the environment in which the system operates and for all concepts used.

It is usually desirable that the statements of the specification are on a high level of abstraction – both to be more easily understood, and to obtain independence from any particular system design. The formal model for high-level concepts can be very complex.

The properties that can be most naturally described using a formal specification are functional properties, e.g. requirements on the safe behaviour of a system. Properties relating to fault tolerance are also amenable to formal analysis, but this is more difficult and not much work has been done in practice.

The formal specification can be used for a number of purposes where a precise description is of value. Formal specifications have been used contractually, to unambiguously specify to a supplier what functional requirements a system should fulfil.

A formal specification can also serve as a reference for training or whenever answers are sought to questions of what the desirable behaviour of the system is in various situations. This use of a formal specification is of greatest value when the specification does not refer to a specific system, but rather to general principles for an entire class of systems.

As an example, the signalling rules of a railway administration could be expressed as a formal specification. Such a *generic* specification does not express requirements on any particular signalling system, but on any signalling system operating in accordance to the given rules. Together with a formal model of a rail yard, a generic specification can be used as a specification for a particular signalling system intended for that yard.

Simply the act of creating the formal specification can improve the quality of a set of requirements, as the process of writing a formal specification requires the consideration of the behaviour of the system in *all* possible situations. During development of a formal specification it is more often than not the case that the underlying traditional specification is found to be inadequate (incomplete or ambiguous).

The formal specification can also be used as the basis for a *formal verification* of a system. Formal verification is a process where the compliance (or non-compliance) of a system with its specification is demonstrated using formal reasoning. To formally verify a system, a formal model of the system is created. This model may have different scope or depth. It may model the system at the design level, at the implementation level or as appropriate for the analysis to be made. Once the model is available, a formal proof is attempted that the model complies with its specification.

If a proof can be found, it is known with certainty that the system model complies with the formal specification. Provided that the system model and the formal specification faithfully reflect the actual system and requirements, the result carries over to the actual system.

The main advantage of formal verification compared with testing is that *all* situations that can possibly occur will be considered by the proof – even if the number of cases is potentially infinite. Testing can only demonstrate compliance with a limited number of test cases. (On the other hand, testing has the advantage that it can be done on the actual implementation – not only on a model.)

The process of finding a proof is quite involved and in practice, some kind of computer assistance is needed. Preferably, the process should be fully automated. The state-of-the-art in computer proof procedures can fully automatically perform proofs of systems provided that the number of possible situations is finite – even if it is very large. This is the case with railway signalling systems. (See Stålmarck and Säflund (4) or Groote (5)).

2. EXPERIENCES

Together with Industrilogik (and earlier other consultants), the Swedish National Rail Administration (Banverket) has investigated the use of formal methods for several years.

One of the earliest projects was the development of a formal specification for the functional safety requirements of interlocking systems. This specification was limited in scope in order to see what could be achieved before any work was spent on developing a more complete specification. It should be emphasised that the specification was independent of any particular interlocking design and only concerned itself with signalling rules.

During development of the formal specification it was found that the existing signalling rules were unclear in some cases. A number of decisions had to be taken to complete the regulations before the specification work could be completed.

An unexpected experience was that the effort of writing the actual requirements was minor compared to the effort of writing formal models of the concepts needed to express the requirements. One simple requirement is that the signal aspect of a distant signal should match the aspect of the next main signal after the distant signal. It turns out that – although intuitively obvious – the concept of ”next signal” was difficult and time-consuming to model.

Using a simulator tool, the specification could be simulated on a computer in the sense that given a model of a rail yard, the computer would simulate the behaviour of an interlocking that satisfied the requirements. This provided additional insight into both the specification itself and the signalling rules. Indeed, given a sufficiently fast computer with the proper interfaces, the simulator could (apart from system safety issues) have worked as an actual interlocking system controlled directly by the formalised signalling rules!

To assess the feasibility of formal verification, an existing interlocking system was formally verified. The system chosen was a small relay based interlocking of a common design. A relay based interlocking was chosen because the only computer based interlockings in Sweden at the time were very complex and intended for large rail yards.

It turned out that the formal proof could be done automatically by computer in a few minutes time.

The result of the verification was that a safety-critical design error was found. Under certain circumstances, a train route could be locked without the proper overlap, resulting in a collision risk if a train would fail to come to a complete stop at the end of the route. The same error affected about two dozen other installations which were subsequently rebuilt.

Although the interlocking was of a very common and well understood design and the error was obvious once pointed out, the error had still slipped past years of design inspections by experienced signalling engineers.

This work is described in Eriksson (1) and, in more detail, in Eriksson (2) and Eriksson (3).

In connection with this work, a knowledge transfer experiment was made in which an engineer from Banverket would carry out a formal verification of another interlocking installation. This experiment was successful. The experiences from this and other case studies show that engineers quickly learn to understand and work with formal specifications. Writing new formal specifications is a more difficult matter, but also one that is less often needed.

A more recent experience is the development of a formal specification for contractual purposes. Banverket has contracted the development of a new generation of small computer-based interlocking systems. The development is being done according to the CENELEC draft standards for safety-critical applications in railway applications.

During the course of the work it turned out that the plain language functional specifications written by Banverket – although considered to be of high quality – did not suffice. As with any plain language specification, this specification was open to interpretation and in many cases it did not specify the interlocking system with sufficient precision. During traditional development, this would have been a minor problem as the administration and supplier could agree on a suitable interpretation in each case.

According to CENELEC procedures, however, the development is overseen by a ”validator” – a third party who will make an independent assessment of the compliance of the developed system with its specifications. In this case the validator felt that having to discuss and agree on how to resolve ambiguities and omissions of the specification prevented them from making an independent assessment.

It was agreed between all parties that this dilemma should be resolved by rewriting the functional specifications as fully formal specifications.

There are many other examples of how formal methods can benefit the development of signalling systems. We will only mention two: Adtranz Signal has recently begun using formal specification and verification as an integral part of the development process for interlocking software.

Matra Transport has been using formal methods (the ”B-method”) in the development of ATP/ATO systems for several years. Behm and Meynadier (6) reports that the quality of formally verified code is so high that it is essentially error-free. Adtranz Signal has also reported a very large decrease in the amount of necessary testing and debugging.

3. AN INTERLOCKING SPECIFICATION LANGUAGE

An important factor for the success of practical use of formal methods is the availability of generally accepted description languages, methods, and tools. In the work done so far, there has been little common ground apart from basic principles.

The development of a generally accepted method to produce specifications that are complete, clear and unambiguous could enhance competition in the signalling systems market.

It would be detrimental to the competition in this market if only one or some suppliers could offer tools for formal specification and analysis of safety properties of signalling systems. Also, the use that each individual supplier makes of formal methods may not coincide with the interests the rail administration has in the use of formal methods.

For these reasons, the Swedish National Rail Administration considers it important to take an initiative in developing a standard for formal description techniques and tools in the railway signalling area. The foremost aim in this initiative is the development - in cooperation with other interested parties – of a formal description language for the description of signalling safety requirements – the Interlocking Specification Language (ISL).

ISL is envisioned as a general purpose specification language augmented with a set of concepts and principles specifically designed to simplify the development of specifications of signalling systems. Such concepts and principles would provide the foundation upon which each individual administration or supplier could build the description of requirements.

A standard language will lead to greater acceptance of formal methods, a greater interest in the development of high quality tools and simplify communication of requirements between administrations, authorities and suppliers. In particular, a formal specification will facilitate development of signalling systems according to CENELEC norms.

The need for such a specialised language is made clear by the fact mentioned earlier that in a formal specification of safety requirements, the actual requirements comprise only a small part of the formal description. The major parts consists of definitions of the (usually intuitively obvious) concepts needed to express the requirements.

The basis for ISL could be an existing language, such as Z, or it could be newly developed. In any case, ISL should include an extensible and adaptable library of concepts used to express requirements – such as the ”next signal” concept.

Since the use of computer tools is of major importance for the practical use of formal methods, it is important that the specification language is carefully designed to obtain maximum benefit from state of the art of tool technology. There is a conflict between the expressibility of a language and of the extent to which reasoning in the language can be automated. A high degree of expressibility means that ”much can be said in few words”, this has the drawback that the task of automated reasoning systems become harder.

The parts of a formal specification generally requiring the greatest expressibility are the ones dealing with concepts. The parts dealing with the actual requirements generally require less expressibility. If the language is sufficiently expressible to permit a natural description of the concepts, the possibility of automating formal reasoning with the language is reduced.

By defining ISL as a general purpose language together with a predefined concept library, it is possible to implement the concepts as built-in special cases in the tools. In that case, the actual formal specification requires less expressibility, making it more amenable to automatic reasoning. If an existing language such as Z is used as the basis for ISL, restrictions could be put on the language to reduce the expressibility. As tool technology advances, these restrictions could later be eased.

The ISL with associated tools should support several tasks:

- formal requirements specification
- simulation of specifications
- design/formal description of new installations
- formal verification of interlocking systems
- automatic generation of interlocking logic (possibly)

4. EXAMPLES

As examples of formal specifications, we take requirements from Banverket’s formal specification. The language used is a variant of predicate logic. The exact formulation of these specifications can be discussed, they are intended only for illustration.

- If a point is occupied, it must be locked

```
ALL pt (occupied(pt) -> point_locked(pt))
```

This requirements states that for every point, if the track circuit of the point is occupied, the point must be locked.

The *universal quantifier* `ALL pt` states that the following formula must hold for every point `pt`. `occupied(pt)` is a representation of the fact that the track circuit of `pt` is occupied. `point_locked(pt)` is a representation of the fact that the `pt` is locked. The *implication symbol* `->` states that the formula on its left side requires the formula on its right side to hold.

- All points in a locked train route must also be locked

```
ALL pt (SOME tr (locked(tr) AND
part_of(tr,pt))) -> point_locked(pt))
```

This requirement states that for every point it must be the case that if there is some train route that is both locked and includes the point, then that point is locked.

The *existential quantifier* `SOME tr` states that the following formula must hold for some train route `tr`. `locked(tr)` is a representation of the fact that the train route `tr` is locked. `part_of(tr,pt)` is a representation of the fact that the point `pt` is a part of the train route `tr`. The *conjunction* `AND` states that the formulae on its left and right sides must both hold.

- A locked point must not be instructed to change its position

```
ALL pt (point_locked(pt) ->
          (PRE left(pt) -> left(pt))) &
ALL pt (point_locked(pt) ->
          (PRE right(pt) -> right(pt)))
```

This requirement states that for every point it must be the case that both if the point is locked and was in the left position in the previous moment, then it must be in the left position now and also the same condition for the right hand position.

The *previous moment-operator* `PRE` states that the following formula refers to the situation “a moment” ago.

5. CONCLUSIONS

The experience of Banverket and others show that formal methods has become a mature technology that can be used in practice for the specification and verification of signalling systems. The quality of specifications and systems increase while the amount of testing and debugging needed is greatly reduced.

Some suppliers of interlocking equipment are already using formal methods as integral parts of their development processes. Many other parties show a great interest in formal methods. We conclude that any interlocking development project would benefit from the use of formal methods.

Also, Experience has shown that it is difficult to carry out development according to CENELEC norms without the use of formal specifications.

To achieve the greatest benefits a common specification language (ISL) should be developed. This will encourage tool construction and facilitate communication of requirements between different parties.

The ISL should be a general-purpose specification language together with a library of predefined railway-oriented concepts.

The development of a generally accepted method to produce specifications that are complete, clear and unambiguous would clearly enhance competition in the signalling systems market.

6. REFERENCES

1. Eriksson, L.-H., 1997, ”Formalising Railway Interlocking Requirements”, Technical report 1997:3, The Swedish National Rail Administration.

2. Eriksson, L.-H., 1997, ”Formal Verification of Railway Interlockings”, Technical report 1997:4, The Swedish National Rail Administration.

3. Eriksson, L.–H. and Johansson, K., 1998, ”Using formal methods for quality assurance of interlocking systems”, In: Mellit, B. et.al. (eds.), ”Computers in Railways IV”, Computational Mechanics publications..

4. Stålmarck, G. and Säflund, 1990, M., ”Modelling and Verifying Systems and Software in Propositional Logic”, Proc. SAFECOMP ’90, Pergamon Press.

5. Groote, J.F., 1998, ”The propositional theorem prover HeerHugo”, World Wide Web, http://www.cwi.nl/~jfg/heerhugo.html.

6. Behm, P. and Meynadier, J.-M., 1998, ”Météor; an Industrial Success in Formal Development”, Proceedings of the first FMErail seminar, Origin Nederland B.V.