Application of Formal Methods for Specification of Safety-Relevant Traffic Process Control Systems

Dr. Géza TARNAI, Balázs SÁGHI

Technical University Budapest
Department of Transport Automation

ABSTRACT

In this paper we would like to demonstrate the current problems of system development and to show, how formal methods can eliminate some of these problems. First the safety process will be shown with its participants and their roles. Following the developing phases will be presented from the customer statement of requirements to the system specification, with their classification and deficiencies. In the next chapter the basics of the formal methods will be shown, and their role in modelling, testing, verification and simulation. In the last chapter some current application examples are presented, and the future tasks a sketched in research and education.

Keywords: formal methods, safety-relevant systems

1. INTRODUCTION

The safety process, which must be carried out by the establishing every new safety-relevant system has mainly four participants, namely the customer or user, the developer or manufacturer, the independent expert, and the authority. The requirements of the system are formed by the customer. The developer checks the requirements in form of an analysis and develops the system. The system must be verified by the independent expert, who is charged by the customer. The positive report of the expert is a precondition for that the customer can get the authority's approval for the installing of the system.

In this process the customer plays the central role (Fig. 1). The connection between the customer and the developer is extraordinary important, where a good understanding between the partners is (or would be) necessary. This can be achieved by using a unified, consistent communication language at every developing phase and communication levels.

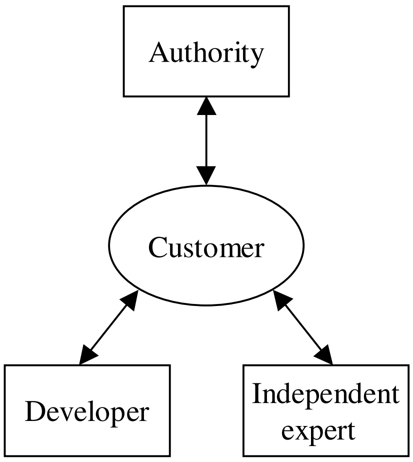

Figure 1

For the specification of a given system different documents are used. The different documents have different objects, they specify the system from different aspects. It is a very difficult task to verify and proof the consistency and the unanimity of this specification, that contains a lot of different documents.

2. FROM CUSTOMER REQUIREMENTS TO SYSTEM SPECIFICATION

The starting point in the vast majority of system development projects is the customer requirement. The requirements are formed in the statement of requirements. This is an informal description of the properties of a new system or an addition to an existing system. Normally it is written in natural language.

The statement of requirements can be classified into several types of statements. There are

- functional requirements which detail what the system is to do;
- non-functional requirements which contain directives which will constrain the developer in subsequent activities in the life cycle;
- goals, that are statements which guide the developer where choice exists;
- references to data used by the proposed system;
- design directives or implementation directives; these may overconstrain the system designer and result in a non-optimal system.

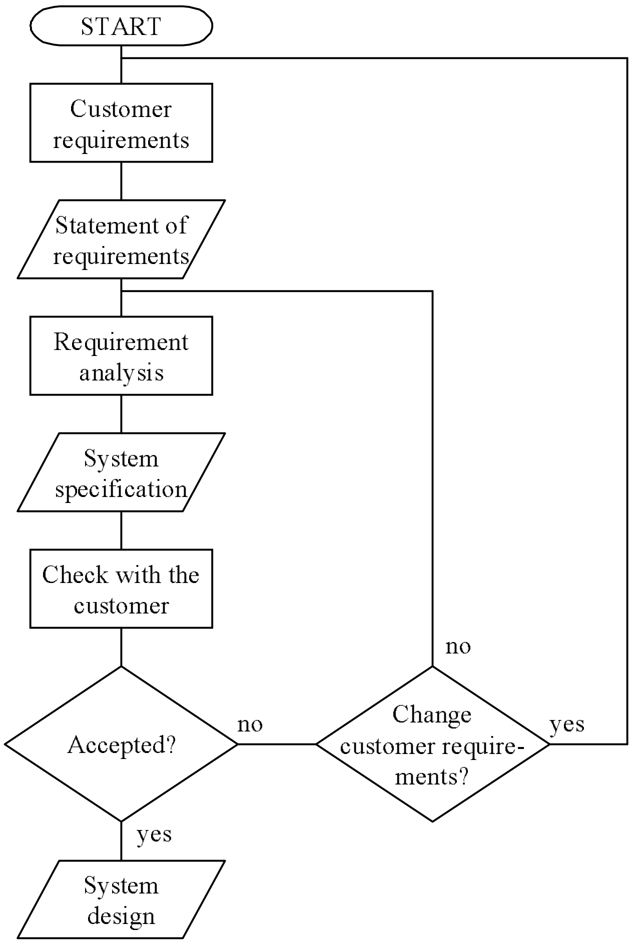

Figure 2

The safety requirements by a safety-relevant system can be formed as a separated class or as a part of the functional and/or non-functional requirements.

Life would be uncomplicated for the system designer if the statement requirements consisted of a series of sections marked as functional and non-functional requirements, goals, data requirements and implementation and design directives, and each were consistent, unambiguous and complete. The most common deficiencies are

- vagueness,
- contradiction,
- incompleteness,
- mixed requirement-classes,
- naivety; the customer may over- and underestimate the capabilities of the future system.

During a requirement analysis the developer produces from the customer statement of requirements the system specification (Fig. 2). As the system specification is the base document for the subsequent phases of the life cycle it must be clear, consistent and unambiguous. To this the mentioned deficiencies must be avoided and/or corrected.

The system specification describes the characteristics of the system as perceived by the developer. So the system specification must be checked by the customer too. If the customer agrees with the specification, he approves it. Then the system specification can be used as base document for subsequent developing activities.

3. FORMAL METHODS IN SYSTEM DEVELOPMENT

A large number of problems of the system development arise from the unsatisfactory nature of the notations used to describe a product as it progresses through its life cycle. A typical example is a software development: the developer processes a statement of requirements in natural language (produced by the customer), constructs a system specification e.g. in a graphical notation, refines this into a detailed system design consisting of a set of control constructs and natural language, and then codes the software in a programming language.

The natural language is a medium in which it is very difficult to be precise, to state essential properties of a system without being cluttered be detail and to reason with. Unfortunately, those notations which are most heavily dependent on natural language are those which occur at the beginning of the life cycle, where undetected errors can lead to massive overruns or even total cancellation.

A second problem occurs because of the fact that developers use a variety of notations in a project. Often a different notation is used for each phase. This results in errors at the interface between phases.

It is well known, that with program languages it much easier to describe more precise algorithms and their implementations than with natural language. It has been tried to use this property at the description of the hardware of digital systems. A number of hardware description languages has been developed, during the last decade. The most common and accepted is VHDL. VHDL allows to use the same syntax for describing the behaviour of a system and for describing the structural design of the system. This is true for element level from logical gates or switches to processors or memories as well as for unambiguous specification of abstract level.

It is most desirable to use a precise, uniform notation for all of the developing phases of the whole system. A possibility is using formal methods. Formal notations allow a full, integrated notation of a system, from all of the different aspects [9].

The formal methods have their roots in the researches for automated function proofing in the late 60s, early 70s. In the last decade the researches concentrated on the mathematical notations and the joined methods for system specification and development.

The formal methods use mathematical (logical, set theoretical) notations to describe a system in each developing phase. Mathematics has a number of properties which are desirable for system specification. These are:

- it is easy to represent levels of abstraction;
- it is easy to reason in mathematics;
- mathematics is concise;
- mathematics has well-defined and unambiguous meanings;
- mathematics can be used to model reality;
- mathematics can be used to suppress unnecessary detail.

If we compare the formal languages and the natural language we can see that the syntax and semantics of informal languages are intuitive, while the formal languages have formal syntax and semantics. It means that the formal techniques have strict conventions and syntactical rules/prescriptions. An informal description is easy to read, but it often ambiguous. The readability of formal notations is worse, but it is always unambiguous. Because of its strict syntax and semantics the formal techniques can be automated, what is hardly possible by informal techniques [5].

For the developer it would be reasonable to describe even the customer statement of requirements or the system specification in a formal language, but the fact, that the customer may not understand the formal notation makes it only in a few cases possible. This difficulty can be solved by producing a simulation model from the formal specification. The model, with scenarios can be used to visualise the system behaviour for the customer.

The formal methods will not eliminate the importance of natural language as the customer requirements will always be formed in natural language, but it will play a smaller role.

3.1. Formal Models

Not only the communication with the customer needs a model: for the system development is also needed a formal model of the projected system, which should be able to implement and useable in different developing phases.

A model is an abstraction of a for us relevant part of the reality. The not important parts must be ignored. The environment of the system must be modelled as well. The environment model contains all the joined system, which have an interface to our system.

To produce a model a notation, a method and a tool is needed. A notation is the set of the symbols and the conventions of their combinations. Notations are used to describe a task, requirements, the implementation and its process and the results and their documentation. A method is a projected process to form requirements, to integrate different aspects, to proof and to the quality assurance. Efficient tools are needed to handle the descriptions and methods. Under tools we understand nearly always a computing systems.

A model is a formal description of a system specification. Such a model provides series of opportunities. If the model can be executed, it is possible to simulate the functional behaviour of the system. With help of simulations the errors can be revealed and of course corrected in an early phase of the system life cycle, in the specification phase. On the base of the simulation it is possible to carry out different tests. The model can be used for quality assurance and quality proof. Last but not least from an executable model it is possible to produce automatically the source code and the program code of the real system.

3.2. Verification and Validation

The verification and validation are essential phases of system development. Verification is concerned with checking that one phase in the project accurately reflects the intentions of the previous phase. Validation is concerned with checking that the product of a phase matches the customer requirements.

As the systems become more and more complex, the verification became one of the heaviest problems. If it is a safety-relevant system, not only the functional behaviour must be verified, but also the safety of the system. The safety requirements can appear in the frame of functional specification. In this case the methods used for safety verification are similar or the same as by the functional verification. But there are special components of the safety of which specification is not possible in the frame of functional specification (e.g. specification of fault tolerance).

3.3. Testing

Both verification and validation are supported by tests. During a test on a predefined input sequence given answer-output sequence is compared with the expected output sequence of the system. The requirements of the test are defined on the base of the purpose of the test with consideration of the time/economical limits. During the test projecting the required test-set and the belonging input sequence are generated. The reference sequence is generated with the help of the input sequence and a simulation model, which creates the expected outputs of a "good" system.

After the prepare phase can be carried out the test itself. The result of the compare is evaluated and fixed in the test documentation.

The formal methods and the using of computer-based tools can simplify some of the basic problems of testing [14].

- If formal model is executable, the verification can be carried out as first test. This reduces the number of necessary traditional test.
- The horizon of the system specification can be a base for the test environment. To the formal model produced environment model is a good and validated test environment.
- If the system is described with formal notations and verified developing tools are available, the number of traditional test can be heavily reduced. The formal methods help to reveal errors already in the specification phase, thus they must not be find by tests.

The formal methods do not eliminate the necessity of test. The customer will always need a demonstration with real data before the installation so as to be sure that the system is functioning properly. In the same way tests are necessary after repairing as well.

3.4. Simulation

The simulation not only for communicating with the customer or verification or test generation can be used, but it can be an independent training system or projection tool. The formal methods play an important role in these applications as well. The simulated object and its environment can be effectively described using formal description languages.

4. RELEVANCE OF FORMAL METHODS TODAY AND TOMORROW

As in this paper presented, the formal methods could help to eliminate some of the major problems in system development. In some partial areas formal methods have already been used advantageously.

For example for the testing of relay-sets a microprocessor based system was been developed on the TU Budapest in the Department of Transport Automation as far back as the 70s. The starting point of the process is the circuit of the relay-set. To describe the circuit a formal language was used. From the formal description the machine generated the required test set automatically. Further the machine executed the tests, compared automatically the real and the expected output sequences, and produced the test documentation, all of these automatically (Fig. 3) [16, 17, 18, 19].

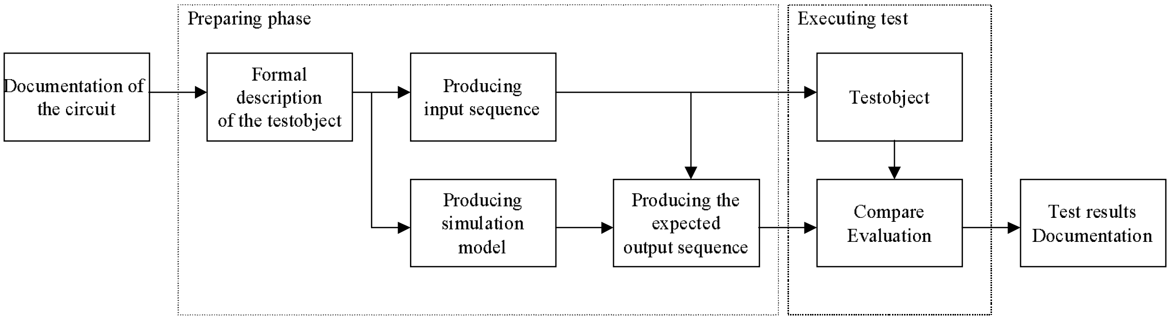

Figure 3

An other example from the area of simulation is a railway interlocking simulator and projecting tool, RailCAD, which was developed by the Stellwerk Ltd. The RailCAD uses formal descriptions for each simulated subsystems. This and a natural language user interface makes it possible to describe easily any kind of interlocking logic, without the modification of the source code. The ability, that the customer can redefine the interlocking logic makes the system an effective developing tool [1]. The scheme of such a training system is shown on Fig. 4.

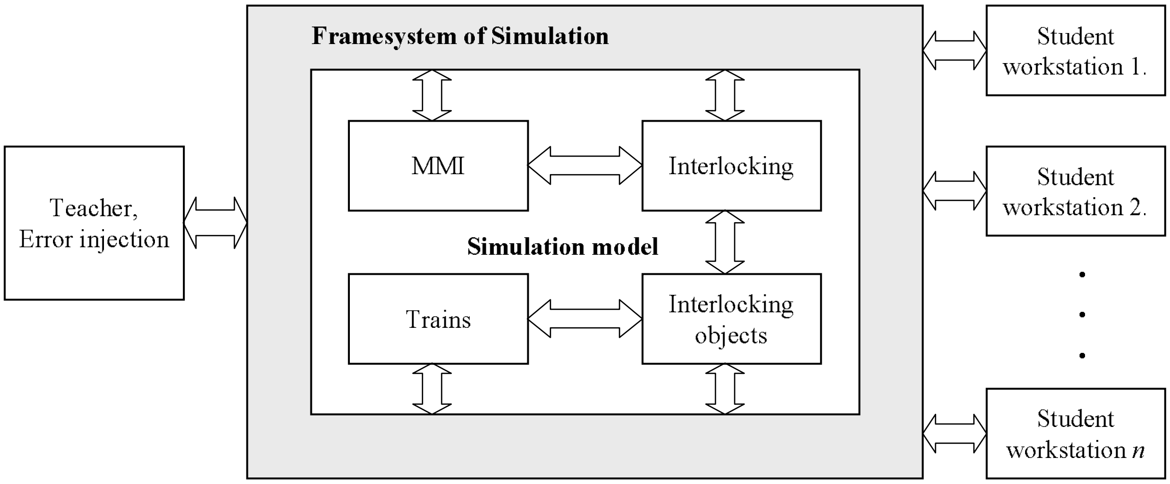

Figure 4

A serious task was the modelling of the European Train Control System (ETCS), which was carried out at the TU Braunschweig, Germany, in the Department of Control and Automation. The produced complete formal modelling contains three models: the model of the vehicle, the model of the track and the model of the environment. The last one contains all the adjoined system with an interface [11].

The last example is a railway interlocking project. In the frame of the modernising the Budapest-Vienna line, a new electronic railway interlocking system was installed by Siemens. The functions of this equipment must have been completed with a train control system. The task was solved with safety PLC-s. In this project the verbal requirements must have been made clear and unambiguous. To do this a formal description language was created, which was easy to understand by the relay interlocking experts of the railway and on the other hand this notation could be used for implementation in STEP 5 program language without any modification. This method simplified the communication with the customer and the verification as well [6].

The experts show great concern about the question of the unambiguity and completeness. Several attempts have been tried to reveal the verbal ambiguity and to form precisely functional relations. But because of the functional and sequential complexity of the traffic process control systems a complete and verifiable formal functional description has not been produced until now.

In the future is expected the developing of such a formal method, that can be applied for complete functional description in each developing phase from customer statement of requirements to the system implementation. An other expectation is to extend the validity of formal methods from new system projects to system additions or developments. This is especially important at the transportation, where a large rate of the projects are developments or additional developments.

As everywhere on the universities, on the TU Budapest in the Department of Transport Automation we have double duties: research and education. The relevance of formal methods are high on both areas. For stepping forward in research we intend to start new PhD programs in this topic.

The formal methods should be involved in graduate education as well. There are two subjects in our department in which formal methods could be inserted. One is Transport Automation, where general safety principles are educated such as risk analysis, reliability theory, safety strategies, safety system life cycle and safety case. In this subject the formal methods can be taught at a lower level. In the sub-module called “Safety-Relevant Traffic Process Control Systems” may the formal methods be educated at an advanced level. In one of the subjects of this sub-module, called “Safety-Relevant Computer Based and Hybrid Systems” the students learn specification techniques. This subject could be completed with formal methods.

The students, who graduate in our faculty (Faculty of Transportation) may work on different sides of the safety process, mentioned in the Introduction. They may be customers, developers and experts too, and may work for the authority as well. Considered this our possibility (and responsibility) is very high to spread the using of formal methods.

REFERNCES

1. Aranyosi, Z., Mosóczi, L., Rácz, G., Tarnai, G.: “Training the Personnel Using Simulator and Training System”. World Congress on Railway Research WCRR’97 Proceedings Firenze 1997 Vol. A pp. 745-751.
2. Bernardeschi, C.: “Proving Safety Properties for Embedded Control Systems” EDCC-2 Taormina, Italy, October 1996 pp. 321-332.
3. Bowie, M., Ingleby, M.: “ERRI A201 Committee: Harmonization of European Interlocking Specification”, IRSE Proceedings, 1997.
4. Brownbridge, D.: ”Specifying BR Signaling Rules”, IRSE Proceedings, 1997.
5. Ehrig, H.: “Relevanz, Integration und Vergleich formaler Spezifikationstechniken”. FORMS ’98 12-13 May 1998. Braunschweig, Germany.
6. Görög, B., Szabó, G., Tarnai, G.: “Safety and Reliability Analysis of PLC based safety functions”. Specialist journal “Vezetékek Világa” Budapest 3/98 pp. 6-10 (in Hungarian)
7. Hall, W., Parádi, F.: “BEST-ESTW – Neues Simulationssystem zur Schulung von Fahrdienstleitern”. Signal+Draht, 1996., H. 7/8.
8. Ince, D.: “An introduction to discrete mathematics, formal system specification and Z” Oxford University Press Inc., New York 1995.
9. Jahnsen, A.: “Anwendung von Modellen zur Unterstützung der Kommunikation und Klarheit von Spezifikationen” FORMS ’98 12-13 May 1998. Braunschweig, Germany.
10. Keresztes, P.: “Theoretical Questions of Functional and Safety Verification of Digital Systems”. Specialist journal “Vezetékek Világa” Budapest 2/96 pp. 16-19 (in Hungarian)
11. Meyer zu Hörste: “Die Formale Modellierung und Simulation von ERTMS/ETCS mit Petrinetzen” ”. FORMS ’98 12-13 May 1998. Braunschweig, Germany.
12. Nelli, M.: “Dependability Modelling and Analysis of Complex Control Systems: An Application to Railway Interlocking” EDCC-2 Second European Dependable Computing Conference Taormina, Italy, October 1996. pp. 93-110.
13. Schnieder, E.: “Zum Geleit” ”. FORMS ’98 12-13 May 1998. Braunschweig, Germany.
14. Schulz, H.-M.: Komplexität des Tests technischer Systeme mit Blick auf Leitsysteme im Schienenverkehr” FORMS ’98 12-13 May 1998. Braunschweig, Germany.
15. Tarnai, G., Parádi, F.: “Supporting Railway Project and Test Processes by Models ” Zeszyty Naukowo-Techniczne Oddzialu Krakowskiego SITK, Zakopane 1992, Nr. 24. pp. 229-239.
16. Tarnai, G.: “Ein algebraisches Modell für Relaiseinheiten von Eisenbahnsicherungsanlagen” Periodica Polytechnica, Transport, Budapest, 12/1984/1.
17. Tarnai, G.: “Automatical Testing of Relay-Sets of Interlocking Systems”. Dissertation for the degree of candidate of sciences. Hungarian Academy of Sciences. Budapest, 1984. pp. 152. (in Hungarian)
18. Tarnai, G.: “Safety Verification for Train Traffic Control Communications” IEEE Journal on Selected Areas in Communications SAC-4 October 1986. No 7. pp. 1118-1120.
19. Tarnai, G.: “Testing microprocessor equipment for railway safety” Sixth Symposium on Reliability in Electronics. Budapest, August 1985. pp. 539-545.