NEW RESULTS AND TRENDS IN FORMAL TECHNIQUES & TOOLS FOR THE DEVELOPMENT OF SOFTWARE FOR TRANSPORTATION SYSTEMS — A REVIEW[1]

Dines Bjørner
Computer Science and Engineering, Informatics and Mathematical Modelling
Technical University of Denmark, DK–2800 Kgs.Lyngby, Denmark
E–Mail: db@imm.dtu.dk, URL: www.imm.dtu.dk/~db

Abstract: We characterise what is meant by a metod in the context of software development. Then what is meant by a formal technique. We refute the possibility of formal methods, but express the desirability of formal techniques. Some such techniques are briefly surveyed. We will outline what has been done recently and what is currently being done using formal techniques in the area predominantly of railway systems. Problems are outlined, as are avenues for future research. Being an invited survey, the paper features an extensive, albeit far from complete, literature reference list of almost 180 entries, taking up half the paper size ! There are no examples of formal techniques being actually shown in this paper — but there should be at least three papers (Pěnička et al., 2003; Strupchanska et al., 2003; Haxthausen and Peleska, 2003b) in these proceedings which illustrate such techniques as are the subject of the current review.
Keywords: Formal Method, Domain Description, Requirements Prescription, Software Design, Provable Correctness, Safety Criticality, Real–time, Embedded Systems, Interlocking

1. INTRODUCTION

Transportation systems pose extraordinary challenge when it comes to their monitoring and control by combinations of classical automatic control systems and digital computers.

1.1 Infra–Structure

This is in particular the case for rail and air systems due to their hard real–time characteristics combined with the need for very high dependability. In these kinds of infra–structure systems we see a need to integrate many diverse management planning, and operational execution, monitoring and control facets.

1.2 Sub–System Interfaces

Thus the challenge is compounded by the possibility, when using computers, of combining many diverse "sub-systems", sub–systems that, in the days of only combinations of classical automatic control system and human monitoring and control, were quite "separate": Where information from one sub–system was basically only handed on to another sub–system via human intervention. Such human intervention often entailed data vetting: Is the information to be passed–on relevant and valid ?

With automated interfaces, even within purely digital computer, ie., software, controlled sub–systems, the problem of "switching domains" is staggering, but enticing.

[1]The writing of this paper, as well as the papers (Pěnička et al., 2003; Pěnička et al., 2003), also contained in these proceedings, and their presentation at Budapest, is sponsored by the EU IST Research Training Network AMORE: Algorithmic Models for Optimising Railways in Europe: `www.inf.uni-konstanz.de/algo/amore/`. Contract no. HPRN-CT-1999-00104, Proposal no. RTN1-1999-00446

1.3 Hybrid Systems

Perhaps, from a scientific and engineering point of view, the most obviously interesting area of study is that of hybrid systems: These are not just combinations of continuous and discrete systems, but are such in which there is not just one, but several controllers — to use a standard terminology in automatic control theory. The software controls the decisions when to exchange one controller for another. Such hybrid systems have been studied at UNU/IIST, the UN University's Intl. Inst. for Software Technology, Macau, nr. Hong Kong: (Wang et al., 1994; Chen et al., 1994b; Chen et al., 1994a; Yu, 1994a; Hung and Wang, 1994; Widjaja et al., 1994; Wang and He, 1995; Zhou et al., 1995). An interesting concept in this connection is 'Hybrid Automata' (Henzinger, 1996). Hybrid automata combine discrete transition graphs with continuous dynamical systems. They are mathematical models for digital systems that interact with analog environments. Hybrid automata can be viewed as infinite-state transition systems, and this view gives insights into the structure of hybrid state spaces.

1.4 Structure of paper

The topic of this invited paper was suggested by the Programme Chair. In effect they chose the title ! It therefore behooves me to explain some of the terms of the title such as they are understood in computing science and software engineering. We therefore first explain — to an audience usually associated with the field of automatic control — what is meant by a method, its principles and techniques; when such techniques can be based on mathematics, and in particular on discrete mathematics, including notably mathematical logic.

2. ON TECHNIQUES & TOOLS

2.1 What is a Method ?

By a (software development) method we shall understand a set of principles for selecting a number of techniques and tools for the study and solution of problems — in the form of the construction of an artifact (here: Software).

2.2 Impossibility of Formal Methods !

The method principles amount to criteria for selection and application: When and what to choose. Such principles can not be automated. Problems under investigation are usually too complex and "never the same". So the choice has to be done by the developers. Hence cannot be formalised. Unfortunately the term 'Formal Method' has stuck.

2.3 Desirability of Formal Techniques & Tools

But many techniques can be formalised, and tools can be provided for the support of such formal techniques. These techniques apply to oftentimes very large scale engineering documents, formally specified, and far too large to be analysed by humans. It is therefore desirable to use such formal techniques and tools — since they may help ensure, amongst others, correctness of software with respect to likewise formally prescribed requirements.

2.4 Examples of Techniques & Tools

We take specification languages, and correctness (of software or hardware) verifiers and model checkers to be tools. By techniques we then mean the specific way in which the developer performs 'calculations' (ie., development steps), including applying these tools. Verifiers are software packages that either assist the developer in conducting proofs of correctness or which perform such proofs more or less automatically. Model checkers are also software packages which symbolically — almost "exhaustively" — tests the software (hardware), while subject to usual computer interpretation, enters only desirable states. Certain kinds of compilers from domain specific language scripts are tools that transform specifications into semantically consistent executable systems.

2.5 Why Formal Techniques ?

There are several, and in the mind of the current author, fully equivalent, good reasons for why one should apply formal calculi (ie., techniques) in the pursuit of computing systems development: Usually one is mentioned as being the most important one: Correctness of software — with respect to requirements (ie., that *"the software is right"*). To this we add that *"it is the right software"*, ie., that it affine to users expectations. Finally: *"it is fun, it is professionally satisfying"* to use formal techniques.

2.6 Convincing the Skeptics

The subject of so–called Formal Methods, is, strangely, to the curent author — who is one of the "pioneers" of the field (within software) — still somewhat "controversial". So a number of popularising papers have been and are being offered: (Wood, 1990; Wing, 1990; Thomas, 1992; Bowen and Stavridou, 1992; Bowen and Stavridou, 1993; Bowen, 1993; Rushby, 1993; Bowen and Hinchey, 1994; Butler et al., 1995; Cleland and MacKenzie, 1995; Hinchey and Bowen, 1995; Kelly, 1995; Liu et al., 1995; Rushby, 1995; Caldwell, 1996; Kelly, 1996).

These papers explain why developers might very well wish to use formal methods in the development of software. (Rushby, 1993; Kelly, 1995; Rushby, 1995; Kelly, 1996) explain very well NASAs position on the need for formal verification of safety critical on–board software.

(Bowen and Hinchey, 1995b; Bowen and Hinchey, 1995a; Hall, 1990) provides (entertaining) capsule advice to "skeptics".

3. SOME TECHNIQUES & TOOLS

In this section we overview, ever so briefly, some of the formal techniques that have shown effectiveness in solving problems — also in the domain of railway operations and management.

The possible distinction between a method, like `ASM, B, RAISE` or `VDM`, a specification language, like `CSP, RSL, VDM-SL` or `Z`, or a technique cum tool, like `SPIN` — for all of these see below — has here been deliberately blurred.

The next section will then comment on specific uses of these formal techniques.

We have ordered the presentation of the techniques chronologically: In the approximate order of their publication.

3.1 Petri Nets

`Petri Nets` are a two–dimensional, ie., a graphic, yet formal notation for expressing concurrent behaviours and true simultaneity. Leading book references are: (Jensen, 1997; Reisig, 1985; Reisig, 1992; Reisig, 1998). The nets are named after their "creator" Carl Adam Petri (Petri, 1962).

3.2 VDM–SL

`VDM` stands for the 'Vienna (software) Development Method'. `VDM-SL` stands for the `VDM` Specification Language. It was researched and developed at the IBM Vienna (Austria) Laboratory in the early 1970s and can be said to be have offered first comprehensive formal techniques for general software development. `VDM-SL` is now an ISO standard (Larsen et al., 1996).

`VDM` basically offers model oriented, ie., discrete mathematics means of specifying and reasoning about software. Major texts are (Bjørner and Jones, 1978; Bjørner and Jones, 1982; Fitzgerald and Larsen, 1997).

The autor of this paper was one of the co–designers of VDM.

3.3 CSP

`CSP` stands for 'Communicating Sequential Processes'. Put forward by Tony Hoare in 1978 (Hoare, 1978) `CSP` has become one of the leading means for specifying, succinctly and elegantly, the interaction between parallel processes. Leading books on `CSP` are: (Hoare, 1985; Roscoe, 1997; Schneider, 2000)

3.4 Z

`Z` derives from the `Z` in Zermelo, who, as a mathematician, together with Frankel, established the Zermelo–Frankel axiomatic basis for a set theory.

Proposed around 1980 by Jean–Raymond Abrial, `Z` has become one of the leading model oriented, ie., discrete mathematics means of specifying and reasoning about software. The `Z` literature is abundant, but we refer only to the delightful text book: (Woodcock and Davies, 1996).

3.5 Statecharts

`Statechart`, primarily put forward by David Harel in the mid 1980s, and supported by the `Statemate` tool set (Harel and Naamad, 1996), is a two dimensional graphics, ie., a pleasant visually oriented way of presenting concurrent behaviours by the ("similar") behaviour of compositions of modularised sets of finite state machines. `Statechart` is a "feature" offered by `UML`.

A leading text book is: (Harel and Politi, 1998).

3.6 HOL

From the `HOL` home page[2] we quote: *"The `HOL` System is an environment for interactive theorem proving in a higher–order logic. Its most outstanding feature is its high degree of programmability through the*

[2] `www.cl.cam.ac.uk/Research/HVG/HOL/`

meta–language ML. The system has a wide variety of uses from formalizing pure mathematics to verification of industrial hardware. Academic and industrial sites world–wide are using HOL. The system is available without charge." Leading texts are: (Gordon and Melham, 1993)

3.7 Isabelle

From the Isabelle home page[3] we quote: *"Isabelle is a popular generic theorem proving environment developed at Cambridge University, England (Larry Paulson), and at the Technical University of Munich, Germany (Tobias Nipkow)."* Isabelle is strongly related to HOL. There is a forthcoming book on Isabelle (Nipkow et al., 2002).

3.8 ccs

ccs stands for 'Calculus of Communication Systems'. Put forward by Robin Milner (Milner, 1980; Milner, 1989) ccs provides a mostly theoretical framework for studying, investigating and experimenting with concurrent behaviours. ccs is reminiscent, but, in most respects, independent of CSP.

3.9 RAISE

RAISE stands for Rigorous Approach to Industrial Software Engineering. RAISE, with its Specification Language RSL, is a derivative of VDM, incorporating algebraic semantics (scheme, class, object) structuring constructs and CSP. Leading texts on RAISE are (George et al., 1992; George et al., 1995). RAISE can be claimed to be an object–oriented (i., OO) language.

The author of the present paper instigated RAISE in the mid 1980s. He is now launching a major text book of software engineering using RAISE: (Bjørner, 2004).

3.10 PVS

From the PVS home page[4] we quote: *"PVS is a verification system: that is, a specification language integrated with support tools and a theorem prover. It is intended to capture the state–of–the–art in mechanized formal methods and to be sufficiently rugged that it can be used for significant applications. PVS is a research prototype: it evolves and improves as we develop or apply new capabilities, and as the stress of real use exposes new requirements."* Leading people "behind" PVS are John Rushby and Natarajan Shankar. Seminal manuals are: (Shankar et al., 1993; Owre et al., 1999a; Shankar et al., 1999; Owre et al., 1999b).

3.11 B

B "derives" from the name of the group of set–theory oriented French mathematicians: Bourbaki. Put forward by the "father" of Z, Jean–Raymond Abrial, B offers utterly elegant means, within again a model–oriented simple, but reasonably abstract (imperatively oriented, as is Z) to specify and notably reason about — and, by means of strong tool support, to formally prove — properties of designs.

The leading text book is (Abrial, 1996).

3.12 ASM

ASM stands for Abstract State Machines. Put forward around 1985 by Yuri Gurevitch, ASM offers operational, some would say algorithmic, ways of specifying and reasoning about software. ASM provides so by means of a state transition system notation. States are algebras. Interpretation of ASM specifications leads to a notion of *evolving algebras.* A leading European "behind" ASM is Egon Börger. The literature on ASM is abundant. Leading books (incl. proceedings) are: (Börger, 1995; Gurevich et al., 2000; Gurevich et al., 2000; Börger and Stärk, 2003; Börger et al., 2003).

3.13 SPIN

SPIN is a reachability analysis tool designed for the general verification of distributed systems. First made available publicly in 1991, SPIN is widely used both for teaching and for industrial applications, and has inspired many other verification tools. In April 2002 the tool was awarded the prestigious System Software Award for 2001 by the ACM. The originator of SPIN is Gerard J. Holzmann. Leading texts are: (Holzmann, 1991; Grégoire et al., 1997; Holzmann, 2003).

[3] www.cl.cam.ac.uk/Research/HVG/Isabelle/

[4] http://pvs.csl.sri.com/

3.14 Duration Calculi (DC)

Of several notations for describing temporal (ie., real–time) properties of systems, we single out the *Duration Calculi*. The main proposer of the DC was and is Zhou Chao Chen, but see (Zhou et al., 1991). DC offers a continuous time temporal logic for specification and reasoning. DC has a number of variants for dealing with probabilistic phenomena, for incorporating first order differential calculi in DC expressions, etc. Leading papers are: (Zhou et al., 1991; Hansen and Zhou, 1992; Zhou, 1993; Liu et al., 1993; Zhou et al., 1993; Zhou and Li, 1994) — with (Zhou and Hansen, 2003) being a monograph on the DC.

3.14.1 HyTech

HyTech (Henzinger et al., 1997a; Henzinger et al., 1997b) *"is a symbolic model checker for linear hybrid automata, a subclass of hybrid automata (Henzinger, 1996) that can be analyzed automatically by computing with polyhedral state sets. A key feature of HyTech is its ability to perform parametric analysis, i.e. to determine the values of design parameters for which a linear hybrid automaton satisfies a temporal-logic requirement."* A leading person "behind" Hybrid Automata and HyTech is Tom Henzinger.

3.15 π–Calculus

The π-Calculus, like ccs, both put forward by Robin Milner, is a reasearch vehicle for studying systems whose behaviour can conveniently be understood in terms of a varying number of processes and channels. Processes can interface over dynamically varying channels. The π-Calculus is not intended as an industry–oriented 'technology'. Main texts are: (Milner, 1999; Sangiorgio and Walker, 2001).

3.16 Remarks

Unlike the natural sciences — where the phenomena studied are manifest, and created by The Almighty God — in the computing sciences, as in mathematics, we deal with concepts conceived by humans. As a result we see, as illustrated above, a plethora of notational systems, diagrammatic and textual. Each reflecting a didactics, a mind–set specific to the time at which the specification language was first proposed — with many such concepts transcending into a long future.

For the natural sciences and its derived engineering branches (civil [building], mechanical, chemical and electrical & electronic engineering), the major notational system of analytic expressions and the major calculi of differentia and integral calculi pervades and have become "standards". No–one would employ a "classical" engineer who was not thoroughly familiar with that mathematics and those calculi.

Alas, this is yet to happen for software engineering !

There are may other formal techniques and tools than those briefly surveyed above. Some will be mentioned in the next section.

4. RECENT WORK

In this section we shall go a little bit into actual applications of formal techniques in connection with railway systems. The present section offers but a mere glimpse. In no way does this section claim to be comprehensive. More, it is a reflection of what the current author has encountered and felt intrigued and/or inspired by.

4.1 FME Rail Workshops

Dr Peter Gorm Larsen, a former student of the current author, initiated a three year, I should say, rather successful, EU sponsored, collaboration (network): 1997–1999.

Proposed through the "offices" of the FME (Formal Methods Europe[5]) association, FME Rail brought practitioners in the railway industry together with researchers from that industry as well as from academia at five workshops: (Larsen, 1998; Woodcock, 1998; Fahlén, 1998; Montigel, 1999; Lecomte and Larsen, 1999).

Many of the references below derive from these workshops. We refer to www.ifad.dk/Projects/FMERail/fmerail.htm[6] .

4.2 A Survey

The survey is ordered by the alphabetic name either of the specific formal technique or tool being predominantly used in the referenced applications, or — in a few cases — by the application subject. Many papers listed under the name of some technique or tool could, as well, have been listed under

[5] www.fmeurope.org/

[6] The author hopes this web page stays alive for some more years.

some application area. This is in particular true for **Interlocking** — as very many papers indeed study that subject.

ASM: (Börger et al., 2000) A report on the use of ASMs at Siemens AG (from May 1998 to March 1999) to redesign and implement the railway process model component of FALKO, a railway timetable validation and construction program.

B: Using B (Guiho and Mejia, 1984; Dehbonei and Mejia, 1994a; Dehbonei and Mejia, 1994b; Dehbonei and Mejia, 1995) reports on what must be considered one of the most successful and spectacular uses of formal methods. The application is that of the software to automatically control high speed urban trains in Paris, France. More than 80.000 lemmas and theorems were proved, using the B tool set `Atelier B`[7], in order to gain confidence in the correctness of the specified design. *"Using the B Method to Design Safety–Critical Software Systems for Railway Systems"*[8] provides an easy overview.

Category theory: In (Roanes-Lozano et al., 1998) the authors put forward a very interesting use of category theory to investiate, by means of some AI techniques, railway interlocking.

ccs: In (Morley, 1991; Morley, 3 4; Morley, 1996; Morley, 1997) Morley uses `ccs` to investigate the well–formedness of the signalling data, ie., the information about rail nets relevant to the switching of rail points, and also studies the use of such data in actual interlocking.

Galois Theory: In (Ingleby and Mitchell, 1992a; Ingleby and Mitchell, 1992b; Ingleby, 1994; Ingleby and Mee, 1995) Michael Ingleby uses classical predicate logic, and, excitingly, also Galois Theory (Ingleby, 1995), to investigate rail net structures for the purposes of structuring proofs of correctness of interlocking schemes. Ingleby proposes to decompose nets into such components that together satisfy a *Galois Connection* criterion — in that way many proofs carry over as lemmas in *Galois Connection* structured theorems.

Constraint Logic Programming: Jimmy Lee and hos colleagues apply constraint–based logic programming methods (Chiu et al., 2002; Chiu et al., 1996) to help the Hong Kong MTR (Metropolitan Transit Railway Corporation)[9] schedule train traffic.

CSP: In (Simpson, 1994; Simpson et al., 1997; Simpson, 1998; Woodcock and McEwan, 2002) `CSP` is used, together with the `CSP` oriented model–checking tool FDR[10], to engineer verified train protection and interlocking systems for the British railway infrastructure. To the present author this work is seminal.

Duration Calculi (DC): In (Zhou and Yu, 1994; Yu and Zhou, 1994; Yu, 1994b) `DC` has been used as a means to study scheduling and stability issues of train traffic. `DC` is slated to be far more used for these purposes than hitherto reported.

Formal Testing: In (Peleska and Siegel, 1996; Peleska, 1996; Peleska, 2002a; Peleska and Tsiolakis, 2002; Peleska, 2002b) formal theories are being established for the actual testing, including test case generation, of safety–critical designs. This work is done for various railway operators (and for the aerospace industry) in Europe.

Interlocking: The switching of rail points (switches, point machines, turn–outs) in groups, from station entry to platform or siding track, is clearly of major safety–related concern: It is also a typical real–time problem that can be computerised. To do so is studied and practiced intensely — as is evidenced by many of the above, and later on below, citations.

In (Cullyer and Wai, 1990; Wong Wai, 1991b; Wong Wai, 1991a; Cullyer and Wai, 1993) Wong uses graphic means to study interlocking. (Holzbacher et al., 1997) uses graph grammars. (Bernardeschi et al., 1996) studies state explosion problems. (Petersen, 1997; Borälv, 1997) apply

[7] `http://www.atelierb.societe.com/index.html`

[8] `http://www.atelierb.societe.com/other_papers/english/using_B/using_B.htm` (French by Pierre Desforges; translated into English by André Danne.)

[9] `www.hkcrystal.com/hiking/mtrkcrmap.htm`

[10] FDR: Failures–Divergence–Refusal are semantic notions of CSP. Formal Systems (Europe) Ltd.: `www.fsel.com/index.html` markets this and related tools and services.

Stålmarck's (model checking) approach, using Prover[11] technology to prove properties of interlocking schemes. (Eriksson, 1997b; Eriksson, 1997a) pursue very similar ideas. (Jackson, 1998) applies process algebraic notions and uses the CMU model checking tool SMV[12] to engineer British Rail interlocking schemes.

Petri Net: It cannot surprise anyone, given the graphical nature and purpose of Petri Nets that it has found widespread use in modelling railway system issues. A large variety of applications can be reported: From studies of railway topologies ((Montigel, 1992; Montigel, 1994)), and interlocking ((Basten et al., 1994; Basten et al., 1995; Billington and Janczura, 1996)) incl. deadlock avoidance, railway stations ((van der Aalst and Odijk, 1995)), via studies of simulation of railway control systems ((zu Hörste, 1999)), and models of train movement ((Decknatel, 1999)), to test case generation for interlocking ((Casaza et al., 1999)), liveness test ((Giua and Seatzu, 2002)), and even an education project ((Berthelot and Petrucci, 2001)).

PVS: In (Skakkebæk, 1994) DC is used, amongst others, to study safety–criticality of railway–road level crossings. In the study models of a DC proof system has been built using PVS.

RAISE: Since the current author, besides being one of the originators of VDM also instigated the development of RAISE, it can come as no wonder that we shall also survey the use of RAISE in connection with railways.

In (YuLin et al., 1994) a Chinese MSc student investigates issues of railway station management. In (Bjørner et al., 1999a; George, 1996) issus of train traffic (global, resp. distributed 'Running Map'–based) scheduling are studied.

In these proceedings (FORMS2003) (Pěnička et al., 2003; Strupchanska et al., 2003), a part result of the EU IST Research and Training Network AMORE on *Algorithmic Methods for Optimising Railways in Europe*, two problems are analysed: What it means to re–schedule train carriages for regular maintenance (service), respectively allocation of railway staff to trains (staff rostering).

In (Bjørner, 2000; Bjørner et al., 2002) the current author suggest a foundation for how to model railway net topologies, respectively the principles and techniques for such domain modelling — irrespective of requirements and actual software design, but as precursors for those development phases.

In (Bjørner, 2003) the present author continues the line of (Bjørner, 2000) and attempts a study of the dynamics of railway nets.

I now come to a series of papers which I believe will be trend–setting: (Haxthausen and Peleska, 2000; Lindegaard et al., 2000; Haxthausen and Gjaldbæk, 2003; Peleska et al., 2000; Haxthausen and Peleska, 2002; Haxthausen and Peleska, 2003b; Haxthausen and Peleska, 2003a) (the last three are treated below, in paragraph **Domain Specific Languages**).

(Haxthausen and Peleska, 2000) concerns the formal development and verification of a distributed railway control system using RAISE. The idea is to start with a domain model of static and dynamic aspects of railway networks, then the safety requirements are defined in terms of that and finally the control system is stepwise developed and verified to satisfy the safety requirements. The RAISE model and verification is generic wrt. the network topology.

(Lindegaard et al., 2000; Haxthausen and Gjaldbæk, 2003) concerns the use of RAISE for the formal modelling and verification of interlocking systems for stations and lines, respectively, at the Danish Railways. These papers build on the same methodological ideas as (Haxthausen and Peleska, 2000), but the control systems are quite different.

Domain Specific Languages: (Peleska et al., 2000; Haxthausen and Peleska, 2002; Haxthausen and Peleska, 2003b; Haxthausen and Peleska, 2003a) *"concerns a development and verification method for railway/tramway control systems based on domain–specific descriptions. The work described in these papers extend previous methodological ideas by providing a domain–specific specification language for railway/tramway control systems. The idea is that for each control system to be developed, application–specific parameters are specified in a domain-specific language, and from this specification a control system is automatically generated and verified to be*

[11] Prover Technology is the name of a Swedish company: http://www.prover.com/.

[12] www-2.cs.cmu.edu/~modelcheck/smv.html

safe. The control components automatically generated from the domain–specific specifications are specifications of the rules of a state transition system that is made executable by a generic interpreter technique. Hence, we generate "executable models". (Haxthausen and Peleska, 2002) (extends (Peleska et al., 2000) with more info) focuses on the domain–specific language, (Haxthausen and Peleska, 2003b) on the automatic generation of control systems from domain-specific descriptions and (Haxthausen and Peleska, 2003a) on the verification and testing issues" — ends the quote.

SPIN: In (Gnesi et al., 2000a; Cimatti et al., 1997; Winter, 2002) the use of the model checking tool SPIN is applied to verification of safety–cricial issues of interlocking.

Statechart: In (Gnesi et al., 1999; Gnesi et al., 2000b) the use of Statechart is studied — together with the informal notation of UML — in order to lend some credibility to the latter, a currently popular approach to software development. Again the problem being studied is that of safety–critical issues of interlocking.

State + Message Sequence Charts: In (Damm and Klose, 2001; Bohn et al., 2002) Statecharts are used together with (versions of) the ITU standard[13] for Message Sequence Chart (MSC) concepts (STD: Symbolic Timing Diagrams, LSC: Live Sequence Charts), and MSC "itself", to verify, model and validate railway signalling schemes.

VDM: In (Hansen, 1994b; Hansen, 1994c; Hansen, 1994a; Hansen, 1996; Hansen, 1998) Kirsten Mark Hansen uses VDM-SL to study interlocking schemes. (Dürr et al., 1995; Ogino and Hirao, 1995a; Ogino and Hirao, 1995b; Agerholm et al., 1998; Ogino T., 1999; Terada, 2002) — four of them from Japan — likewise apply VDM-SL to study safety–critical issues in railway systems.

Z: In (King, 1994) a formalisation of (then) British Rail's Signalling Rules was proposed, while in (Anot, 2000) a study was made of interlocking safety. There are (probably "zillions") additional, and relevant, publications on the use of Z for railway applications — but these must suffice for now.

5. FUTURE RESEARCH

Future research is sometimes based on current problems. Some such problems can perhaps best be undertaken in the context of 'integrating' two or more formal approaches. Some such research may be undertaken in the form of a "Grand ("Man–on–the–Moon") Challenge". The next three subsections deal with the previous three sentences !

5.1 Technical/Scientific Problems

Problem 1: The foremost pressing current technical/scientific problem seems to be that most realistic software developments need combine two or more techniques (languages etc.). Where, for example Petri Nets or Statecharts haven proven very useful for expressing concurrency and transitions, there is no easy "other" formalism in which to express the "contents" of the transition actions. Not one whose semantics and proof rules "fit" the graphics of either of the two techniques just mentioned.

Where, for example RAISE, with RSL, has been used successfully to express action "contents", concurrency and synchronised communication /"rendez–vous") proposals have now been made to supplement this with language constructs for expressing temporal properties: Timing and durations (a la DC): (Xia and George, 1999; Haxthausen and Xia, 2000).

The above, the foremost pressing current technical/scientific problem, is general, not specific to the railway domain.

Problem 2: The secondmost pressing current technical/scientific problem is, in my, undoubtedly prejudiced mind, that the very many otherwise sound approaches to the formal treatment of railway problems do not build on a common understanding of what a railway systems is.

One easily runs the risk that one, say a train scheduling algorithm's developer's conception of a railway net, differs substantially from that of a signalling engineer's conception — with the possible result that automatically generated reschedulings

[13] ITU: Intl.Telecomm.Union.: www.itu.int/. Recommendation Z.120 (11/99) Message Sequence Charts (MSC): www.itu.int/itudoc/itu-t/aap/sg10aap/z120c1.html

are at odds with "corresponding" interlock schemes, and can be so fatally !

In the engineerings based on the natural sciences there are such common domain understandings. These have been and are being provided by their "back–up" science: Physics (mechanics (as from Kepler and Newton), electricity), chemistry, etc., and have been hundreds of years under way.

In Section 5.3 we put forward a proposal for a joint, world–wide *"Grand Challenge"* project that aims at providing a theory of [railway][14] transportation !

5.2 Unifying Theories of Programming

Similar such "integration" of two or more "formal methods" have been and are increasingly being proposed. It seems that a general semantics framework for combining notations have been put forward by Tony Hoare and he JiFeng in (Hoare and Feng, 1997).

Applications of the Hoare/He concept of 'Unifying Theories of Programming' are appearing, for example, in (Butterfield and Woodcock, 2002; Woodcock, 2002; Cavalcanti et al., 2002; Sampaio et al., 2002; Woodcock and Hughes, 2002).

5.3 A Project Proposal

We propose that researchers in university computing as well as transport engineering departments together with scientists and engineers at railway infrastructure providers and at train operators go together around the following possibly 6–10 year joint R&D project: One that establishes a set of commensurate, finely interface "tuned", formally, as well as precisely, but informally, described models of a conceptual railway system, ie., "the railway system". By a conceptual railway system we mean one that designates a class of actual railway systems, that is one to which each of the actual railway systems around the world relate in precisely described ways.

Such a conceptual model would include precisely harmonised descriptions of such "sub–systems", cf. Section 1.2, as for example: (1) statics and dynamics of railway net topologies (Bjørner, 2000; Bjørner, 2003), (2) time table creation based on passenger statistics, (3) scheduling and rescheduling of trains (George, 1996; Bjørner et al., 1999a), (4) train maintenance (Pěnička et al., 2003), (5) crew rostering (Strupchanska et al., 2003), (6) train composition and decomposition (along train routes, and according to seats reserved and to statistics) (Karras and Bjørner, 2002), (7) passenger and freight seat, resp. space inquieries, reserveration (and for freight also tracing), sales etc., (8) railway net development (downsizing and upgrading, net maintenance, etc.), and much more.

We emphasize that we are searching for a model of the railway domain as it is, not as it should be. That is: The "advertised" domain models can then serve as a basis for — hence "normalised" — requirements to computing (monitoring, control & communication) systems. From the requirements one can then develop software. And, provided the software is correct wrt. the "normalised" requirements, we shall conjecture that it is significantly simpler to make sure that otherwise independently developed software is easy to fit safely and securely together !

We cite (Bjørner et al., 1999b; Bjørner et al., 1999c) as a pair of technical reports that suggests domain, respectively requirements models.

We refer to www.imm.dtu.dk/~db/train/train.html, and www.imm.dtu.dk/~db/train/train.ps for HTML and postscript documents which, although a few years old, outline details of such a project — called TRAIN, for: The RAilway INfrastructure project.

Care to join ?

6. CONCLUSION

So what can we conclude ? We formulate the conclusion in the form, first, of questions or conjectures, and, subsequent to all questions, to part answers, respectively comments: (0) What are the trends — in summary ? (1) There is a need for using formal techniques, not only for safety–critical, real–time and embedded software–based systems, but for any related software — also for railways. But who is takning care of the need ? (2) There are a plethora of formal techniques (cum methods, languages and tools) available — and that poses, perhaps, a problem: Which ones to choose ? (3) Among this multitude: Which one are "The Winners !" ? (4) Are these formal techniques being taught sufficiently at universities ? (5) And are these formal techniques being accepted, adopted and adapted by industry ?

[14] — the "Grand Challenge" project easily "carries over" to other transportation modes.

6.1 The Trends

General: The trend is towards increasing replacement of classical control equipment with such which is predominantly controlled by software. Such software need, it is increasingly being mandated — by the transportation system regulatory bodies — to be shown correct by mans of formal techniques. The trend is furthermore to compose such software components into larger systems wherein functionalities spreading across the entire railway planning, development, operation and maintenance spectrum "deliver" data to one another. The complexity of such, in the past, rather mundane, ie., "down–to–earth" applications, thereby increases "exponentially" — furthermore calling for the use of highly professionalised, accredited and certified software houses, support software and software developers.

Along general lines a trend 'Transformation Systems' (Peleska et al., 2000). They transform specifications into semantically consistent executable systems. If transformation has been proven, model checking or theorem proving is no longer relevant or only needed for validation of the generated software.

International Standardisation: The European CENELEC norm requires that applications involving safety with safety–critical integrity levels be implemented using formal techniques. This applies also to railways.

Specific & Personal: The current author — being who he is — cannot refrain, given the opportunity, to point out the trends that he himself, with many colleagues around the world, are pursuing. Using such approaches as are designated by `B, CSP, VDM, RAISE,` and Domain Specific Languages, in order to achieve trustworthy, first abstract, subsequently more concrete models of domains, then requirements, and finally software designs. We refer to the paragraphs above on **B** ((Guiho and Mejia, 1984; Dehbonei and Mejia, 1994a; Dehbonei and Mejia, 1994b; Dehbonei and Mejia, 1995)), **CSP** ((Simpson et al., 1997; Simpson, 1998; Woodcock and McEwan, 2002)), **VDM** ((Hansen, 1994b; Hansen, 1994c; Hansen, 1994a; Hansen, 1996; Hansen, 1998)), **RAISE** ((Bjørner, 2000; Bjørner et al., 2002; Pěnička et al., 2003; Strupchanska et al., 2003; Bjørner, 2003; Haxthausen and Peleska, 2000; Lindegaard et al., 2000; Haxthausen and Gjaldbæk, 2003)), **Domain Specific Languages** ((Peleska et al., 2000; Haxthausen and Peleska, 2002; Haxthausen and Peleska, 2003b; Haxthausen and Peleska, 2003a)) —and the text accompanying these references. **Formal Testing** along the lines of (Peleska and Siegel, 1996; Peleska, 1996; Peleska, 2002a; Peleska and Tsiolakis, 2002; Peleska, 2002b) goes hand–in–hand with the above.

6.2 Caring for the Need

Thus the need for using formal techniques will increase significantly. Those software houses which can demonstrate professionalism in this area will simply replace those which cannot. Already now we see the emergence of a number of European software consultancy & design houses specialising in providing formal techniques–based software to the transportation industry.

6.3 The Multitude

It is too early to give definitive and unique advice on which formal techniques to deply. Surely, for highly concurrent systems `Petri Nets` have shown great use. But `CSP` and `RAISE,` to mention two examples, can also serve well here — they lack the appealing graphics of `Petri Nets,` however, so an integration, a "unification", might be desirabe. For such concurrent systems which are furthermore highly reactive, using `Statechart` in a software design stage seems most reasonable. For major parts of actual domain, requirements and software development, any of the `B, VDM, RAISE` or `Z` approaches will do. `HOL, Isabelle, PVS,` and `SPIN` can, and should be used in conjunction with several of the above techniques and tools — it being noted that there are today reasonably powerful theorem proving assistants or automation provided for `B, CSP, RAISE` and `Z.`

6.4 "Winners ?"

Time will tell ! We are simply far too short into the era of professionally sound, industrially scalable formal techniques and tools. Man will not stop thinking up profoundly new didactic bases for software development.

6.5 University Education & Industry Take–up

An increasing number of graduate students are now being offered courses at most European universities in which one or more of

the techniques and tools covered in this paper play a substantial rôle. There are, however, in my mind, a triplet dichotomy: (i) Students of software engineering very often "know better" than the lecturer/scientists, and, claiming that industry is not using formal techniques, avoid these courses. (ii) The computer science department staff increasingly turn to research in exactly the area of formal techniques. And (iii) increasingly we see the emergence of now dozens of small software houses, all over Europe, industries whose main livelihood depends on their using formal techniques and tools. So the students seem to become "loosers". They look at the large software houses — which will, eventually die out because of fossilisation: They missed the boat on professional, ie., responsible software engineering.

6.6 Closing Remarks

We have provided a bried survey with a long list of supporting references. The references of Section 3 were predominantly to leading books on their subject, while the references of Section 4 were to papers illustrating the application of formal techniques and tools to railway problems.

7. ACKNOWLEDGEMENTS

The author acknowledges the immense benefits he has had from technical/scientific interactions with Søren Prehn and Chris George over the last more then 25, resp. more than 18 years.

8. BIBLIOGRAPHY

The bibliography is extensive — typical of a survey paper. But it is approximately half of the number of entries culled, some years ago, during the FME Rail project mentioned in Section 4.1.

We refer to www.imm.dtu.dk/~db/-fmerail/fmerail and www.imm.dtu.dk/~db/-fmerail/fmerail.ps for a 340 item literature list

References

Abrial, J.-R. (1996). *The B Book: Assigning Programs to Meanings*. Tracts in Theoretical Computer Science. Cambridge University Press, Cambridge, England.

Agerholm, S., Lecoeur, P.-J., and Reichert, E. (1998). Formal Specification and Validation at Work: A Case Study using VDM-SL. In *Proceedings of Second Workshop on Formal Methods in Software Practice*. ACM.

Anot, A. J. (2000). Using Z Specification for Railway Interlocking Safety. Periodica Polytechnica, Transport Engineering Series vol.28, no. 1–2, pp 39–53, Department of Information and Safety Systems Faculty of Electrical Engineering University of Zilina, Vel'ký diel, Zilina 010 26, Slovak Republic.

Basten, T., Bol, R., and Voorhoeve, M. (1994). Simulating and Analyzing Railway Interlocking in ExSpect. Technical Report 94-37, Department of Computing Science, Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands. See (Basten et al., 1995).

Basten, T., Bol, R., and Voorhoeve, M. (Fall 1995). Simulating and analyzing railway interlockings in ExSpect. *IEEE Parallel & Distributed Technology: Systems & Applications*, 3(3):50–62.

Bernardeschi, C., Fantechi, A., Gnesi, S., and Mongardi, G. (1996). Proving safety properties for embedded control systems. In Hlawiczka, A., Silva, J., and L.Simoncini, editors, *Dependable Computing — EDCC–2. Second European Dependable Computing Conference Proceedings; Taormina, Italy*, pages 321–32. Springer–Verlag; Berlin, Germany.

Berthelot, G. and Petrucci, L. (2001). Specification and validation of a concurrent system: an educational project. *International Journal on Software Tools for Technology Transfer*, 3(4):372–381. Special section on the practical use of high-level Petri Nets.

Billington, J. and Janczura, C. (1996). Removing Deadlock from a Railway Network Specification. In *Australian Engineering Mathematics Conference (AEMC'96)*, pages 193–200, Sydney, Australia.

Bjørner, D. (2000). Formal Software Techniques in Railway Systems. In Schnieder, E., editor, *9th IFAC Symposium on Control in Transportation Systems*, pages 1–12, Technical University, Braunschweig, Germany. VDI/VDE-Gesellschaft Mess– und Automatisieringstechnik, VDI-Gesellschaft für Fahrzeug– und Verkehrstechnik. Invited talk. *Postscript document*[15].

[15] www.imm.dtu.dk/~db/documents/2ifacpaper.ps

Bjørner, D. (2003). Dynamics of Railway Nets: On an Interface between Automatic Control and Software Engineering. In *CTS2003: 10th IFAC Symposium on Control in Transportation Systems*, Oxford, UK. Elsevier Science Ltd. Symposium held at Tokyo, Japan. Editors: S. Tsugawa and M. Aoki.

Bjørner, D. (2003–2004). *The SE Book: Principles and Techniques of Software Engineering*, volume I: Abstraction & Modelling (750 pages), II: Descriptions and Domains (est.: 500 pages), III: Requirements, Software Design and Management (est. 450 pages). [Publisher currently (March 2003) being negotiated]. *TheSEBook Home Page*[16].

Bjørner, D., George, C., and Prehn, S. (1999a). *Scheduling and Rescheduling of Trains*, chapter 8, pages 157–184. *Industrial Strength Formal Methods in Practice*, Eds.: Michael G. Hinchey and Jonathan P. Bowen. FACIT, Springer–Verlag, London, England. *Postscript document*[17].

Bjørner, D., George, C. W., and Prehn, S. (2002). Computing Systems for Railways — A Rôle for Domain Engineering. Relations to Requirements Engineering and Software for Control Applications. In *Integrated Design and Process Technology. Editors: Bernd Kraemer and John C. Petterson*, page 26 pages, P.O.Box 1299, Grand View, Texas 76050-1299, USA. Society for Design and Process Science. *PS*[18] *PDF*[19].

Bjørner, D. and Jones, C. (1978). *The Vienna Development Method: The Meta-Language*, volume 61 of *LNCS*. Springer–Verlag.

Bjørner, D. and Jones, C. (1982). *Formal Specification and Software Development*. Prentice-Hall.

Bjørner, D., Prehn, S., and George, C. W. (1999b). Formal Models of Railway Systems: Domains. Technical report, Dept. of IT, Technical University of Denmark, Bldg. 344, DK–2800 Lyngby, Denmark. Presented at the FME Rail Workshop on Formal Methods in Railway Systems, FM'99 World Congress on Formal Methods, Toulouse, France. Avaliable on CD ROM. *Postscript document*[20].

Bjørner, D., Prehn, S., and George, C. W. (1999c). Formal Models of Railway Systems: Requirements. Technical report, Dept. of IT, Technical University of Denmark, Bldg. 344, DK–2800 Lyngby, Denmark. Presented at the FME Rail Workshop on Formal Methods in Railway Systems, FM'99 World Congress on Formal Methods, Toulouse, France. Avaliable on CD ROM. *Postscript document*[21].

Bohn, J., Damm, W., Klose, J., Moik, A., and Wittke, H. (2002). Modeling and Validating Train System Applications Using Statemate and Live Sequence Chats. In Ehrig, H., Krämer, B. J., and Ertas, A., editors, *Proceedings of IDPT2002 - Integrated Design and Process Technology*. Society for Design and Process Science.

Borälv, A. (1997). A Fully Automated Approach for Proving Safety Properties in Interlocking Software Using Automatic Theorem-Proving. In S. Gnesi and D. Latella, editor, *Proceedings of the Second International ERCIM Workshop on Formal Methods for Industrial Critical Systems*, pages 39–62. Consiglio Nazionale Ricerche, Pisa.

Börger, E., editor (1995). *Specification and Validation Methods*. Oxford University Press.

Börger, E., Gargantini, A., and Riccobene, E., editors (2003). *Abstract State Machines 2003–Advances in Theory and Applications*, volume 2589 of *Lecture Notes in Computer Science*. Springer-Verlag. This is the Proc. 10th Int. ASM Workshop (Taormina March 2003).

Börger, E., Päppinghaus, P., and Schmid, J. (2000). Report on a Practical Application of ASMs in Software Design. In Y. Gurevich and P. Kutter and M. Odersky and L. Thiele, editor, *Abstract State Machines: Theory and Applications*, volume 1912 of *LNCS*, pages 361–366. Springer-Verlag.

Börger, E. and Stärk, R., editors (2003). *Abstract State Machines. A Method for High-Level System Design and Analysis*. Springer-Verlag ISBN 3-540-00702-4.

Bowen, J. (1993). Formal methods in safety-critical standards. In *Software Engineering Standards Symposium (SESS'93)*, pages 168–177. Brighton, UK, IEEE Computer Society Press.

[16] www.imm.dtu.dk/~/TheSEBook

[17] www.it.dtu.dk/~db/racosy/scheduling.ps

[18] www.imm.dtu.dk/~db/documents/idpt.ps

[19] www.imm.dtu.dk/~db/documents/idpt.pdf

[20] www.imm.dtu.dk/~db/racosy/domain.ps

[21] www.imm.dtu.dk/~db/racosy/requirements.ps

Bowen, J. and Hinchey, M. (1994). Formal methods and safety-critical standards. *IEEE Computer*, pages 69–71.

Bowen, J. and Stavridou, V. (1992). Formal methods and software safety. In Frey, H., editor, *Safety of Computer Control Systems 1992 (SAFECOMP'92)*, pages 93–98. Pergamon Press.

Bowen, J. and Stavridou, V. (1993). The industrial take-up of formal methods in safety-critical and other areas: A perspective. In Woodcock, J. and Larsen, P., editors, *FME'93: Industrial-Strength Formal Methods*, pages 183–195. Formal Methods Europe, Springer-Verlag. Lecture Notes in Computer Science 670.

Bowen, J. P. and Hinchey, M. G. (1995a). Seven more myths of formal methods. *IEEE Software*, 12(3):34–41.

Bowen, J. P. and Hinchey, M. G. (1995b). Ten Commandments of Formal Methods. *IEEE Computer*, 28(4):56–62.

Butler, R. W., Caldwell, J. L., Carreno, V. A., Holloway, C. M., Miner, P. S., and Vito, B. L. D. (1995). Nasa langley's research and technology transfer program in formal methods. In *Tenth Annual Conference on Computer Assurance (COMPASS 95)*. Gaithersburg, MD. (expanded version available from `atb-www.larc.nasa.gov/fm.html`).

Butterfield, A. and Woodcock, J. (2002). Semantics of Prialt in Handel-C. In *Concurrent Systems Engineering, Proceedings of the Conference on Communicating Processing Architectures*. IOS Press.

Caldwell, J. L. (1996). Formal methods technology–transfer: a view from nasa. In Gnesi, S. and Latella, D., editors, *Proceedings of the ERCIM Workshop on Formal Methods for Industrial Critical Systems*, Oxford, England.

Casaza, A., Comini, D., Morzenti, A., Pradella, M., Pietro, P. S., and Schreiber, F. (1999). Interlocking: Specification and Test Case Generation for the Safety Kernel of the Naples Subway. In Montigel, M., editor, *FME Rail Workshop # 3*, volume # 3 of *FME Rail Workshop; St. Pölten*. FME: Formal Methods Europe, Fachhochschulstudiengang St. Pölten, Herzogenburgerstr. 68, A-3100 St. Pölten, Austria; Phone: +43 2742 313 228, Fax: +43 2742 313 229.

Cavalcanti, A., Sampaio, A., and Woodcock, J. (2002). Refinement of Actions in Circus. In *Proceedings of REFINE'2002*, Electronic Notes in Theoretical Computer Science. Invited Paper.

Chen, Z., Wang, J., and Zhou, C. (1994a). A Design Approach of Hybrid Control Systems. Research Report 27, UNU/IIST, P.O.Box 3058, Macau.

Chen, Z., Wang, J., and Zhou, C. (1994b). An Abstraction of Hybrid Control Systems. Research Report 26, UNU/IIST, P.O.Box 3058, Macau.

Chiu, C., Chou, C., Lee, J., Leung, H., and Leung, Y. (1996). A constraint-based interactive train rescheduling tool. In *Proceedings of the Second International Conference on Principles and Practice of Constraint Programming (LNCS 1118)*, pages 104–118. Springer Verlag.

Chiu, C., Chou, C., Lee, J., Leung, H., and Leung, Y. (2002). A constraint-based interactive train rescheduling tool. *Constraints*, 7(2):139–174.

Cimatti, A., Giunchiglia, F., Mongardi, G., Romano, D., Torielli, F., and Traverso, P. (1997). Model Checking Safety Critical Software with SPIN: an Application to a Railway Interlocking System. In *SPIN NEWS Letter Nr. 16*.

Cleland, G. and MacKenzie, D. (1995). Inhibiting factors, market structure and the industrial uptake of formal methods. In *Workshop on Industrial-Strength Formal Specification Techniques*, pages 46–60. IEEE Computer Society.

Cullyer, J. and Wai, W. (Feb. 1993). Application of formal methods to railway signalling - a case study. *Computing & Control Engineering Journal*, 4(1):15–22.

Cullyer, W. J. and Wai, W. (1990). A formal approach to railway signalling. In *Compass '90: 5th Annual Conference on Computer Assurance*, pages 102–108, Gaithersburg, Maryland. National Institute of Standards and Technology.

Damm, W. and Klose, J. (2001). Verification of a Radio-based Signaling System Using the Statemate Verification Environment. *Formal Methods in System Design*, 19(2).

Decknatel, G. (1999). Modelling Train Movement with Hybrid Petri Nets. In Bjørner, D. and Fahlén, M., editors, *FME Rail Workshop # 4*, volume # 4 of *FME Rail Workshop; Stockholm, Sweden*. FME: Formal Methods Europe, Banverket, Falun, Sweden. Univ. Braunschweig, Germany. .

Dehbonei, B. and Mejia, L.-F. (1994a). Formal development of software in safety-critical railway systems. In Murthy, B. T. K. S., Mellitt, C. A., Brebbia, G., and Sciutto, S., editors, *Railway Operations*, volume 2, pages 213–219. COMPRAIL94, Computational Mechanics Publications.

Dehbonei, B. and Mejia, L.-F. (1994b). Formal methods in the railways signalling industry. In Naftalin, M., Denvir, T., and Bertran, M., editors, *FME '94: Industrial Benefit of Formal Methods. Second International Symposium of Formal Methods Europe. Proceedings; Barcelona, Spain*, pages 26–34. Springer-Verlag, Berlin, Germany; Lecture Notes in Computer Science LNCS.

Dehbonei, B. and Mejia, L.-F. (1995). Formal development of safety-critical software systems in railway signaling. In Hinchey, M. G. and Bowen, J. P., editors, *Applications of Formal Methods*, Series in Computer Science, pages 227–252. Prentice Hall International.

Dürr, E., Plat, N., and de Boer, M. (1995). CombiCom: Tracking and Tracing Rail Traffic using VDM++. In Hinchey, M. G. and Bowen, J. P., editors, *Applications of Formal Methods*, pages 203–225. Prentice-Hall International. ISBN 0-13-3-366949-1.

Eriksson, L. (1997a). Formal verification of railway interlockings. Technical Report 1997:4, Swedish National Rail Administration. Previously published in Swedish as report 1997:2

Eriksson, L. (1997b). Formalising railway interlocking requirements. Technical Report 1997:3, Swedish National Rail Administration. Previously published in Swedish as report 1997:1.

Fahlén, M., editor (1998). *FME Rail Workshop # 3*, volume 3 of *FME Rail Seminars*, Falun, Sweden. FME: Formal Methods Europe, Banverket. ESSI Project 26538. Workshop venue: Stockholm, Sweden. Organised by Banverket (Swedish Rail's Infrastructure Division), Falun, Sweden.

Fitzgerald, J. and Larsen, P. G. (1997). *Developing Software using VDM–SL*. Cambridge University Press, The Edinburgh Building, Cambridge CB2 1RU, England.

George, C. (1996). A Theory of Distributed Train Rescheduling. In Gaudel, M.-C. and Woodcock, J., editors, *FME'96: Industrial Benefit and Advances in Formal Methods*, pages 499–517. Springer-Verlag.

George, C., Haff, P., Haxthausen, A., Havelund, K., Milne, R., Nielsen, C.B., Prehn, S., and Wagner, K.R. (1992). *The RAISE Specification Language*. The BCS Practitioners Series. Prentice-Hall International.

George, C., Haxthausen, A., Hughes, S., Milne, R., Prehn, S., and Pedersen, J. S. (1995). *The RAISE Method*. The BCS Practitioner Series. Prentice-Hall, Hemel Hampstead, England.

Giua, A. and Seatzu, C. (2002). Liveness Enforcing Supervisors for Railway Networks Using ES^2PR Petri Nets. In *Sixth International Workshop on Discrete Event Systems*, Zaragoza, Spain. WODES'02.

Gnesi, S., Latella, D., Lenzini, G., Abbaneo, C., Amendola, A., and Marmo, P. (2000a). An automatic SPIN validation of a safety critical railway control system. In *International Conference on Dependable Systems & Networks*, pages 119–124. IEEE Computer Society Press.

Gnesi, S., Latella, D., and Massink, M. (1999). Model Checking UML Statechart Diagrams using JACK. In *Proc. Fourth IEEE International Symposium on High Assuarance Systems Enginering*. IEEE Press.

Gnesi, S., Latella, D., and Massink, M. (2000b). A stochastic extension of a behavioural subset of UML statechart diagrams. In *5th IEEE International Symposium on High Assuarance Systems Enginering*. IEEE.

Gordon, M. J. C. and Melham, T. F., editors (1993). *Introduction to HOL: A Theorem Proving Environment for Higher–Order Logic*. Cambridge University Press, Cambridge, UK.

Grégoire, J.–C., Holzmann, G. J. and Peled, D., editor (1997). *The SPIN Verification System*, volume 32 of *DIMACS series*. American Mathematical Society. ISBN 0-8218-0680-7, 203p.

Guiho, G. and Mejia, L.-F. (1984). Operational safety critical software methods in railways. In Anon, editor, *IFIP Transactions A (Computer Science and Technology),*, pages 262–9. IFIP World Congress, Hamburg, Germany.

Gurevich, Y. and Kutter, P. and Odersky. M and Thiele, L, editor (2000). *Abstract State Machines: Theory and Applications*, volume 1912 of *LNCS*. Springer-Verlag.

Gurevich, Y., Kutter, P., Odersky, M., and Thiele, L., editors (2000). *Abstract State Machines – ASM 2000, International Workshop*

on Abstract State Machines, Monte Verita, Switzerland, Local Proceedings, number 87 in TIK-Report. Swiss Federal Institute of Technology (ETH) Zurich.

Hall, A. (1990). Seven Myths of Formal Methods. *IEEE Software*, 7(5):11–19.

Hansen, K. (1994a). Formalising railway interlocking systems. In *Nordic Seminar on Dependable Computing Systems*, pages 83–94, Technical University of Denmark. Department of Computer Science.

Hansen, K. (1994b). Validation of a railway interlocking model. In Naftalin, M., Denvir, T., and Bertran, M., editors, *FME '94: Industrial Benefit of Formal Methods. Second International Symposium of Formal Methods Europe. Proceedings; Barcelona, Spain*, pages 582–601. Springer-Verlag, Berlin, Germany; Lecture Notes in Computer Science LNCS.

Hansen, K. (1996). *Linking Safety Analysis to Safety Requirements*. PhD thesis, Department of Computer Science, Technical University of Denmark, Building 344, DK-2800 Lyngby, Denmark.

Hansen, K. (1998). Formalising Railway Interlocking Systems. In *FME Rail Workshop 2*, ScanRail Consult, Signalling Assessment, Pilestræde 58/6, DK–1112 Copenhagen K, Denmark. Danish National Railway Agency. In: (Woodcock, 1998). Simulation, end-user validation.

Hansen, K. M. (1994c). Validation of a railway interlocking model. *Lecture Notes in Computer Science*, 873.

Hansen, M. and Zhou, C. (1992). Semantics and Completeness of Duration Calculus. In de Bakker, J., Huizing, C., de Roever, W.-P., and Rozenberg, G., editors, *Real-Time: Theory in Practice, REX Workshop*, volume 600 of *Lecture Notes in Computer Science*, pages 209–225. Springer Verlag.

Harel, D. and Naamad, A. (1996). The STATEMATE Semantics of Statecharts. *ACM Transactions on Software Engineering and Methodology*, 5(4):293–333.

Harel, D. and Politi, M. (1998). *Modelling Reactive Systems with Statecharts: The Statemate Approach*. McGraw Hill. 258 pages.

Haxthausen, A. and Gjaldbæk, T. (2003). Modelling and Verification of Interlocking Systems for Railway Lines. In *10th IFAC Symposium on Control in Transportation Systems, Tokyo, Japan*.

Haxthausen, A. and Peleska, J. (1998a). Formal Development and Verification of a Distributed Railway Control System. In *FME Rail Workshop 1*, DK–2800 Lyngby Denmark; P.O.Box 340440, D–28334 Bremen, Germany. Dept. of IT, Techn. Univ. of Denmark; BISS, Bremen Univ. See also: (Haxthausen and Peleska, 1998b). In: (Larsen, 1998).

Haxthausen, A. and Peleska, J. (1998b). Formal Development and Verification of a Distributed Railway Control System. In *FME Rail Workshop 2*, DK–2800 Lyngby Denmark; P.O.Box 340440, D–28334 Bremen, Germany. Dept. of IT, Techn. Univ. of Denmark; BISS, Bremen Univ.

Haxthausen, A. and Peleska, J. (2000). Formal Development and Verification of a Distributed Railway Control System. *IEEE Transaction on Software Engineering*, 26(8):687–701.

Haxthausen, A. and Peleska, J. (2003a). Automatic Verification, Validation and Test for Railway Control Systems based on Domain-Specific Descriptions. In *10th IFAC Symposium on Control in Transportation Systems, Tokyo, Japan*.

Haxthausen, A. and Peleska, J. (2003b). Generation of Executable Railway Control Components from Domain-Specific Descriptions. In *Proceedings of the Symposium on Formal Methods for Railway Operation and Control Systems (FORMS'2003)*. L'Harmattan Hongrie. To appear.

Haxthausen, A. and Xia, Y. (2000). Linking DC together with TRSL. In *Proceedings of 2nd International Conference on Integrated Formal Methods (IFM'2000), Schloss Dagstuhl, Germany, November 2000*, number 1945 in Lecture Notes in Computer Science, pages 25–44. Springer-Verlag.

Haxthausen, A. E. and Peleska, J. (2002). A Domain Specific Language for Railway Control Systems. In *Sixth Biennial World Conference on Integrated Design and Process Technology, (IDPT2002), Pasadena, California*, P.O.Box 1299, Grand View, Texas 76050-1299, USA. Society for Design and Process Science.

Henzinger, T. A. (1996). The Theory of Hybrid Automata. In *LICS: 11th Annual Symposium on Logic in Computer Science*, pages 278–292. IEEE, IEEE Computer Society Press. An extended version appeared in Verification of Digital and Hybrid Systems (M.K. Inan, R.P. Kurshan, eds.), NATO ASI Series F: Computer and Systems Sciences, Vol. 170, Springer-Verlag, 2000, pp. 265-292.

Henzinger, T. A., Ho, P.-H., and Wong-Toi, H. (1997a). HyTech: A Model Checker for Hybrid Systems. In *Ninth International Conference on Computer-Aided Verification*, volume 1254, pages 460–463. CAV: Computer Aided Verification, Springer–Verlag.

Henzinger, T. A., Ho, P.-H., and Wong-Toi, H. (1997b). HyTech: A Model Checker for Hybrid Systems. *Software Tools for Technology Transfer*, 1:110–122. Preliminary version appeared in (Henzinger et al., 1997a).

Hinchey, M. G. and Bowen, J. P., editors (1995). *Applications of Formal Methods*. Prentice Hall. ISBN 0-13-366949-1.

Hoare, C. (1978). Communicating sequential processes. *Communications of the ACM*, 21(8).

Hoare, C. (1985). *Communicating Sequential Processes*. Prentice-Hall International.

Hoare, C. and Feng, H. J. (1997). *Unifying Theories of Programming*. Prentice Hall.

Holzbacher, A., Perin, M., and Sudholt, M. (1997). Modeling railway control systems using graph grammars: a case study. In Garlan, D. and Metayer, D. L., editors, *Coordination Languages and Models. Second International Conference COORDINATION '97*.

Holzmann, G. (1991). *Design and Validation of Computer Protocols*. Prentice-Hall, Englewood Cliffs, New Jersey.

Holzmann, G. (2003). *The Spin Model Checker, Primer and Reference Manual*. Addison-Wesley, Reading, Massachusetts.

Hung, D. V. and Wang, J. (1994). On The Design of Hybrid Control Systems Using Automata Model. Research Report 35, UNU/IIST, P.O.Box 3058, Macau. Published in V. Chandru and V. Vinay (Eds.) *Foundations of Software Technology and Theoretical Computer Science* (FST&TCS16), LNCS 1180, Springer-Verlag, Dec 1996, pp. 156–167.

Ingleby, M. (1994). Safety properties of a control network: local and global reasoning in machine proof. In *Proceedings of Real Time Systems*. Paris.

Ingleby, M. (1995). A galois theory of local reasoning in control systems with compositionality. In *Proceedings of Mathematics of Dependable Systems*. Oxford UP (UK).

Ingleby, M. and Mee, D. (1995). A calculus of hazard for railway signalling. In *Workshop on Industrial-Strength Formal Specification Techniques (Cat. No.95TH8051); Boca Raton, FL, USA*, pages 146–58. IEEE Comput. Soc. Press, Los Alamitos, CA, USA.

Ingleby, M. and Mitchell, I. (1992a). Proving Safety of a Railway Signaling System Incorporating Geographic Data. In Frey, H., editor, *SAFECOM'92 Conference Proceedings of IFAC*, pages 129–134, Zürich (CH). Pergamon Press.

Ingleby, M. and Mitchell, I. (1992b). Proving safety of a railway signalling system incorporating geographic data. *SAFECOMP 1992: Safety of Computer Control Systems 1992*, pages 129–134.

Jackson, D. (1998). Verification of BR Interlocking. In *FME Rail Workshop 2*, Bath, England. Praxis Critical Systems. In (Woodcock, 1998).

Jensen, K. (1985, revised and corrected second version: 1997). *Coloured Petri Nets*, volume 1: Basic Concepts (234 pages + xii), Vol. 2: Analysis Methods (174 pages + x), Vol. 3: Practical Use (265 pages + xi) of *EATCS Monographs in Theoretical Computer Science*. Springer–Verlag, Heidelberg.

Karras, P. and Bjørner, D. (2002). Train composition and decomposition: From passenger statistics to schedules. Technical report, Informatics and Mathematical Modelling, Building 322, Richard Petersens Plads, Technical University of Denmark, DK–2800 Kgs.Lyngby, Denmark. This is a report in the AMORE project series: (Strupchanska et al., 2003; Pěnička et al., 2003). *Postscript document*[22].

Kelly, J. (1995). Formal methods, specification and verification guidebook for software an computer systems – planning and technology insertion. Technical Report NASA-GB-002-95 (Release 1.0), NASA, Washington, DC 20546, USA.

Kelly, J. (1996). Formal methods, specification and verification guidebook for software and computer systems – a practitioner's companion. Technical Report Draft 2.0, NASA, Washington, DC 20546, USA.

King, T. (1994). Formalising British Rail's Signalling Rules. In M. Naftalin, T. Denvir, M. B., editor, *FME'94: Industrial Benefit of Formal Methods*, pages 45–54. Springer-Verlag.

[22] `www.imm.dtu.dk/~db/amore/amoretcad.ps`

Larsen, P., editor (1998). *FME Rail Workshop #1*, volume 1 of *FME Rail Seminars*, Forskerparken, DK–6000 Odense, Denmark. FME: Formal Methods Europe, IFAD. ESSI Project 26538. Workshop venue: Breukelen, The Netherlands. Organised by Origin Nederland, a member of the Philips group of companies, P.O.Box 1444, NL–3430 BK Nieuwegein, The Netherlands.

Larsen, P. G., Hansen, B. S., Plat, H. B. N., Toetenel, H., Andrews, D. J., Dawes, J., Parkin, G., et al. (1996). Information technology — Programming languages, their environments and system software interfaces — Vienna Development Method — Specification Language — Part 1: Base language.

Lecomte, T. and Larsen, P. G., editors (1999). *FME Rail Workshop #5*, volume 5 of *FME Rail Seminars*. FME: Formal Methods Europe, Springer Verlag. ESSI Project 26538. Workshop venue: Toulouse, France. Organised as part of FM'99: World Congress of Formal Methods.

Lindegaard, M. P., Viuf, P., and Haxthausen, A. (2000). Modelling Railway Interlocking Systems. In *Proceedings of the 9th IFAC Symposium on Control in Transportation Systems 2000, June 13-15, 2000, Braunschweig, Germany*, pages 211–217.

Liu, S., Stavridou, V., and Dutertre, B. (1995). The practice of formal methods in safety critical systems. *Journal of Systems and Software*, 28:77–87.

Liu, Z., Ravn, A., Sørensen, E., and Zhou, C. (1993). A probabilistic duration calculus. In Kopetz, H. and Kakuda, Y., editors, *Responsive Computer Systems*, volume 7 of *Dependable Computing and Fault-Tolerant Systems*, pages 30–52. Springer Verlag Wien New York.

Milner, R. (1980). *Calculus of Communication Systems*, volume 94 of *LNCS*. Springer–Verlag.

Milner, R. (1989). *Communication and Concurrency*. C.A.R. Hoare Series in Computer Science. Prentice Hall.

Milner, R. (1999). *Communicating and Mobile Systems: The π–Calculus*. Cambridge University Press. 161 pages, Amazon price:US $ 28.00.

Montigel, M. (1992). Formal representation of track topologies by double vertex graphs. In *Proceedings of Railcomp 92 held in Washington DC, Computers in Railways 3*, volume 2: Technology. Computational Mechanics Publications.

Montigel, M. (1994). *Modellierung und Gewährleistung von Abhängigkeiten in Eisenbahnsicherungsanlagen*. PhD thesis, ETH: Swiss Federal Institute of Technology, ETH Honggerberg, CH-8093 Zürich, Swtizerland.

Montigel, M., editor (1999). *FME Rail Workshop #4*, volume 4 of *FME Rail Seminars*, Herzogenburgerstr. 68, A-3100 St. Pölten, Austria. FME: Formal Methods Europe, Fachhochschulstudiengang St. Pölten. ESSI Project 26538. Workshop venue: St. Pölten, Austria. Organised by Fachhochschulstudiengang St. Pölten and Alcatel, Austria.

Morley, M. (1991). Modelling British Rail's Interlocking Logic: Geographic Data Correctness. Technical Report ECS-LFCS-91-186, University of Edinburgh.

Morley, M. (1993–4). Safety in railway signalling data: A behavioural analysis. In Joyce, J. and Seger, C., editors, *Proc. 6th annual workshop on higher order logic and its applications, Vancouver, 4-6 August*, pages 465–474. Springer–Verlag Lecture Notes in Computer Science, Vol.780.

Morley, M. (1996). *Safety Assurance in Interlocking Design*. PhD thesis, University of Edinburgh.

Morley, M. (1997). Safety-level communication in railway interlockings. *Science of Computer Programming*, 29(1-2):147–170.

Nipkow, T., Paulson, L. C., and Wenzel, M. (2002). *Isabelle/HOL, A Proof Assistant for Higher-Order Logic*, volume 2283 of *LNCS*. Springer-Verlag, Heidelberg, Germany.

Ogino, T. and Hirao, Y. (1995a). Formal methods and their applications to safety-critical systems of railways. *Quarterly Report of RTRI. Ken-yusha for Railway Technical Research Institute of Japan, Tokyo*, 36(4).

Ogino, T. and Hirao, Y. (1995b). Formal Methods and their Applications to Safety-Critical Systems of Railways. *Quarter Review of RTRI (Japanese Railway Technical Research Institute)*, 36(4):198–203.

Ogino T., H. Y. (1999). Interest in Formal Methods from Japanese Perspective. In Larsen, P. G., editor, *Proceeding of FMERail Workshop number 5*, Toulouse. FM99.

Owre, S., Shankar, N., Rushby, J. M., and Stringer-Calvert, D. W. J. (1999a). *PVS Language Reference*. Computer Science Laboratory, SRI International, Menlo Park, CA.

Owre, S., Shankar, N., Rushby, J. M., and Stringer-Calvert, D. W. J. (1999b). *PVS System Guide*. Computer Science Laboratory, SRI International, Menlo Park, CA.

Peleska, J. (1996). Test Automation for Safety-Critical Systems: Industrial Application and Future Developments. In Gaudel, M.-C. and Woodcock, J., editors, *FME'96: Industrial Benefit and Advances in Formal Methods*, pages 39–59. Springer-Verlag.

Peleska, J. (2002a). Formal Methods for Test Automation - Hard Real-Time Testing of Controllers for the Airbus Aircraft Family. In Kraemer, B. and Petterson, J. C., editors, *Sixth Biennial World Conference on Integrated Design & Process Technology (IDPT2002)*, P.O.Box 1299, Grand View, Texas 76050-1299, USA. Society for Design and Process Science.

Peleska, J. (2002b). Hardware/Software Integration Testing for the new Airbus Aircraft Families. In Schieferdecker, I., König, H., and Wolisz, A., editors, *Testing of Communicating Systems XIV. Application to Internet Technologies and Services*, pages 335–351, The Netherlands. Kluwer Academic Publishers.

Peleska, J., Baer, A., and Haxthausen, A. (2000). Towards Domain-Specific Formal Specification Languages for Railway Control Systems. In *Proceedings of the 9th IFAC Symposium on Control in Transportation Systems 2000, June 13-15, 2000, Braunschweig, Germany*, pages 147–152.

Peleska, J. and Siegel, M. (1996). From testing theory to test driver implementation. In Gaudel, M.-C. and Woodcock, J., editors, *FME '96: Industrial Benefit and Advances in Formal Methods. Third International Symposium of Formal Methods Europe. Proceedings; Oxford, UK*, pages 538–56. Springer-Verlag; Berlin, Germany.

Peleska, J. and Tsiolakis, A. (2002). Automated Integration Testing for Avionics Systems. In *3rd ICSTEST, International Conference on Software Testing*, Düsseldorf. Anon.

Pěnička, M., Strupchanska, A. K., and Bjørner, D. (2003). Train maintenance routing. In *FORMS2003: Symposium on Formal Methods for Railway Operation and Control Systems*. L'Harmattan Hongrie. Conf. held at Techn.Univ. of Budapest, Hungary. Editors: G. Tarnai and E. Schnieder, Germany.

Petersen, J. (1997). Formal Requirement Verification of a Swedish Railway Interlocking System. Technical Report IT-TR: 1997-005, Technical University of Denmark, Department of Information Technology.

Petri, C. (1962). *Kommunikation mit Automaten*. PhD thesis, Univ. Bonn, Germany.

Reisig, W. (1985). *Petri Nets: An Introduction*, volume 4 of *EATCS Monographs in Theoretical Computer Science*. Springer Verlag.

Reisig, W. (1992). *A Primer in Petri Net Design*. Springer Verlag. 120 pages.

Reisig, W. (1998). *Elements of Distributed Algorithms: Modelling and Analysis with Petri Nets*. Springer Verlag. 400 pages.

Roanes-Lozano, E., Laita, L. M., and Roanes-Macias, E. (1998). An application of an AI methodology to railway interlocking systems using computer algebra. *Lecture Notes in Computer Science*, 1416:687.

Roscoe, A. (1997). *Theory and Practice of Concurrency*. Prentice–Hall.

Rushby, J. (1993). Formal methods and the certification of critical systems. Technical Report SRI-CSL-93-7, Computer Science Laboratory, SRI International, Menlo Park, CA., USA. Also issued under the title "Formal Methods and Digital Systems Validation for Airborne Systems" as NASA Contractor Report 4551, December 1993. See also (Rushby, 1995).

Rushby, J. (1995). Formal methods and their role in the certification of critical systems. Technical Report SRI-CSL-95-1, Computer Science Laboratory, SRI International, Menlo Park, CA. Also available as NASA Contractor Report 4673, August 1995, and to be issued as part of the FAA Digital Systems Validation Handbook (the guide for aircraft certification). See also (Rushby, 1993).

Sampaio, A., Woodcock, J., and Cavalcanti, A. (2002). Refinement in Circus. In Eriksson, L. and Lindsay, P., editors, *FME 2002: Formal Methods - Getting IT Right*, volume 2391 of *Lecture Notes in Computer Science*, pages 451–470. Springer-Verlag.

Sangiorgio, D. and Walker, D. (2001). *The π–Calculus*. Cambridge University Press. Amazon price:US $ 90.00.

Schneider, S. (2000). *Concurrent and Real–time Systems — The CSP Approach*. Worldwide Series in Computer Science. John Wiley & Sons, Ltd., Baffins Lane, Chichester, West Sussex PO19 1UD, England.

Shankar, N., Owre, S., and Rushby, J. M. (1993). *PVS Tutorial*. Computer Science Laboratory, SRI International, Menlo Park, CA. Also appears in Tutorial Notes, *Formal Methods Europe '93: Industrial-Strength Formal Methods*, pages 357–406, Odense, Denmark, April 1993.

Shankar, N., Owre, S., Rushby, J. M., and Stringer-Calvert, D. W. J. (1999). *PVS Prover Guide*. Computer Science Laboratory, SRI International, Menlo Park, CA.

Simpson, A. (1994). A formal specification of an automatic train protection system. In M. Naftalin, T. Denvir, M. B., editor, *FME'94: Industrial Benefit of Formal Methods*, pages 602–617. Springer-Verlag.

Simpson, A. (1998). Model Checking for Interlocking Safety. In *FME Rail Workshop 2*, Parks Road, Oxford OX1 3QG, UK. Oxford Univ., Computing Lab. In: (Woodcock, 1998). Safety.

Simpson, A., Woodcock, J., and Davies, J. (1997). The mechanical verification of Solid State Interlocking geographic data. In Groves, L. and Reeves, S., editors, *Proceedings of Formal Methods Pacific*, pages 223–242, Wellington, New Zealand. Springer–Verlag.

Skakkebæk, J. U. (1994). *A Verification Assistant for a Real-Time Logic*. PhD thesis, Department of Computer Science, Technical University of Denmark, Lyngby, Denmark. Available as Technical Report ID-TR: 1994-150.

Strupchanska, A. K., Pěnička, M., and Bjørner, D. (2003). Railway staff rostering. In *FORMS2003: Symposium on Formal Methods for Railway Operation and Control Systems*. L'Harmattan Hongrie. Conf. held at Techn.Univ. of Budapest, Hungary. Editors: G. Tarnai and E. Schnieder, Germany.

Terada, N. (2002). Integrity analysis of digital ATC track database with automatic proofs. Technical report, Railway Techn. Research Inst., Tokyo, Japan. Presented at FLoC'02: VDM Workshop, Copenhagen, July 2002.

Thomas, M. (1992). The industrial use of formal methods. *Microprecessors and Microsystems*, 17(1):31–36.

van der Aalst, W. and Odijk, M. (1995). Analysis of railway stations by means of interval timed colored Petri Nets. *Real-Time Systems*, 9(3):241–263.

Wang, J. and He, W. (1995). Formal Specification of Stability in Hybrid Control Systems. Research Report 56, UNU/IIST, P.O.Box 3058, Macau.

Wang, J., Yu, X., and Zhou, C. (1994). Hybrid Refinement. Research Report 20, UNU/IIST, P.O.Box 3058, Macau.

Widjaja, B. H., He, W., Chen, Z., and Zhou, C. (1994). A Cooperative Design for Hybrid Systems. Research Report 36, UNU/IIST, P.O.Box 3058, Macau. Published in: Logic and Software Engineering International Workshop in Honor of Chih-Sung Tang, pp 127-150, Edited by A. Pnueli and H. Lin World Scientific, 1996.

Wing, J. M. (1990). A Specifier's Introduction to Formal Methods. *IEEE Computer*, 23(9):8–24.

Wing, J. M., Woodcock, J., and Davies, J., editors (1999). *FM'99 — Formal Methods*, volume 1709 of *LNCS: Lecture notes in computer science*. Springer–Verlag. This is volume II of a two volume proceedings from the first World Congress on Formal Methods in the Development of Computing Systems. Organised jointly by FME (Formal Methods Europe) and ONERA (The French Government's Space Research Centre), Toulouse, France, Sept. 20–24, 1999.

Winter, K. (2002). Model checking railway interlocking systems. In *Proceedings of the twenty-fifth Austral–Asian Conference on Computer Science*, pages 303–310, Darlinghurst, Australia. Australian Computer Society, Inc.

Wong Wai (1991a). *A Formal Theory of Railway Track Networks in Higher-order Logic and its Applications in Interlocking Design*. PhD thesis, University of Warwick.

Wong Wai (1991b). A simple graph theory and its application in railway signaling. In M. Archer, J.J. Joyce, K.N. Levitt, and P.J. Windley, editors, *International Workshop on Higher Order Logic Theorem Proving and its Applications*, pages 395–410, Davis, California. IEEE Computer Society, ACM SIGDA, IEEE Computer Society Press.

Wood, W. G. (1990). Application of formal methods to system and software specification. *ACM Software Engineering Notes*, 15(4):144–146. Proceedings of the ACM Workshop on Formal Methods and Software Development (Napa, California, December 1989).

Woodcock, J., editor (1998). *FME Rail Workshop #2*, volume 2 of *FME Rail Seminars*, Parks Road, Oxford OX1 3QD, England. FME: Formal Methods Europe, Oxford Univ., Computing Lab. ESSI Project 26538. Workshop venue: Canary Wharf, London Docklands, England. Organised by Formal Systems Ltd., Oxford. Hosted by London Underground.

Woodcock, J. (2002). Semantics of Parallel Programming Languages in the Unifying Theory. In *Logic and Algebra for Engineering Software*. IOS Press.

Woodcock, J. and Hughes, A. (2002). Unifying Theories of Parallel Programming. Technical University of Munich. Notes for Marktoberdorf Summer School.

Woodcock, J. and McEwan, A. (2002). Verifying the Properties of a Railway Signalling Device. In *Proceedings of the International Conference on Integrated Design and Process Technology*, Passadena. IDPT Press. Winner of the Rudolph Christian Karl Diesel best paper award.

Woodcock, J. C. P. and Davies, J. (1996). *Using Z: Specification, Proof and Refinement*. Prentice Hall International Series in Computer Science.

Xia, Y. and George, C. W. (1999). An Operational Semantics for Timed RAISE. In Wing, J. M., Woodcock, J., and Davies, J., editors, *FM'99 — Formal Methods*, pages 1008–1027. FME, Springer–Verlag. Cf. (Wing et al., 1999).

Yu, H. and Zhou, C. (1994). A Duration Model for Railway Scheduling. Technical Report 24b, UNU/IIST, P.O.Box 3058, Macau.

Yu, X. (1994a). On Stability of Hybrid Systems. Technical Report 29, UNU/IIST, P.O.Box 3058, Macau.

Yu, X. (1994b). Stability of Railway Systems. Technical Report 28, UNU/IIST, P.O.Box 3058, Macau.

YuLin, D., Bjørner, D., and Prehn, S. (1994). Domain Analysis: A Case Study of Railway Station Management. Technical Report db/03/01, UNU/IIST, the UN University's International Institute for Software Technology, P.O.Box 3058, Macau; E-Mail: `library@iist.unu.edu`. Presented at KICS'94: The Kunming (Yunnan, PRC) Intl. CASE Symposium, Nov. 1994.

Zhou, C. (1993). Duration Calculi: An Overview. In *Proceedings of Formal Methods in Programming and Their Applications, D. Bjørner, M Broy, and I.V. Pottosin (Eds.)*, pages 256–266. LNCS 735, Springer-Verlag.

Zhou, C. and Hansen, M. R. (2003). *Duration Calculus: A formal approach to real–time systems*. Monographs in Theoretical Computer Science. Springer–Verlag.

Zhou, C., Hoare, C., and Ravn, A. (1991). A Calculus of Durations. *Information Processing Letters*, 40(5):269–276.

Zhou, C. and Li, X. (1994). A mean value calculus of durations. In Roscoe, A., editor, *A Classical Mind: Essays in Honour of C.A.R. Hoare*, pages 431–451. Prentice Hall International.

Zhou, C., Ravn, A., and Hansen, M. (1993). An extended duration calculus for hybrid systems. In Grossman, R., Nerode, A., Ravn, A., and Rischel, H., editors, *Hybrid Systems*, volume 736 of *Lecture Notes in Computer Science*, pages 36–59. Springer-Verlag.

Zhou, C., Wang, J., and Ravn, A. P. (1995). A Formal Description of Hybrid Systems. Research Report 57, UNU/IIST, P.O.Box 3058, Macau. Published in *Hybrid Systems III,* R. Alur, T. Henzinger, and E. Sontag (Editors), LNCS 1066, pp. 511-530, Springer-Verlag, 1996.

Zhou, C. and Yu, H. (1994). A duration Model for Railway scheduling. Technical Report 24b, UNU/IIST, P.O.Box 3058, Macau.

zu Hörste, M. M. (1999). Modelling and Simulation of Train Control Systems with Petri Nets. In *FME Rail Workshop #3*, volume # 3. FME: Formal Methods Europe, Fachhochschulstudiengang St. Pölten, Herzogenburgerstr. 68, A-3100 St. Pölten, Austria; Phone: +43 2742 313 228, Fax: +43 2742 313 229. Technische Universitt Braunschweig (D).

White Box Fuzzing (SAGE) Results. Since 2007: many new security bugs found. â€“ Apps: decoders, media players, document processors, â€¦ â€“ Bugs: Write A/Vs, Read A/Vs, CrashesÂ â€¢ Formal methods are surprisingly feasible for mainstream software development and give good return on investment. â€¢ At Amazon, formal methods are routinely applied to the design of complex real-world software, including public cloud services. Chris Newcombe, AWS. â€¢ Formal methods find bugs in system designs that cannot be found through aâ€œnTyLAo+thisetrhteemchonstiqvauleuawblee kthninogwthoaft I've learned in. my professional career. It has changed how I work, by giving me an immensely powerful tool to find subtle.