Software and Systems Modeling (2025) 24:1935–1954
https://doi.org/10.1007/s10270-025-01276-3

EXPERT VOICE

# Models for formal methods and tools: the case of railway systems

**M. H. ter Beek**[1]

Received: 16 May 2024 / Revised: 30 August 2024 / Accepted: 30 January 2025 / Published online: 28 February 2025


**Abstract**
Formal methods and tools are successfully applied to the development of safety-critical systems for decades now, in particular in the transport domain, without a single technique or tool emerging as the dominant solution for system design. Formal methods are highly recommended by the existing safety standards in the railway industry, but railway engineers typically lack the knowledge to transform their semi-formal models into a formal model, with a precise semantics, that can serve as input to formal methods tools. We share the results of performing empirical studies in the field, including usability analyses of formal methods tools involving railway practitioners. We discuss, in particular with respect to railway systems and their modelling, our experiences in applying formal methods and tools to a variety of case studies, for which we interacted with a number of companies from the railway domain. We report on lessons learned from these experiences and provide pointers to steer future research towards facilitating further synergies between researchers and developers of formal methods and tools on the one hand and practitioners from the railway industry on the other.

**Keywords** Formal methods · Models · Railway systems

## 1 Introduction

For decades, railways are controlled by real-time computer-based systems. To fulfil stringent safety requirements, railway control systems require extensive verification and validation, often relying on *formal methods*, as highly recommended by the CENELEC European standards [1, 2] for the development of the most critical software for use in the railway industry. This is demonstrated by hundreds of recent projects financed by the European Shift2Rail Joint Undertaking[1] and its successor Europe's Rail[2] and by related initiatives outside the EU, like the UK Rail Research and Innovation Network (UKRRIN)[3] and the Chinese State Key Laboratory of Rail Traffic Control and Safety[4].

We participated in the Shift2Rail projects ASTRail[5] and 4SECURail[6] and in the Italian regional projects TRACE-IT (TRAin Control Enhancement via Information Technology), SISTER (SIgnalling & Sensing TEchnologies in Railway application), STINGRAY (SmarT station INtelliGent RAilwaY) and SmaRIERS (Smart Railway Infrastructures: Efficiency, Reliability and Safety). Currently, we are involved in Spoke 4 on Rail Transportation of the National Centre for Sustainable Mobility (MOST), a NextGenerationEU project for more sustainable and digital mobility, in which we contribute to the development of moving block signalling systems, including satellite technologies for train positioning, and to the increase of rail transport digitalisation oriented towards predictive maintenance approaches based on big data analytics for safe and sustainable infrastructure maintenance, both with trackside and on-board systems. Through these projects, we interacted with a number of companies from the railway domain, which allowed us to apply more than three decades of experience with formal methods and tools accumulated in our research lab FMT (Formal Methods and Tools) to a variety of case studies from the railway domain.

The main safety-critical railway signalling systems can be classified in two large classes of applications: train

Communicated by G. Taentzer, A. Cicchetti, A. Pierantonio, and T. Kühne.

✉ M. H. ter Beek
maurice.terbeek@isti.cnr.it

[1] Formal Methods and Tools lab, CNR–ISTI, Via G. Moruzzi 1, Pisa 56124, Italy

[1] https://shift2rail.org/
[2] https://rail-research.europa.eu/
[3] https://www.ukrrin.org.uk/
[4] https://en.bjtu.edu.cn/research/institute/laboratory/16583.htm
[5] https://www.astrail.eu/
[6] https://www.4securail.eu/

movement and distancing control systems, including the Automatic Train Control (ATC), Automatic Train Operation (ATO), Automatic Train Protection (ATP) and Automatic Train Supervision (ATS) subsystems, and interlocking systems. All these subsystems respect international standards to ensure interoperability between the different subsystems described. These include ERTMS/ETCS (European Rail Traffic Management Systems/European Train Control System), its Chinese counterpart CTCS (Chinese Train Control System), both focusing on interoperability for passenger, high speed and freight lines, and CBTC (Communication-Based Train Control) systems, mainly aimed at the automatic operation of high capacity metro lines.

The above-mentioned classes of subsystems have been subject to formal specification and verification for several decades now, as witnessed by the many success stories of the application of the formal method B, which include the verification of the ATP system for the RER Line A of Paris [3], the Subway Speed Control System (SSCS) of the Calcutta subway [4], and Line 14 of the Paris Metro [5], as well as derivatives thereof, like line 1 or the NY Canarsie line [6], and the driverless Paris–Roissy Airport shuttle [7]. Moreover, B was also used for an industrial scale analysis of Alstom's U400 system [8], which is in operation in about 100 metro lines worldwide.

Further success stories of applying formal methods and tools to railway systems include the metro control system of Rio de Janeiro, with the support of Simulink/Stateflow [9], the ERTMS/ETCS standard with NuSMV [10] and Hybrid ERTMS/ETCS Level 3 with a variety of formal methods and tools as part of the ABZ 2018 case study [11–18]. Moreover, in [19], the system structure of a movement authority scenario of CTCS Level 3 was modelled in AADL [20] and Hybrid CSP [21], and verified with the Hybrid Hoare Logic Prover [22], an interactive theorem prover based on Isabelle/HOL [23]. Recently, in [24], the autonomous positioning system in development for the Florence tramways was verified with the UPPAAL model checking toolset[7].

In this paper, we first provide some background on formal methods and tools in Sect. 2, after which we share the results of performing empirical studies in the railway domain, including usability analyses of formal methods tools involving railway practitioners, in Sects. 3 and 4. We then present some applications of formal methods and tools to case studies from the railway domain in Sect. 5. Throughout the paper, we report lessons learned from these experiences and provide pointers to drive future research, which we discuss in more detail in Sect. 6, which is an original contribution, after which we summarise the paper in the concluding Sect. 7.

[7] https://uppaal.org/

## 2 Formal methods and tools

The development of industrial systems typically follows some standardised methodology that distinguishes different phases. In general, there is a requirements analysis phase, an architecture and design phase, a detailed design phase, an implementation phase, a testing, verification and validation phase and a maintenance phase. Each phase yields a set of artefacts (*models*). The methodology not only prescribes the type of models delivered at each phase, but also the verification and validation methods and activities needed to ensure the internal consistency of a model as well as the consistency of models across different phases. For instance, the models produced during the requirements phase need to be non-contradictory, while the models produced in the design phase need to be consistent with the models delivered in the requirements phase. *Formal methods*, and their implementation in *tools*, provide automated, repeatable, easily checkable evidence to support these needs.

Formal methods are rigorous mathematics-based techniques and tools for the specification (*modelling*) and manual or automated verification (*analysis*) of software or hardware systems (*designs*) [25]. They encompass a wide choice of techniques and tools, which are widely applied in industry [26], in development phases ranging from requirements elicitation and early design to deployment, configuration and runtime monitoring of actual systems. Formal methods allow us to precisely specify the requirements and properties that a system should satisfy, the environment in which the system operates, the code embedded in the final implementation and—most importantly—the models of the system used during the various design steps and proper conformance relations between these specifications. This requires formal models with a precise semantics, i.e. unlike UML models which are *semi-formal*.

More specifically, systems are interpreted as mathematically precise structures (e.g. state machines) describing system behaviour, specifications are given in a mathematically precise manner (e.g. logical properties) and also the notion of a system satisfying a specification is defined mathematically. Formally verifying a system then involves constructing a mathematical proof that the system satisfies the specification (e.g. absence of deadlocks). The key motivation for the use of formal methods is the strength of the correctness guarantees they provide: a proof conclusively demonstrates that the system under scrutiny, at the provided level of abstraction, is correct with respect to its specification. Formal methods may complement techniques like testing or simulation [27]. These popular non-exhaustive methods for verification reason over the *input space* of a system by producing a set of individual system executions. By aggregating a large set of such executions, these methods can provide

probabilistic answers to questions like: *how often/likely is it for some behaviour to occur?*

Since neither testing nor simulation explores all possible system behaviour (state space), paraphrasing Dijkstra [28], testing can provide a proof of the existence of erroneous behaviour, such as a system execution that exhibits a fault, but it can never show the absence of such behaviour. For that, we need formal methods to exhaustively reason over the entire *behaviour space* of a system. Formal methods yield Boolean (typically "yes"/"no", not probabilistic) answers to questions like: *is it possible for something to occur?* While testing involves executing the system many times to gather examples for verification, formal methods tools generally execute once but reason exhaustively over the complete set of system behaviour, covering all possible executions. This is also why there is such a variety of formal methods: different systems require different proof strategies to enable exhaustive reasoning. Moreover, while testing basically concerns supplying an input and checking the output, formal methods require some knowledge of the underlying system. In practice, formal methods can prove both the presence and the absence of given behaviour. Finally, by nature, formal methods can also check systems being used in ways they were not intended, whereas traditional testing easily ignores unanticipated inputs.

There are two core formal methods: *model checking* [29, 30] and *theorem proving* [31, 32]. Theorem provers like Coq [33], Isabelle [34], KeYmaera X [35, 36] or Z3 [37] allow users to mechanically prove generic statements about system models formalised as mathematical theories. Theorem provers are quite powerful tools that can reason about very large, or even infinite-state systems, but they require substantial user knowledge. Theorem proving are at the basis of tool sets for state-based refinement like Atelier-B[8] and Rodin[9].

Model checkers like SPIN [38], NuSMV [39]/nuXmv [40], ProB [41], UPPAAL [42–46], FDR [47], CADP [48] and mCRL2 [49] provide a convenient "push-button" technology: "the use of model checking requires neither a high degree of user interaction nor a high degree of expertise" [29, Section 1.2.2: Strengths and Weaknesses]. Model checking addresses the verification problem: given a formal definition of what a system does (a *model*) and what it should do (a *property*), model checking shows that the model *satisfies* the property, i.e. that the system does what you think it should do and nothing else [30].

In this way, both theorem provers and model checkers effectively prove both the presence and the absence of bugs. However, both verify a model of the system rather than the system itself and verify only stated requirements, meaning that "any obtained result is thus as good as the system model" and "there is no guarantee of completeness" [29, Section 1.2.2: Strengths and Weaknesses]. Moreover, theorem provers require guidance from a deeply-knowledgeable user to structure and organise the specification and its proof, while model checkers are sensitive to the shape and size of the state space of the system, due to their exhaustive exploration of the state space according to which the number of states needed to model the system accurately may exceed the amount of available computer memory.

Compared to model checking, which focuses on absolute guarantees of correctness, probabilistic model checking focuses on modelling and analysing systems exhibiting probabilistic behaviour [29, 50], which is essential in cases of unreliable or unpredictable system behaviour and performance evaluation. Rather than providing a Boolean answer to the question as to whether a system model satisfies a logic property (typically expressed in a probabilistic temporal logic), the answers are of the form: *with a likelihood of 99%, the model will satisfy the property*.

Statistical model checking (SMC) [51, 52] uses a simulation- and sample-based approach to reason about precise properties defined in a probabilistic temporal logic, offering scalability advantages over exhaustive (or probabilistic) model checking as there is no need to analyse full state spaces. Moreover, while the output of sample-based methods is not always correct, statistical inference allows to quantify the confidence in the result, thus compensating for the lack of exact results (100% confidence). However, SMC is known to have difficulties in dealing with rare events. Arguably the best known tool is UPPAAL SMC [46].

Static analysis involves deriving properties of interest from source code (or an intermediate representation) without actually executing it [53], meaning precision is the price to pay. Typically, one has to decide between under- and overapproximations (e.g. abstract interpretation [54]) of all possible behaviour, with the possibility of both false positives and false negatives. The choice depends on the specific goals of the analysis and the trade-off between precision and completeness, i.e. finding a suitable abstraction that is both computationally feasible yet still provides meaningful insights into the behaviour.

Finally, model-based testing is a formal methods approach to testing that complements formal verification and model checking and increases the efficiency and effectiveness of software testing [55]. It uses a formal or semi-formal model to represent the desired behaviour of a system under test, which serves as the basis for generating test cases and executing tests. It is typically more efficient than traditional testing, since it automates the test case generation process. Moreover, by systematically deriving test cases from the model, often a better coverage of system behaviour is achieved. However, the models need to be kept up-to-date as the system evolves,

[8] https://www.atelierb.eu/

[9] https://www.event-b.org/

which may require additional effort. Model-based testing is part of the broader landscape of Model-Driven Engineering (MDE) [56].

Concrete symbolic (concolic) testing is a hybrid verification technique that performs symbolic execution along concrete execution paths to systematically explore the execution of a program or software system, focusing on both specific input values (as for traditional testing) and symbolic representations of alternative paths, to achieve improved path coverage compared to traditional testing. Scalability is a challenge due to the path explosion problem, i.e. the number of possible paths grows significantly as the program's complexity increases [57]. Fuzzing is a more light-weight testing technique focused on quickly exploring a large input space by providing (semi-)random inputs, typically generated by mutation-based or generation-based fuzzing, to discover vulnerabilities or unexpected behaviour [58].

## 3 Systematic mapping study

We conducted the first systematic mapping study[10] on formal methods in the railway domain [59], focusing on railway signalling systems, with the goal to "identify, understand and characterise studies on the application of formal methods to the development of railway systems, identifying recent trends and considering industrial applications, for the purpose of supporting formal methods practice and research."This complements other empirical studies that we conducted, considering the perspective of stakeholders [62, 63] and surveying tools for railway system design [64–66] (cf. Section 4). To address the above goal, we posed ourselves several research questions (RQs), among which the following[11]:

RQ1: *How is research demographically and empirically characterised in the field of applications of formal methods in the railway domain?*
RQ2: *What formal methods are used in the railway domain?*

RQ2.1: *What is the degree of* ***formality****?*
RQ2.2: *What* ***formal techniques*** *are used?*
RQ2.3: *Which specification* ***languages****?*
RQ2.4: *Which* ***tools****?*

RQ3: *In which way are formal methods applied to railway system development?*

RQ-I: *What are the characteristics of the studies reporting* ***industrial*** *applications?*
RQ-T: *What are the emerging* ***trends*** *of the last 5 years?*

Starting from the base terms "formal methods" (representing the object of research) and "railways" (representing the domain of application), we used alternative keywords and wildcards to elaborate the following search string:

"formal" OR "model check*" OR "model based"
OR "model driven" OR "theorem prov*"
OR "static analysis"
**AND**
"railway*" OR "CBTC" OR "ERTMS"
OR "ETCS" OR "interlocking"
OR "automatic train" OR "train control"
OR "metro" OR "CENELEC"

As shown in Figure 1, after originally retrieving 4346 studies from four main scientific databases, eventually 328 high-quality studies were selected from the literature for the period 1989–2020. We analysed the 328 papers in detail.

Figure 2 answers RQ1 and contributes to answer RQ-I by showing that formal methods for railways is a hot topic with a strong industrial focus, given that 143 papers were published solely during the last five years (44% of the total), while no less than 79 papers (24%) involve industry (papers with at least one industrial co-author).

Concerning the degree of formality of the methods used in these papers, Fig. 2 answers RQ2.1 by showing that most are strictly formal (212, 65%) and only a minimal part is purely semi-formal (31, 9%). Interestingly, as a contribution to answer RQ-I, semi-formal methods are more prominent in industrial papers, of which only half use exclusively formal methods (40, 51%), while the other half uses at least some semi-formal method.

Figure 4 answers RQ2.2 and contributes to answer RQ-I by showing that models are at the basis of practically all papers and that formal verification and model checking are the main analysis techniques [29, 30], but also simulation [27], theorem proving [31, 32] and refinement [67–69] are prominently applied techniques.

Figure 5 provides a further demonstration of the fact that models are at the basis, given that the development phases considered in the vast majority of the papers concern Architecture (218, 66%) and Detailed Design (149, 45%), thus providing a partial answer to RQ3. To a certain extent, this trend is followed by the industrial papers, even though these focus more on later phases of system development and on lower-level system representations. In particular, a relevant number of industrial papers considers Validation (20 out of 30, 67%) and Implementation (11 out of 14, 79%), while a lower percentage of industrial papers considers Architecture (47, i.e. 59% vs. 66%) and Detailed Design (36, 46% vs. 45%), thus contributing to answer RQ-I.

[10] A variant of systematic literature reviews aiming at classifying the literature [60, 61].

[11] RQ1–RQ3 are independent questions, while RQ-I and RQ-T, instead, are cross-cutting questions that aim to address *industrial applications* and *recent trends* (e.g. certain formal methods identified in RQ2 may be more trendy or industrially validated). As such, RQ-I and RQ-T are answered in relation to RQ1–RQ3 to facilitate the interpretation of the data and have a more concise visualisation of the statistics.

Search String

Inclusion Criteria (*Abstract/Title*)

| | ACM Digital Library | | | IEEE Xplore | | | Science Direct | | | SpringerLink - CS | | | SpringerLink - ENG | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Primary Search** | I | II | III | I | II | III | I | II | III | I | II | III | I | II | III |
| **Retrieved** | 72 | 9 | 321 | 631 | 88 | 205 | 321 | 86 | 141 | 1780 | 243 | 318 | 124 | 4 | 3 |
| **Included** | 34 | 0 | 6 | 163 | 15 | 28 | 74 | 12 | 8 | 298 | 22 | 55 | 8 | 1 | 0 |

Exclusion Criteria (*Full Text*)
Quality Criteria

| **Primary Search** | I (X - 2017) | II (2017-2018) | III (2018 - 2020) | **Secondary Search** |
|---|---|---|---|---|
| **Selected** | 198 | 27 | 67 | 36 |

| | **TOTAL** |
|---|---|
| **Retrieved** | 4346 |
| **Included** | 724 |
| **Selected** | **328** |

**Fig. 1** Process of study selection and numerical results. The value of TOTAL in the bottom-right table is obtained by summing-up the cells in retrieved, included and selected from the other tables. ©2022 ACM. Reprinted (and annotated) with permission from ACM Computing Surveys [59]

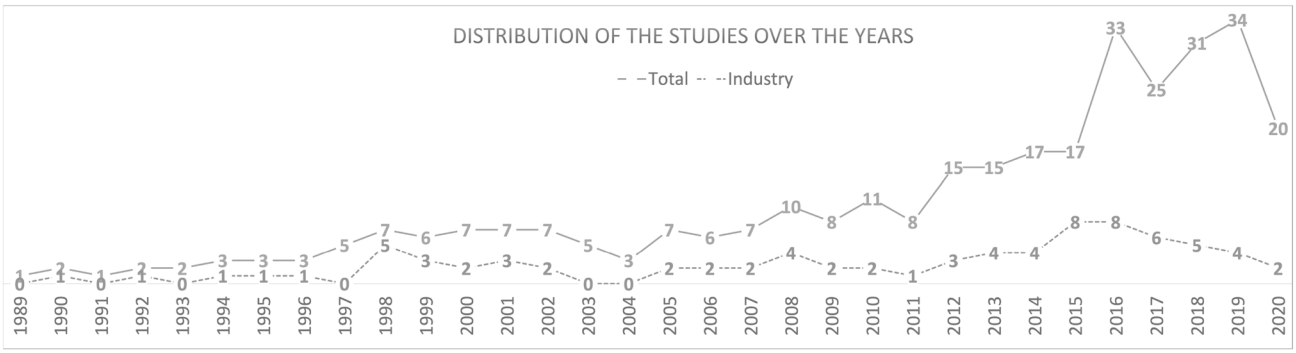


**Fig. 2** Studies by year. ©2022 ACM. Reprinted with permission from ACM Computing Surveys [59]

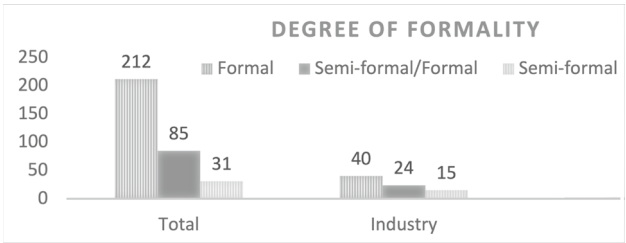


**Fig. 3** Degree of formality. ©2022 ACM. Reprinted (and adapted) with permission from ACM Computing Surveys [59]

In terms of concrete languages and tools, Figs. 6 and 7 answer RQ2.3 and RQ2.4, respectively, and contribute to answer RQ-I by illustrating that the landscape is highly diversified, but the dominant modelling languages are the semi-formal method UML (18% in total and 18% of the industrial papers) [70–76] and the formal method B (15% in total and 22% of the industrial papers) [7, 8, 16, 76–81], while tools of the B family (i.e. ProB, Atelier-B and Rodin) clearly dominate (18% overall and 20% of the industrial papers) [8, 12, 16, 18, 76, 78–80, 82–84], whereas the IBM Rational family for UML and SysML models is hardly used. Interestingly, UML is used in combination with all main tools even though, with the exception of IBM Rational, none of the tools is specifically oriented to support UML. Typically, UML is used to model the system, but then the model is translated into the input language of specific formal methods tools. This is in line with the fact that UML is the de facto industrial standard for documentation and communication among stakeholders.

Our systematic mapping study shows that the empirical maturity of the field is still limited, as many of the selected papers present only examples or experience reports. We call for more empirical rigour in the field, with case studies, which can leverage the strong link with industries, and for controlled experiments, which can address issues related to the learnability of formal methods and aspects related to human factors. In fact, many papers do not focus on a particular railway system or standard, but we signal an improvement with high-quality papers on the ERTMS/ETCS [16, 85], CBTC [8] and CTCS [86] standards. Interlocking is still the most popular subject of study, but other railway subsystems (e.g. ATC, ATO, ATP and ATS) are being considered more frequently in recent years [74, 75, 87].

For an elaborate answer to RQ-T, we refer to [59], which includes versions of Figs. 3–7 restricted to papers from the last 5 years. Figure 2 does contribute to answer RQ-T by showing an increase of papers during the last 5 years, totalling almost half of the total number of papers (143, 44%), but a decrease of industrial papers.

In recent years [59], the B language has surpassed UML as the most common modelling language (20%) [8, 16, 76, 81], while a quantitative analysis tool like UPPAAL (20, 14%) [86, 87] has emerged over the last 5 years, closely followed by the B model checker ProB (13%) [8, 16, 76].

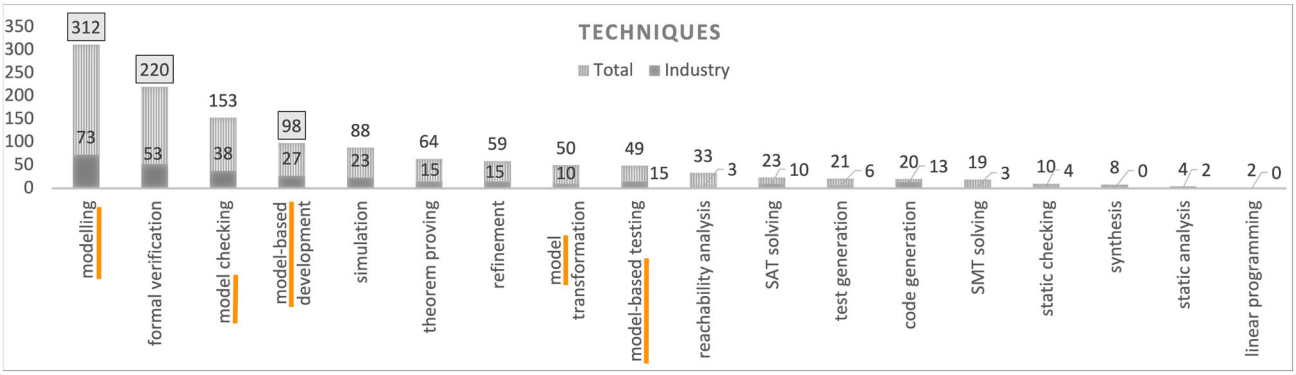


**Fig. 4** Techniques. ©2022 ACM. Reprinted (and annotated) with permission from ACM Computing Surveys [59]

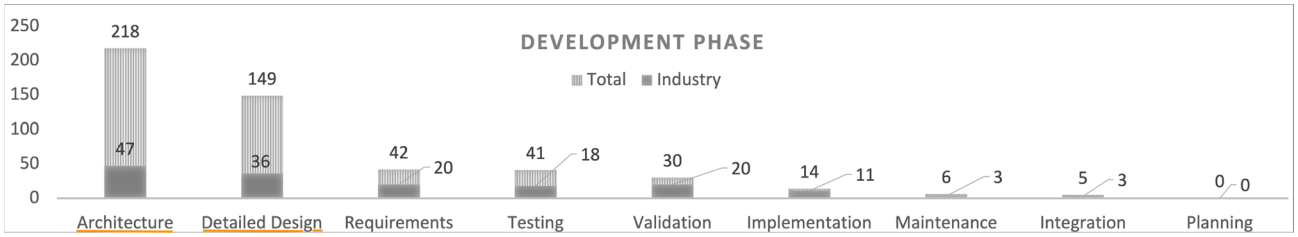


**Fig. 5** Railway development phase. ©2022 ACM. Reprinted (and annotated) with permission from ACM Computing Surveys [59]

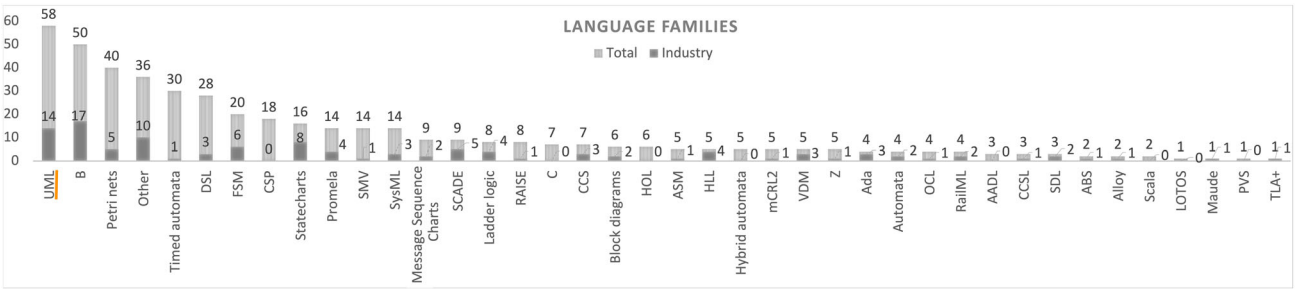


**Fig. 6** Modelling language families considered in the studies. ©2022 ACM. Reprinted (and annotated) with permission from ACM Computing Surveys [59]

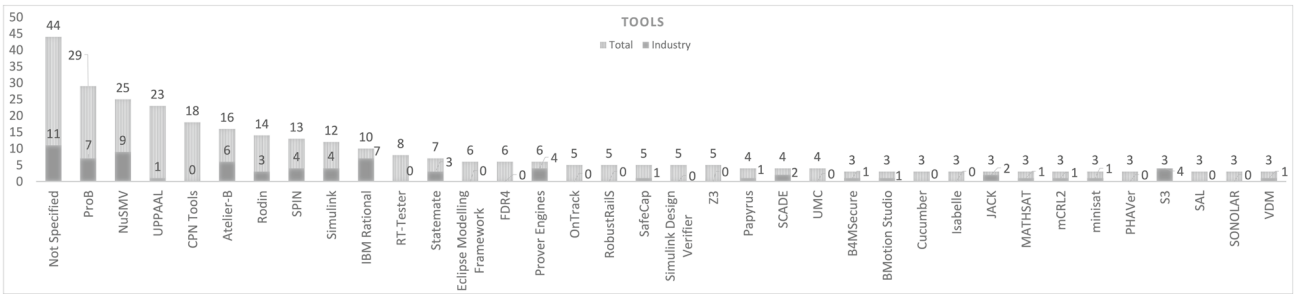


**Fig. 7** Tools. ©2022 ACM. Reprinted with permission from ACM Computing Surveys [59]

Take-away messages [59]:
- − The later phases of the development process, in particular testing, implementation and validation, are currently not sufficiently addressed.
- − Formal methods are typically applied on abstract, high-level models, and source code is only marginally considered.
- + UML is typically used for high-level representation, after which models are translated into the input language of some specific formal methods tool.
- + Almost all core railway development phases can be addressed with the support of formal methods, which is in line with the recommendations of the norms [1, 2].

## 4 Systematic evaluation and usability analysis

We conducted the first systematic tool evaluation[12] and usability analysis of formal methods tools for railway signalling system design [66]. This complements other empirical studies that we performed, which surveyed various tools for railway system design [64, 65] selected based on surveys with stakeholders [63, 89] and a literature review [59] (cf. Section 3). The 13 selected tools are: SPIN, Simulink [90], nuXmv, ProB, Atelier-B, UPPAAL[13], FDR4, CPN Tools [91], CADP, mCLR2, SAL [92], TLA+ [93] and UMC [94].

The first seven tools plus CADP and UMC were reported among the most mature formal tools for railways [65], based on the fact that the first six tools and UMC were top ranked in surveys with stakeholders [63, 89], while FDR4 and CADP are representative tools for the analysis of process algebras that are explicitly mentioned in the railway norms [2]. The remaining four tools were added in [66] to obtain a more representative set of the different flavours of formal system modelling and verification. In particular, CPN Tools was included as a representative for the analysis of Petri Nets, which are widely used also for railway modelling [63], mCLR2 as an open-source representative for the analysis of process algebras, and SAL and TLA+ as representatives of tools that integrate theorem proving and model checking. The licensing costs did not allow the inclusion of relevant commercial and industry-relevant tools like SCADE, Systerel products, or Prover in [66].

[12] Based on the DESMET methodology for evaluating software engineering tools [88].

[13] In [66], UPPAAL denoted all variants now integrated in UPPAAL 5: https://uppaal.org/ (i.e. UPPAAL 4.0 [43], UPPAAL SMC, UPPAAL Stratego [45] and UPPAAL Tiga [44]).

We posed ourselves the following RQs [66]:

RQ1: *Which are the features to consider for evaluating a formal methods tool?*
RQ2: *How do different tools compare with respect to these features?*
RQ3: *How do different tools compare with respect to their usability?*

Table 1 summarises the characteristics and expertise of the participants who evaluated the tools based on hands-on experience. Next to the assessors and academic experts (including the authors of [66]), we involved nine railway practitioners, who had no prior experience with applying formal methods tools yet over 10 years of experience in the railway domain, as industry experts.

The systematic feature evaluation was performed by the assessors, after elicitating the features with a collaborative approach inspired by the KJ method[14] [96]. During a 3-hour workshop, the assessors, the academic experts, and two industry experts came up with 33 features that should be considered when evaluating a formal methods tool for railway system design, hierarchically categorised into functional, expressiveness, and quality features, comprising 8 subcategories. The assessors produced an evaluation sheet for each tool based on the following use:

1. Install and run the tool;
2. Consult the tool's website for official documentation;
3. Search for additional documentation useful to fill the evaluation sheet;
4. Consult the 114 papers on formal methods and railways from the literature review [89] to check for the tool's application in railways;
5. Perform some trials with the tool to confirm the claims reported in the documentation, and assign the value to those features that required hands-on activity to be evaluated;
6. Report the evaluation on the sheet, together with links to the consulted documents and papers, and appropriate notes when the motivation of an assignment needed clarification.

The evaluation sheets were revised after face-to-face meetings to align visions and balance judgements and reviewed externally as part of a deliverable of the ASTRail project; they are publicly available [64]. Figure 8 reports the table resulting from the feature evaluation activity.

[14] The KJ method is an exploratory and creative requirements elicitation technique, according to which requirements are first individually written on cards and then collectively discussed and grouped [95, Chapter 23.4: KJ Method].

**Table 1** Characteristics and expertise of the study participants

| ID | Role in Study | Milieu | Main function | Age | Sex | Years of experience in | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | Formal methods (FM) | Railway industry | FM in railways |
| 1 | Assessor | Academic | Workpackage leader | 39 | M | > 13 | 3 | 13 |
| 2 | Assessor | Academic | Tool developer | 62 | M | > 20 | 0 | 9 |
| 3 | Assessor | Academic | Researcher | 36 | M | > 6 | 0 | 4 |
| 4 | Expert | Academic | Group leader | 48 | M | > 15 | 0 | 9 |
| 5 | Expert | Academic | Project leader | 66 | F | > 30 | 0 | > 25 |
| 6 | Expert | Academic | Professor | 65 | M | > 30 | 0 | > 25 |
| 7 | Expert | Industry | System engineer | NA | M | 0 | > 10 | 0 |
| 8 | Expert | Industry | System engineer | 52 | M | 0 | > 10 | 0 |
| 9 | Expert | Industry | System engineer | 48 | M | 0 | > 10 | 0 |
| 10 | Expert | Industry | Software developer | 43 | M | 0 | > 10 | 0 |
| 11 | Expert | Industry | Product manager | NA | M | 0 | > 10 | 0 |
| 12 | Expert | Industry | System engineer | 48 | M | 0 | > 10 | 0 |
| 13 | Expert | Industry | Innovation engineer | NA | M | 0 | > 10 | 0 |
| 14 | Expert | Industry | Software developer | 45 | M | 0 | > 10 | 0 |
| 15 | Expert | Industry | Innovation engineer | NA | F | 0 | 3–10 | 0 |



| Category | Name | SPIN | Simulink | nuXmv | ProB | Atelier B | UPPAAL | FDR4 | CPN Tools | CADP | mCRL2 | SAL | TLA+ | UMC |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Development Functionalities | Specification / Modeling | TEXT | GRAPH | TEXTIM | TEXT | TEXT | GRAPH | TEXTIM | GRAPH | TEXTIM | TEXT | TEXTIM | TEXT | TEXT |
| | Code Generation | NO | YES | NO | NO | YES | NO | NO | NO | YES | NO | NO | NO | NO |
| | Documentation / Report Generation | PARTIAL | YES | NO | PARTIAL | PARTIAL | PARTIAL | PARTIAL | NO | PARTIAL | PARTIAL | NO | NO | PARTIAL |
| | Requirements Traceability | NO | YES | NO | NO | NO | NO | NO | NO | NO | NO | NO | NO | NO |
| | Project Management | NO | YES | NO | YES | YES | NO | NO | NO | NO | NO | NO | NO | NO |
| Verification Functionalities | Simulation | TEXT | GRAPH | TEXT | MIX | NO | GRAPH | TEXT | GRAPH | TEXT | TEXT | TEXT | NO | TEXT |
| | Formal Verification | MC-L | MC-O | MC-L,MC-B | MC-L,MC-B,RF | TP | MC-L,RF | RF | MC-B | MC-B,RF | MC-B,RF | MC-L,TP | MC-L,TP | MC-B |
| | Large-scale Verification Technique | FLY,POR,PAR | BMC | BMC,SYM | SCT | SCT | SMC,SYM | COM,POR | BMC | COM,PAR | COM | PAR,SCT | SYM,SCT | FLY |
| | Model-based Testing | NO | YES | NO | YES | NO | YES | NO | NO | YES | NO | YES | NO | NO |
| Language Expressiveness | Non-determinism | INT | EXT | INT,EXT | INT,EXT | INT,EXT | INT,EXT | INT,EXT | INT | INT,EXT | INT,EXT | INT,EXT | INT | INT |
| | Concurrency | ASYNCH | NO | SYNCH | NO | NO | SYNCH | ASYNCH | ASYNCH | ASYNCH | ASYNCH | A/SYNCH | ASYNCH | A/SYNCH |
| | Timing Aspects | NO | YES | YES | NO | NO | YES | YES | YES | NO | YES | YES | NO | NO |
| | Stochastic or Probabilistic Aspects | NO | NO | NO | NO | NO | YES | NO | NO | NO | YES | NO | NO | NO |
| | Modularity of the Language | HIGH | HIGH | MEDIUM | LOW | LOW | MEDIUM | HIGH | HIGH | HIGH | HIGH | MEDIUM | MEDIUM | HIGH |
| | Supported Data Structures | BASIC | COMPLEX | COMPLEX | COMPLEX | COMPLEX | COMPLEX | COMPLEX | COMPLEX | COMPLEX | COMPLEX | COMPLEX | COMPLEX | COMPLEX |
| | Float Support | NO | YES | YES | NO | NO | YES | NO | NO | NO | NO | NO | NO | NO |
| Tool Flexibility | Backward Compatibility | LIKELY | LIKELY | LIKELY | LIKELY | MODERATE | LIKELY | MODERATE | LIKELY | LIKELY | LIKELY | MODERATE | MODERATE | MODERATE |
| | Standard Input Format | OPEN | PARTIAL | OPEN | OPEN | OPEN | PARTIAL | OPEN | PARTIAL | STANDARD | OPEN | OPEN | OPEN | STANDARD |
| | Import / Export vs. Other Tools | MEDIUM | LOW | MEDIUM | HIGH | MEDIUM | LOW | MEDIUM | MEDIUM | HIGH | HIGH | MEDIUM | LOW | MEDIUM |
| | Modularity of the Tool | LOW | HIGH | LOW | HIGH | MEDIUM | HIGH | LOW | LOW | HIGH | MEDIUM | LOW | LOW | MEDIUM |
| | Team Support | NO | NO | NO | NO | YES | NO | NO | NO | NO | NO | NO | NO | NO |
| Maturity | Industrial Diffusion | HIGH | HIGH | HIGH | HIGH | HIGH | HIGH | MEDIUM | MEDIUM | MEDIUM | MEDIUM | LOW | MEDIUM | LOW |
| | Stage of Development | MATURE | MATURE | MATURE | MATURE | MATURE | MATURE | MATURE | MATURE | MATURE | MATURE | MATURE | MATURE | PROTOTYPE |
| Usability | Availability of Customer Support | PARTIAL | YES | PARTIAL | YES | YES | YES | PARTIAL | PARTIAL | PARTIAL | PARTIAL | PARTIAL | PARTIAL | PARTIAL |
| | Graphical User Interface | LIMITED | YES | NO | PARTIAL | PARTIAL | YES | LIMITED | PARTIAL | LIMITED | PARTIAL | NO | LIMITED | PARTIAL |
| | Mathematical Background | MEDIUM | BASIC | MEDIUM | MEDIUM | ADVANCED | MEDIUM | ADVANCED | MEDIUM | ADVANCED | ADVANCED | ADVANCED | ADVANCED | MEDIUM |
| | Quality of Documentation | GOOD | EXCELLENT | GOOD | GOOD | EXCELLENT | GOOD | EXCELLENT | GOOD | GOOD | GOOD | GOOD | GOOD | LIMITED |
| Company Constraints | Cost | FREE | PAY | MIX | FREE | FREE | MIX | MIX | FREE | MIX | FREE | FREE | FREE | FREE |
| | Supported Platforms | ALL | ALL | ALL | ALL | ALL | ALL | ALL | Windows | ALL | ALL | ALL | ALL | ALL |
| | Complexity of License Management | EASY | ADEQUATE | EASY | EASY | EASY | MODERATE | MODERATE | EASY | MODERATE | EASY | EASY | EASY | EASY |
| | Easy to Install | YES | YES | YES | YES | YES | YES | YES | YES | PARTIAL | YES | YES | YES | YES |
| Railway-specific Criteria | CENELEC Certification | NO | PARTIAL | NO | NO | NO | NO | NO | NO | NO | NO | NO | NO | NO |
| | Integration in the CENELEC Process | MEDIUM | YES | MEDIUM | YES | YES | MEDIUM | MEDIUM | MEDIUM | MEDIUM | LOW | LOW | LOW | MEDIUM |
| | | SPIN | Simulink | nuXmv | ProB | Atelier B | UPPAAL | FDR4 | CPN Tools | CADP | mCRL2 | SAL | TLA+ | UMC |

**Fig. 8** Evaluation table. ©2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. Reprinted with permission from IEEE Transactions on Software Engineering [66]

We selected a subset of seven tools for the usability evaluation, excluding those requiring advanced mathematical background (except for Atelier-B, as it is one of the few tools used in the development of real-world railway products, cf. "Integration in the CENELEC process", Figure 8) and those that are known from the literature to be inadequate for

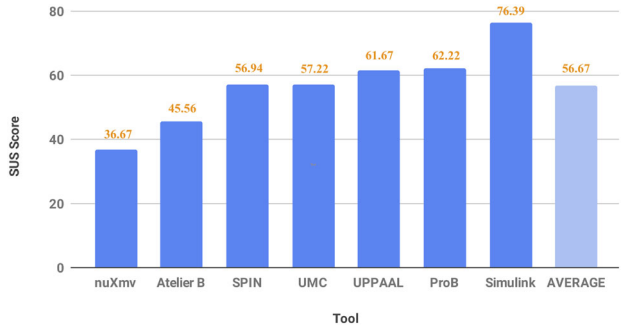


**Fig. 9** SUS scores for the different tools. ©2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. Reprinted (and annotated) with permission from IEEE Transactions on Software Engineering [66]

handling industry-size problems. We performed the usability evaluation of the resulting seven tools (SPIN, Simulink, nuXmv, ProB, Atelier-B, UPPAAL and UMC) with railway experts adopting the following methodology.

First, the assessors developed a model of the moving block system for each of the tools. The experts were already familiar with the sample system, which has moreover been used as a reference by other papers in the literature [11, 65].

Next, using the predeveloped models as reference, the different characteristics of the tools were illustrated during a 3-hour meeting with the experts. They were asked to evaluate the usability of each tool based on their first impression. After an introduction, each tool was presented in a 15-minutes demo, covering its general structure, then opening, navigating and describing the model, followed by a guided simulation and a description and presentation of a formal verification session. After each tool's presentation, the experts filled a usability questionnaire for the tool.

We used the widely adopted System Usability Scale (SUS) questionnaire of Brooke [97, 98] and his guidelines to calculate the SUS scores and Bangor et al. [99] for the interpretations: 100 = Best Imaginable; 85 = Excellent; 73 = Good; 52 = OK; 39 = Poor; 25 = Worst Imaginable.

Figure 9 presents the results of the SUS questionnaire. The tool that clearly stands out as being considered the most usable is Simulink (SUS Score = 76.39), a formal model-based development tool, with an appealing GUI and powerful simulation capabilities, followed by two other tools that score above 60. ProB (62.22) requires input models in textual form but presents simulation results also in graphical form, whereas UPPAAL (61.67) is entirely graphical and presents a graphical simulation style reminiscent of message sequence charts, which are well known by railway practitioners. Overall, tool usability is acceptable, given the average SUS Score (56.67) between OK and Good.

Take-away messages [66]:

- − Many of the formal methods tools lack support for development functionalities and process-integration aspects.
- − Most of the formal methods tools are independent ecosystems, with unique, non-standard languages and specialised verification capabilities.
- \+ Formal methods tools are mature, as highly desired by railway industry [62, 63].
- \+ Most usability aspects appear to be low in principle, but, when the formal methods tools are assessed by railway practitioners, usability is considered acceptable.

## 5 Success stories

In this section, we briefly discuss some of our experiences in applying formal methods and tools to case studies that resulted from our participation in European Shift2Rail projects like ASTRail and 4SECURail and Italian regional projects like TRACE-IT, SISTER, STINGRAY and SmaRIERS with industrial partners from the railway domain. We report on lessons learned from these experiences and provide pointers to steer future research.

### 5.1 Next-generation railway signalling systems

The railway domain is known to be cautious concerning the adoption of technological innovations, in particular when compared with other transport domains. Hence, while satellite-based positioning systems are in use for quite some time now in the avionics and automotive domains, current railway signalling systems still typically use traditional ground-based train detection systems with fixed block distancing. So-called ERTMS/ETCS Level 2 signalling systems make use of trackside equipment like track circuits or axle counters for exact train position detection and train integrity supervision and of fixed blocks starting and ending at signals, with the block sizes being determined by parameters like the speed limit, the train's speed and braking characteristics and the drivers' sighting and reaction times. Yet, the faster trains are allowed to run, the longer their worst-case braking distance, resulting in an increased safety distance and a decreased line capacity. Therefore, to increase the competitiveness, robustness and attractiveness of the railway domain, a recognised challenge consists of transitioning to the next generation of railway signalling system based on effective, precise moving block signalling systems by GNSS-based satellite positioning [100, 101].

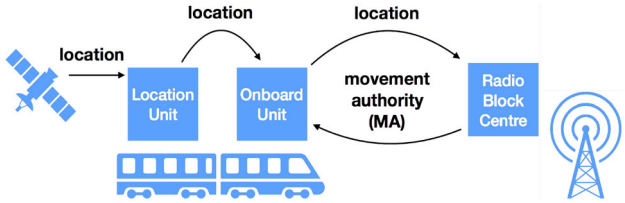


**Fig. 10** ERTMS/ETCS Level 3 moving block railway signalling (reprinted from [102])

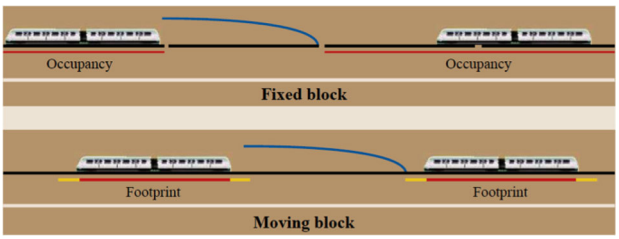


**Fig. 11** Safety train separation with fixed block and moving block signalling systems (Image by Israel.abad/Wikimedia Commons distributed under CC BY-SA 3.0 license)

Such next-generation ERTMS/ETCS Level 3 signalling systems no longer rely on trackside equipment for train position detection and train integrity supervision, but an on-board odometry system monitors the train's position, which it receives from a satellite, and autonomously computes its current speed. This on-board unit frequently sends the train's position to a radio block centre which, in turn, sends each train a movement authority (cf. Fig. 10), computed by exploiting its knowledge of the position of the rear end of the train ahead. The resulting moving block signalling systems allow trains in succession to close up, since a safe zone around the moving trains can be computed, thus considerably reducing headways between trains, in principle to the braking distance (cf. Fig. 11). This allows for more trains to run on existing railway tracks and the removal of trackside equipment moreover results in lower capital and maintenance costs [100].

Since our involvement in ASTRail, we have contributed to the many experiments and case studies that are being conducted and validated before actually starting to adopt ERTMS/ETCS Level 3 signalling systems [11–18, 85, 103–106]. We developed UPPAAL models by means of model transformations [105, 106], starting from (i) a Real-Time UML (RT UML) [107, 108] model provided by the project's industrial partners and (ii) a Simulink model obtained from a requirements elicitation and refinement activity performed with the project's industrial partners, which was carried out to consolidate an initial set of requirements for the moving block signalling system into an executable specification. Our task thus consisted of translating the semi-formal UML and Simulink models into a formal model of (stochastic) timed automata[15], amenable to formal verification by UPPAAL. We chose UPPAAL SMC since it allows for real-time as well as probabilistic aspects, both of which occur in RT UML models. The model transformation from RT UML to UPPAAL was rather straightforward, except for a few issues concerning the precise meaning of (time-related) modelling choices which had to be cleared with our industrial partners. The fact that UPPAAL's automata models are similar to UML state machine diagrams eased understanding by our partners; UPPAAL's feature to visualise them as message sequence charts also helped.

Simulink includes Stateflow, a graphical language inspired by Harel's hierarchical statecharts [109]. The fact that both formalisms use state machines simplified the model transformation from Simulink/Stateflow to UPPAAL (cf, e.g. [110]). We enriched the UPPAAL models with probabilities and stochastic events, taken from additional specifications of the moving block system by our industrial project partners. The UPPAAL models in [105, 106] concern a simplified moving block specification with only one train, and we used UPPAAL SMC for formal verification and sensitivity analysis based on SMC.

We presented an extension and refinement of the aforementioned UPPAAL models, considering more trains which concurrently communicate with the same (trackside) radio block centre [102]. The movement authority that was previously considered constant was now computed dynamically according to the traffic on the railway line, and in particular the tail of the train ahead. The physical behaviour of the trains was tuned and validated according to parameters about high-speed trains from the literature [111]. We formally verified the correctness of the function that computes the movement authority, formalised in first-order logic. The concurrent nature of the radio block centre led to the detection and mitigation of corner cases. We also carried out experiments to validate candidate parameter set-ups to reduce the risk of trains exceeding their movement authority.

### 5.2 Synthesis of autonomous driving strategies

We performed the first application of synthesis techniques to autonomous driving for next-generation railway signalling systems [112]. As described in Section 5.1, the RT UML, Simulink and UPPAAL SMC models from [105, 106] offered the possibility to fine-tune communication parameters that are fundamental for the reliability of their operational behaviour; however, they did not account for the synthesis of autonomous driving strategies. We presented a UPPAAL Stratego model (i.e. a stochastic priced timed game) of a satellite-based moving block railway signalling system that does account for autonomous driving. The autonomous

[15] Finite-state machines extended with clock variables [42].

driving module was not modelled manually, but it was synthesised automatically as a strategy based on a safety requirement that the model must respect, after which both standard model checking and SMC were applied under the resulting (safe) strategy. We moreover considered reliability aspects, and the autonomous driving strategy also provided guarantees for the minimal expected arrival time. It must be noted that the original model had to be simplified considerably to undergo strategy synthesis and verification, since UPPAAL SMC scales well to large systems by applying simulations rather than full state-space explorations, whereas UPPAAL Stratego requires full state-space exploration of the timed game for strategy synthesis.

This was our first experience with strategy synthesis and optimisation of a case study from the railway domain and also with UPPAAL Stratego. This is a rather recent formal methods tool which has not been much experimented with. In fact, while developing the model we ran into corner cases that needed interactions with the developers, which led to the release of new versions, with patches fixing the issues discovered through our modelling efforts.

In an attempt to be able to model more complex case studies from the railway domain, we mentioned that a promising line of future research would be to adapt the statistical synthesis techniques [113] to learn safety objectives, thus avoiding the full state-space exploration (as performed in UPPAAL Stratego at that time) while guaranteeing the scalability of SMC. In the work of Gu et al. [114–116], this idea has been successfully worked out in detail and integrated in UPPAAL Stratego by replacing the exhaustive graph search for synthesis with random simulation and reinforcement learning, i.e. the model's state space is explored randomly using a reinforcement learning algorithm to generate a strategy, which is then verified by model checking to ensure its correctness.

### 5.3 Smart railway systems and stations of the future

Traditionally, railway stations have a private energy distribution and communication system, mainly to ensure uninterrupted power supply and security. However, there are important drawbacks, such as prohibiting proper integration in the smart cities concept based on exploiting information between different transport systems (e.g. bike sharing, car sharing, urban transport) and failing to benefit from state-of-the-art energy-saving techniques. In STINGRAY and SmaRIERS, we aimed to enhance the integration of railway stations into smart cities of the future and study advanced energy-saving techniques. In this section, we report work on two project case studies on smart energy consumption and on a railway positioning case study from SISTER, which investigated the substitution of track circuits installed on tramway lines with an integrated on-board solution including (satellite) positioning.

**Fig. 12** Gas switch heating (Image by FabiBerg/Wikimedia Commons distributed under CC BY-SA 3.0 license)

First, we compared two formal methods tools [117] by experimenting the modelling and analysis features of UPPAAL SMC and Möbius [118]. As for the systematic tool evaluation described in Section 4, we applied both tools to the same case study. This time, the features on which we compared them ranged from modelling (e.g. communication primitives and delay distributions) to property specification (e.g. measures of interest) and experiments and presentation of results (e.g. experiment parameter set-up).

The case study is a cyber-physical system from the railway domain, viz., a railroad switch heater that is meant to assure the correct operation of switches in case of snow and ice (cf. Fig. 12) through a central control unit in charge of managing policies of energy consumption while satisfying reliability constraints. It contains physical components (the heater), cyber components (the heating policies and the related coordinator), stochastic aspects (failure events and weather forecasts), and logical/physical dependencies. The models and analyses with stochastic activity networks and Möbius were originally presented in [119], whereas those with stochastic hybrid automata and UPPAAL SMC were originally presented in [120].

To improve their usability, we concluded that Möbius could provide primitive support for non-anonymous replicas and channel communication (a suggestion which has since been implemented by the tool's developers) and primitive support for ordinary differential equations, whereas UPPAAL SMC could provide primitive support for batches of experiments with different parameters (a suggestion which was well received by one of the tool's main developers) and further distribution delays (e.g. deterministic time).

Second, we addressed the design of future smart station lighting management applications [121], which aim to reduce station illumination whenever (time) and wherever (space) possible while guaranteeing minimum illumination levels as required by current legislation. The (ceiling) lights (LEDs) along a station's platforms are equipped with a data acquisition module called MADILL. A C-MAD unit collects the

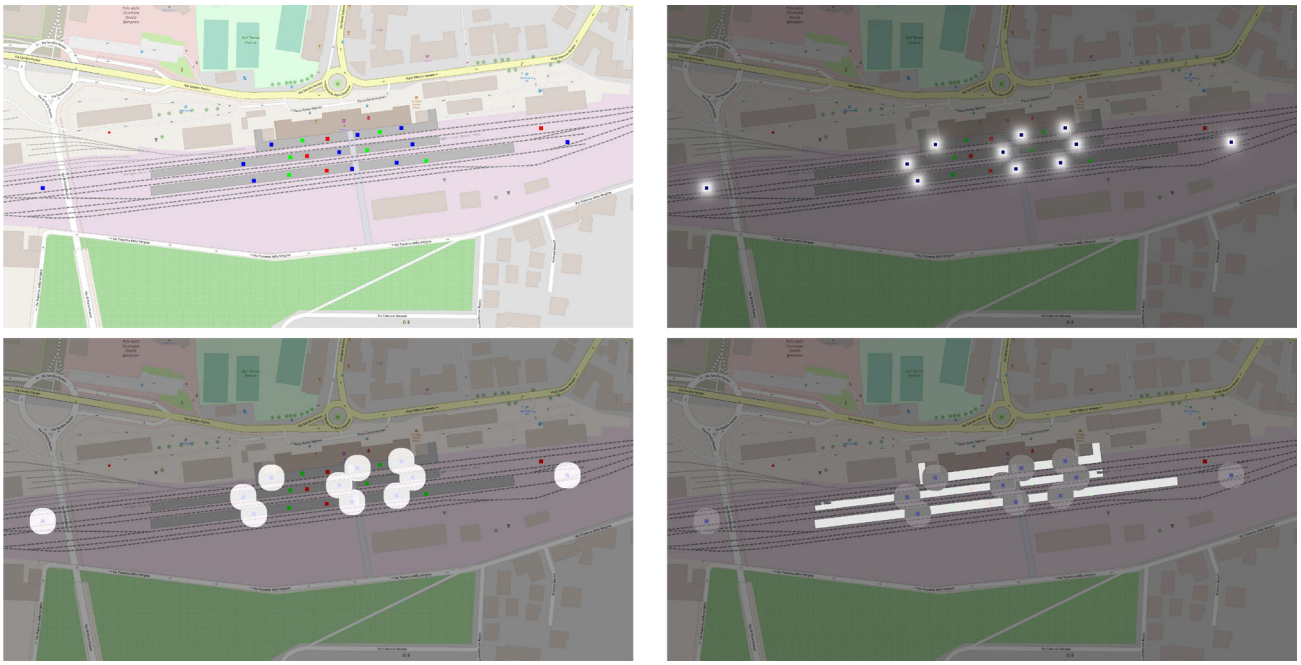

**Fig. 13** Illustration of an experiment aimed at identifying poorly illuminated platform areas. **Top-left**: Pistoia station. Blue squares: a design with MADILL units, clearly insufficient in number. Red squares: some C-MAD units. Green squares: indicate the platforms open to the public. **Top-right**: illumination computed using an attenuation formula with VoxLogicA (overlay is made with an external program). **Bottom-left**: by a threshold on the illumination value, areas that are sufficiently illuminated have been computed (output from VoxLogicA). **Bottom-right**: the parts of the platforms that are not sufficiently illuminated are computed using VoxLogicA (shown in white). (Color figure online, reprinted from [121].)

messages from each MADILL and it is equipped with brightness sensors and commands to switch lights on, off or dim them—either individually or for groups of lights.

We considered user-experience-related requirements like "*passengers should always be able to rely on an illuminated pathway when getting on or off a train, from the main entrance, to the platform*", to avoid passengers transiting or waiting in non-illuminated areas, with the associated risks (e.g. theft or injury), or "*there should be an illumination level greater than x on platforms where a train is about to arrive, even if the train is late*". Such requirements are inherently spatial or spatio-temporal, as they deal with the possibly complex reachability relations and pathways of a train station. We envisioned how to tackle these concretely by applying spatial model checking techniques and the VoxLogicA model checking tool [122, 123], as illustrated in Figure 13 through images produced by VoxLogicA.

Third, we built a toolchain from CATLib [124] and VoxLogicA for modelling and solving mobility problems in which positioning plays an important role, introducing an original approach to combine strategy synthesis with spatial model checking [125]. We performed an experiment involving a realistic scenario from the railway domain [24]. This scenario, depicted in Fig. 14, concerns an ATP system in which physical track circuits detecting the occupancy of portions of the railway track are substituted by virtual track circuits (VTCs) given as positions on a map. One junction area (commanded by one interlocking) is composed of two VTCs and there is one train outside the junction area and one train inside, viz., Train 2 traverses its assigned route while Train 1 is waiting at a red signal for its route to be assigned. VTC0 is used to detect route occupation, whereas VTC1 is used to detect a route's release.

Initially, both trains are located behind the semaphore. Then, the first train communicates its route to the interlocking, which sets the route, possibly moving the junction point. Once set, the interlocking signals this to the train by opening the semaphore, after which the train enters the junction point and the semaphore is closed again. While the first train traverses its route, the second train stops at the (closed) semaphore to ask for its route. Once assigned, the junction point moves, and the semaphore opens only after the first train has exited the junction area. Otherwise, moving the junction point could cause the derailment of the train inside the junction area.

We modelled this scenario using images. In our experiment, the PNG map image used as planimetry is derived from the junction scenario in Fig. 14. The image representing the initial configuration of this experiment is depicted in Fig. 15 (left): both trains (green and red) are waiting to enter

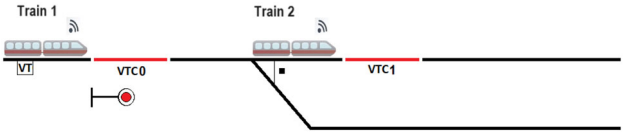


**Fig. 14** The railway scenario from [24], with Train 1 waiting to enter a junction area, while Train 2 is traversing it. (Color figure online, reprinted from [126].)

**Fig. 15** Illustration of an experiment from the railway domain. **Left**: initial configuration of the scenario. **Middle**: the junction area emphasised. **Right**: a final configuration with both trains exiting the junction from the same route. (Color figure online, reprinted (and adapted) from [125].)

the junction area, whose access is signalled by a semaphore (blue); whenever the semaphore is present (blue), it models the red signal (closed), while its absence models the green signal. The VTC0 detecting the occupancy of the junction area is at the right side of the semaphore, as emphasised (in purple) in Fig. 15 (middle), while the last two pixels on the right borders of the image are both assumed to contain VTCs for realising the junction. Figure 15 (right) shows a final configuration in which both trains have exited the junction area from the same exit.

We successfully synthesised a strategy such that both trains can safely traverse the junction area (for a game in which the semaphore is the player and the opponents are the trains): once the semaphore is opened and the first train enters the junction area, the semaphore is closed prior to the second train entering the junction area. After the first train has exited the junction area, the semaphore is opened again, and the second train can reach the exit of the junction area. In this case, trains cannot travel backwards (from right to left). The successful final states are those where both trains have exited the junction area, i.e. they both reach the right side of the image, without traversing forbidden states. The forbidden states are those in which both trains are inside the junction area as well as those in which one train is inside the junction area, the other train is before the semaphore, and the semaphore is open (thus creating a hazardous scenario).

## 6 Discussion and future work

Railway transportation by train, metro or tram is among the most energy-efficient and environmentally friendly means of transportation. In the near future, the railway domain is expected to contribute significantly to the European Green Deal by improved *digitalisation* and *big data analytics*[16].

To start with the latter, as confirmed by a recent survey of AI applications in railway systems covering 139 papers from the literature for the period 2010–2020 [127], AI research in railways is still in its early stages, yet major efforts are being dedicated to the use of AI for innovating rail maintenance policies. This is not surprising, since the deployment of effective and efficient maintenance strategies holds the promise of minimising the downtime of equipment and thus reducing the high operational costs that are characteristic of the railway domain.

In particular, *predictive maintenance* aims to detect failures before they actually occur. The development of effective and efficient predictive maintenance solutions for the railway domain is a challenging and emerging research field, as witnessed by 24 papers in the literature for the period 2016–2021 [128]. Typically, one distinguishes several classes of predictive maintenance on the basis of how the data is collected and exploited to implement the prognostics application [129, 130], i.e. (physical) model-based [131, 132], data-driven-based [133], knowledge-based [134] and digital twin-based [135]. To produce the predictions, model-based approaches use the provided input data on a previously defined physical or mathematical model; data-driven approaches use a statistical model inferred from the data that was available at the time of the training of the prognostics application; knowledge-based approaches use domain knowledge (e.g. ontologies) or expertise of the system; and digital twin-based approaches use a real-time digital representation of the physical system to generate data imitating the real events.

In the context of the NextGenerationEU project on Rail Transportation of the National Centre for Sustainable Mobility (MOST), we have started to work on predictive maintenance approaches based on big data analytics for safe and sustainable infrastructure maintenance. More effort is needed to drive this research further.

We are currently developing a data-driven approach to define cost-effective predictive maintenance strategies for on-board equipment, focusing for now on the railway Traction Control Unit (TCU) and the Door Control Unit (DCU) of local commuter trains. The goal is to identify operational anomalies and potential defects in the data logs of these on-board units, enabling the scheduling of maintenance ahead of time [136].

The traction system is important since it enables adherence to acceleration and deceleration values mandated by the regulations, whereas the door system is related to the safety, security and efficiency of railway operations, implying that

[16] https://transport.ec.europa.eu/system/files/2021-04/2021-mobility-strategy-and-action-plan.pdf

its failure can lead to operational disruptions or delays that may propagate through the railway network, thus causing economic loss and bad social reputation. Surprisingly, the door system is responsible for 30–60% of the total failures in railway vehicles [137]. This high failure rate is due to the complexity of the system's mechatronic structure as well as to the high stress during its lifecycle (e.g. frequent opening and closing due to high passenger flow, in particular on local commuter trains).

Other project partners work on the development of model-based prognostic techniques for the TCU and DCU. The goal is to identify correlations of diagnostic events in the data to extract a fault tree, enabling analysis with the MDE tool FaultFlow [138].

While data-driven approaches require huge amounts of data to correctly infer a statistical model, model-based approaches are less dependent on data. Yet model-based solutions can become quite complex in case the modelled system is complex, whereas the domain-agnostic nature of data-driven solutions guarantees instant applicability on the data without the need of a model or detailed knowledge of the system.

Finally, one of the current challenges for the use of improved digitalisation to further increase the levels of safety, security, reliability and comfort in the railway domain concerns the extension of formal methods and tools to cope with AI-based systems, such as equipping verification tools with certificate generation in case of a "yes"-answer for improved trustworthiness (e.g. through *explainability* or *certified model checking*) and the integration of AI-based systems into the CENELEC standards [139, 140]. There are, however, several limiting factors to an increased adoption of formal methods and tools in industry, not limited to the railway domain nor to the integration of AI-based systems. More effort is needed to tackle these limitations.

First, we need to close the gap between the semi-formal models that are popular and suitable to communicate with industry and the formal models that are required to apply formal methods tools in safety-critical domains, like railways.

Furthermore, each formal methods tool currently requires modelling expertise in a different input language and expert knowledge of different analysis techniques, making it important to pick the right tool based on input from industry and the requirements at hand. While it is no problem, or even a plus, for our FMT lab (and for other research groups on formal methods and tools worldwide) to host researchers with many different and complementary kinds of expertise, in-house expertise on formal methods tools is rare in industry (companies like AWS and ASML are notable exceptions [26, 141–145]).

## 7 Conclusion

We presented and discussed the results and lessons learned from two major empirical studies that we performed on the use of formal methods and tools in the railway industry. These papers contain extensive threats to validity sections which we decided not to repeat here. We also presented and discussed lessons learned from several experiences in projects involving railway practitioners in which we successfully applied formal methods and tools. Throughout these presentations, we provide pointers to steer future research on the application of formal methods and tools in the railway domain, after which we discuss a number of specific challenges for future work.

The road to the successful application of formal methods and tools, possibly integrated with AI-based systems, in industries like railways, that we foresee is to start from the basis. Together with many experts [25], we advocate a prominent role of formal methods in computer science education [146], for several reasons. First, because of the importance of formal methods *thinking* in computer science education [147], since it provides the necessary rigour in reasoning on correctness and the fundamental skill of *abstraction* [148, 149], which is at the heart of MDE [150, 151]. Then, because of the importance of *knowing* formal methods [152], since the skills and knowledge acquired from studying formal methods provide the indispensable solid foundation that forms the backbone of computer science practice. This is confirmed by the recent increase in *using* formal methods in industry [25, 26], not limited to the safety-critical domain.

**Acknowledgements** I would like to thank all my co-authors of the work recollected in this contribution: Davide Basile, Laura Bussi, Vincenzo Ciancia, Felicita Di Giandomenico, Alessandro Fantechi, Alessio Ferrari, Stefania Gnesi, Diego Latella, Axel Legay, Mieke Massink, Franco Mazzanti, and Giorgio Spagnolo.

**Funding** Open access funding provided by ISTI - PISA within the CRUI-CARE Agreement. Part of this work was carried out within the MUR PRIN 2022 PNRR P2022A492B project ADVENTURE (ADVancEd iNtegraTed evalUation of Railway systEms) and the MOST – Sustainable Mobility National Research Center and received funding from the European Union NextGenerationEU (PIANO NAZIONALE DI RIPRESA E RESILIENZA (PNRR) – MISSIONE 4, COMPONENTE 2, INVESTIMENTO 1.4 – D.D. 1033 17/06/2022, CN00000023. This manuscript reflects only the authors' views and opinions, neither the European Union nor the European Commission can be considered responsible for them.

## References


1. Boulanger, J.-L.: CENELEC 50128 and IEC 62279 standards. Wiley, Hoboken (2015). https://doi.org/10.1002/9781119005056
2. European Committee for Electrotechnical Standardization: CENELEC EN 50128 — Railway applications–communication, signalling and processing systems–software for railway control and protection systems. https://standards.globalspec.com/std/1678027/cenelec-en-50128 (2011)
3. Guiho, G., Hennebert, C.: SACEM Software Validation. In: Proceedings of the 12th International Conference on Software Engineering (ICSE'90), pp. 186–191. IEEE, (1990). https://doi.org/10.1109/ICSE.1990.63621
4. DaSilva, C., Dehbonei, B., Mejia, F.: Formal specification in the development of industrial applications: Subway speed control system. In: Diaz, M., Groz, R. (eds.) Proceedings of the IFIP TC6/WG6.1 5th International Conference on Formal Description Techniques for Distributed Systems and Communication Protocols (FORTE'92). IFIP Transactions, vol. C-10, pp. 199–213. North-Holland, The Netherlands (1992)
5. Dollé, D., Essamé, D., Falampin, J.: B dans le transport ferroviaire: L'expérience de Siemens transportation systems. Tech. Sci. Inf. **22**(1), 11–32 (2003). https://doi.org/10.3166/tsi.22.11-32
6. Essamé, D., Dollé, D.: B in Large Scale Projects: The Canarsie Line CBTC Experience. In: Julliand, J., Kouchnarenko, O. (eds.) Proceedings of the 7th International Conference of B Users (B'07). LNCS, vol. 4355, pp. 252–254. Springer, (2007). https://doi.org/10.1007/11955757_21
7. Behm, P., Benoit, P., Faivre, A., Meynadier, J.-M.: Météor: A successful application of B in a large project. In: Wing, J.M., Woodcock, J., Davies, J. (eds.) Proceedings of the 1st World Congress on Formal Methods in the Development of Computing Systems (FM'99). LNCS, vol. 1708, pp. 369–387. Springer, (1999). https://doi.org/10.1007/3-540-48119-2_22
8. Comptier, M., Leuschel, M., Mejia, L.-F., Molinero Perez, J., Mutz, M.: Property-based modelling and validation of a CBTC zone controller in event-B. In: Collart Dutilleul, S., Lecomte, T., Romanovsky, A.B. (eds.) Proceedings of the 3rd International Conference on Reliability, Safety, and Security of Railway Systems: Modelling, Analysis, Verification, and Certification (RSSRail'19). LNCS, vol. 11495, pp. 202–212. Springer, (2019).https://doi.org/10.1007/978-3-030-18744-6_13
9. Ferrari, A., Grasso, D., Magnani, G., Fantechi, A., Tempestini, M.: The Metrô Rio case study. Sci. Comput. Program. **78**(7), 828–842 (2013). https://doi.org/10.1016/j.scico.2012.04.003
10. Chiappini, A., Cimatti, A., Macchi, L., Rebollo, O., Roveri, M., Susi, A., Tonetta, S., Vittorini, B.: Formalization and validation of a subset of the European Train Control System. In: Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering (ICSE'10), vol. 2, pp. 109–118. ACM, (2010). https://doi.org/10.1145/1810295.1810312
11. Butler, M., Hoang, T.S., Raschke, A., Reichl, K.: Introduction to the special section on the ABZ 2018 case study: hybrid ERTMS/ETCS level 3. Int. J. Softw. Tools Technol. Transf. **22**(3), 249–255 (2020). https://doi.org/10.1007/s10009-020-00562-3
12. Abrial, J.-R.: The ABZ-2018 case study with Event-B. Int. J. Softw. Tools Technol. Transf. **22**(3), 257–264 (2020). https://doi.org/10.1007/s10009-019-00525-3
13. Arcaini, P., Kofroň, J., Ježek, P.: Validation of the hybrid ERTMS/ETCS level 3 using SPIN. Int. J. Softw. Tools Technol. Transf. **22**(3), 265–279 (2020). https://doi.org/10.1007/s10009-019-00539-x
14. Cunha, A., Macedo, N.: Validating the hybrid ERTMS/ETCS level 3 concept with Electrum. Int. J. Softw. Tools Technol. Transf. **22**(3), 281–296 (2020). https://doi.org/10.1007/s10009-019-00540-4
15. Dghaym, D., Dalvandi, M., Poppleton, M., Snook, C.: Formalising the hybrid ERTMS level 3 specification in iUML-B and Event-B. Int. J. Softw. Tools Technol. Transf. **22**(3), 297–313 (2020). https://doi.org/10.1007/s10009-019-00548-w
16. Hansen, D., Leuschel, M., Körner, P., Krings, S., Naulin, T., Nayeri, N., Schneider, D., Skowron, F.: Validation and real-life demonstration of ETCS hybrid level 3 principles using a formal B model. Int. J. Softw. Tools Technol. Transf. **22**(3), 315–332 (2020). https://doi.org/10.1007/s10009-020-00551-6
17. Mammar, A., Frappier, M., Tueno Fotso, S.J., Laleau, R.: A formal refinement-based analysis of the hybrid ERTMS/ETCS level 3 standard. Int. J. Softw. Tools Technol. Transf. **22**(3), 333–347 (2020). https://doi.org/10.1007/s10009-019-00543-1
18. Tueno Fotso, S.J., Frappier, M., Laleau, R., Mammar, A.: Modeling the hybrid ERTMS/ETCS level 3 standard using a formal requirements engineering approach. Int. J. Softw. Tools Technol. Transf. **22**(3), 349–363 (2020). https://doi.org/10.1007/s10009-019-00542-2
19. Ahmad, E., Dong, Y., Larson, B.R., Lü, J., Tang, T., Zhan, N.: Behavior modeling and verification of movement authority scenario of Chinese Train Control System using AADL. Sci. China Inf. Sci. **58**(11), 1–20 (2015). https://doi.org/10.1007/s11432-015-5346-2
20. SAE International: Architecture analysis & design language (AADL), (2022). https://doi.org/10.4271/AS5506D
21. Jifeng, H.: From CSP to hybrid systems. In: Roscoe, A.W. (ed.) A Classical Mind: Essays in Honour of C.A.R. Hoare, pp. 171–189. Prentice Hall, Hoboken (1994)
22. Wang, S., Zhan, N., Zou, L.: An improved HHL prover: an interactive theorem prover for hybrid systems. In: Butler, M.J., Conchon, S., Zaïdi, F. (eds.) Proceedings of the 17th International Conference on Formal Engineering Methods (ICFEM'15). LNCS, vol. 9407, pp. 382–399. Springer, (2015). https://doi.org/10.1007/978-3-319-25423-4_25
23. Zhan, B., Lv, Y., Wang, S., Zhao, G., Hao, J., Ye, H., Xia, B.: Compositional verification of interacting systems using event monads. In: Andronick, J., de Moura, L. (eds.) Proceedings of the 13th International Conference on Interactive Theorem Proving (ITP'22). LIPIcs, vol. 237, pp. 33:1–33:21. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, (2022). https://doi.org/10.4230/LIPIcs.ITP.2022.33
24. Basile, D., Fantechi, A., Rucher, L., Mandò, G.: Analysing an autonomous tramway positioning system with the UPPAAL statistical model checker. Form. Asp. Comput. **33**(6), 957–987 (2021). https://doi.org/10.1007/s00165-021-00556-1
25. Garavel, H., ter Beek, M.H., van de Pol, J.: The 2020 expert survey on formal methods. In: ter Beek, M.H., Ničković, D. (eds.) Proceedings of the 25th International Conference on Formal Methods for Industrial Critical Systems (FMICS'20). LNCS, vol. 12327, pp. 3–69. Springer, (2020). https://doi.org/10.1007/978-3-030-58298-2_1
26. ter Beek, M.H., Chapman, R., Cleaveland, R., Garavel, H., Gu, R., ter Horst, I., Keiren, J.J.A., Lecomte, T., Leuschel, M., Rozier, K.Y., Sampaio, A., Seceleanu, C., Thomas, M., Willemse, T.A.C., Zhang, L.: Formal methods in industry. Form. Asp. Comput. **37**(1), 7:1–7:38 (2025). https://doi.org/10.1145/3689374
27. Hierons, R.M., Bogdanov, K., Bowen, J.P., Cleaveland, R., Derrick, J., Dick, J., Gheorghe, M., Harman, M., Kapoor, K., Krause,

P.J., Lüttgen, G., Simons, A.J.H., Vilkomir, S.A., Woodward, M.R., Zedan, H.: Using formal specifications to support testing. ACM Comput. Surv. **41**(2), 9:1–9:76 (2009). https://doi.org/10.1145/1459352.1459354
28. Dijkstra, E.W.: Structured programming. In: Buxton, J.N., Randell, B. (eds.) Software engineering techniques, pp. 84–88. NATO Science Committee, Belgium (1969)
29. Baier, C., Katoen, J.-P.: Principles of Model Checking. MIT Press, Cambridge (2008)
30. Clarke, E.M., Henzinger, T.A., Veith, H., Bloem, R. (eds.): Handbook of Model Checking. Springer, (2018). https://doi.org/10.1007/978-3-319-10575-8
31. Newborn, M.: Automated Theorem Proving. Springer, Berlin (2001). https://doi.org/10.1007/978-1-4613-0089-2
32. Robinson, J.A., Voronkov, A. (eds.): Handbook of Automated Reasoning. Elsevier, (2001). https://www.sciencedirect.com/book/9780444508133/handbook-of-automated-reasoning
33. Bertot, Y., Castéran, P.: Interactive Theorem Proving and Program Development – Coq'Art: The Calculus of Inductive Constructions. Springer, Berlin (2004). https://doi.org/10.1007/978-3-662-07964-5
34. Nipkow, T., Paulson, L.C., Wenzel, M. (eds.): Isabelle/HOL: A proof assistant for higher-order logic. LNCS, vol. 2283. Springer, (2002). https://doi.org/10.1007/3-540-45949-9
35. Platzer, A., Quesel, J.-D.: KeYmaera: A hybrid theorem prover for hybrid systems (system description). In: Armando, A., Baumgartner, P., Dowek, G. (eds.) Proceedings of the 4th International Joint Conference on Automated Reasoning (IJCAR'08). LNCS, vol. 5195, pp. 171–178. Springer, (2008). https://doi.org/10.1007/978-3-540-71070-7_15
36. Platzer, A.: Logical Foundations of Cyber-Physical Systems. Springer, Berlin (2018). https://doi.org/10.1007/978-3-319-63588-0
37. de Moura, L., Bjørner, N.: Z3: An efficient SMT solver. In: Ramakrishnan, C.R., Rehof, J. (eds.) Proceedings of the 14th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'08). LNCS, vol. 4963, pp. 337–340. Springer, (2008). https://doi.org/10.1007/978-3-540-78800-3_24
38. Holzmann, G.J.: The SPIN Model Checker: Primer and Reference Manual. Addison-Wesley, Boston (2003)
39. Cimatti, A., Clarke, E.M., Giunchiglia, E., Giunchiglia, F., Pistore, M., Roveri, M., Sebastiani, R., Tacchella, A.: NuSMV 2: An opensource tool for symbolic model checking. In: Brinksma, E., Larsen, K.G. (eds.) Proceedings of the 14th International Conference on Computer Aided Verification (CAV'02). LNCS, vol. 2404, pp. 359–364. Springer, (2002). https://doi.org/10.1007/3-540-45657-0_29
40. Cavada, R., Cimatti, A., Dorigatti, M., Griggio, A., Mariotti, A., Micheli, A., Mover, S., Roveri, M., Tonetta, S.: The nuXmv symbolic model checker. In: Biere, A., Bloem, R. (eds.) Proceedings of the 26th International Conference on Computer Aided Verification (CAV'14). LNCS, vol. 8559, pp. 334–342. Springer, (2014). https://doi.org/10.1007/978-3-319-08867-9_22
41. Leuschel, M., Butler, M.J.: ProB: an automated analysis toolset for the B method. Int. J. Softw. Tools Technol. Transf. **10**(2), 185–203 (2008). https://doi.org/10.1007/s10009-007-0063-9
42. Behrmann, G., David, A., Larsen, K.G.: A tutorial on Uppaal. In: Bernardo, M., Corradini, F. (eds.) Revised Lectures of the 4th International School on Formal Methods for the Design of Computer, Communication and Software Systems: Formal Methods for the Design of Real-Time Systems (SFM-RT'04). LNCS, vol. 3185, pp. 200–236. Springer, (2004). https://doi.org/10.1007/978-3-540-30080-9_7
43. Behrmann, G., David, A., Larsen, K.G., Håkansson, J., Pettersson, P., Yi, W., Hendriks, M.: UPPAAL 4.0. In: Proceedings of the 3rd International Conference on the Quantitative Evaluation of SysTems (QEST'06), pp. 125–126. IEEE, (2006). https://doi.org/10.1109/QEST.2006.59
44. Behrmann, G., Cougnard, A., David, A., Fleury, E., Larsen, K.G., Lime, D.: UPPAAL-Tiga: Time for playing games!. In: Damm, W., Hermanns, H. (eds.) Proceedings of the 19th International Conference on Computer Aided Verification (CAV'07). LNCS, vol. 4590, pp. 121–125. Springer, (2007). https://doi.org/10.1007/978-3-540-73368-3_14
45. David, A., Jensen, P.G., Larsen, K.G., Mikučionis, M., Taankvist, J.H.: UPPAAL STRATEGO. In: Baier, C., Tinelli, C. (eds.) Proceedings of the 21st International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'15). LNCS, vol. 9035, pp. 206–211. Springer, (2015). https://doi.org/10.1007/978-3-662-46681-0_16
46. David, A., Larsen, K.G., Legay, A., Mikučionis, M., Poulsen, D.B.: UPPAAL SMC tutorial. Int. J. Softw. Tools Technol. Transf. **17**(4), 397–415 (2015). https://doi.org/10.1007/s10009-014-0361-y
47. Gibson-Robinson, T., Armstrong, P.J., Boulgakov, A., Roscoe, A.W.: FDR3 – A modern refinement checker for CSP. In: Ábrahám, E., Havelund, K. (eds.) Proceedings of the 20th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'14). LNCS, vol. 8413, pp. 187–201. Springer, (2014). https://doi.org/10.1007/978-3-642-54862-8_13
48. Garavel, H., Lang, F., Mateescu, R., Serwe, W.: CADP 2011: a toolbox for the construction and analysis of distributed processes. Int. J. Softw. Tools Technol. Transf. **15**(2), 89–107 (2013). https://doi.org/10.1007/s10009-012-0244-z
49. Atif, M., Groote, J.F.: Understanding Behaviour of Distributed Systems Using mCRL2. Springer, Berlin (2023). https://doi.org/10.1007/978-3-031-23008-0
50. Baier, C., de Alfaro, L., Forejt, V., Kwiatkowska, M.: Model Checking Probabilistic Systems. In: Clarke, E.M., Henzinger, T.A., Veith, H., Bloem, R. (eds.) Handbook of model checking, pp. 963–999. Springer, Berlin (2018). https://doi.org/10.1007/978-3-319-10575-8_28
51. Agha, G., Palmskog, K.: A survey of statistical model checking. ACM Trans. Model. Comput. Simul. **28**(1), 6–1639 (2018). https://doi.org/10.1145/3158668
52. Legay, A., Lukina, A., Traonouez, L.-M., Yang, J., Smolka, S.A., Grosu, R.: Statistical Model Checking. In: Steffen, B., Woeginger, G.J. (eds.) Computing and software science: state of the art and perspectives, pp. 478–504. Springer, Berlin (2019). https://doi.org/10.1007/978-3-319-91908-9_23
53. Rival, X., Yi, K.: Introduction to Static Analysis. MIT Press, Cambridge (2020)
54. Cousot, P.: Principles of Abstract Interpretation. MIT Press, Cambridge (2021)
55. Peleska, J.: Industrial-strength model-based testing: state of the art and current challenges. In: Petrenko, A.K., Schlingloff, H. (eds.) Proceedings of the 8th Workshop on Model-Based Testing (MBT'13). EPTCS, vol. 111, pp. 3–28 (2013). https://doi.org/10.4204/EPTCS.111.1
56. Bucchiarone, A., Cabot, J., Paige, R.F., Pierantonio, A.: Grand challenges in model-driven engineering: an analysis of the state of the research. Softw. Syst. Model. **19**(1), 5–13 (2020). https://doi.org/10.1007/S10270-019-00773-6
57. Sen, K., Marinov, D., Agha, G.: CUTE: A concolic unit testing engine for C. In: Proceedings of the 10th European Software Engineering Conference Held Jointly with 13th ACM SIGSOFT International Symposium on Foundations of Software Engineering (ESEC/FSE'05), pp. 263–272. ACM, (2005). https://doi.org/10.1145/1081706.1081750

58. Sutton, M., Greene, A., Amini, P.: Fuzzing: Brute Force Vulnerability Discovery. Addison-Wesley, Boston (2007)
59. Ferrari, A., ter Beek, M.H.: Formal methods in railways: a systematic mapping study. ACM Comput. Surv. **55**(4), 69:1–69:37 (2023). https://doi.org/10.1145/3520480
60. Kitchenham, B.: Procedures for performing systematic reviews. Technical Report TR/SE-0401, Keele University. (2004)
61. Petersen, K., Vakkalanka, S., Kuzniarz, L.: Guidelines for conducting systematic mapping studies in software engineering: An update. Inf. Softw. Technol. **64**, 1–18 (2015). https://doi.org/10.1016/j.infsof.2015.03.007
62. Basile, D., ter Beek, M.H., Fantechi, A., Gnesi, S., Mazzanti, F., Piattino, A., Trentini, D., Ferrari, A.: On the industrial uptake of formal methods in the railway domain: a survey with stakeholders. In: Furia, C.A., Winter, K. (eds.) Proceedings of the 14th International Conference on Integrated Formal Methods (iFM'18). LNCS, vol. 11023, pp. 20–29. Springer, (2018). https://doi.org/10.1007/978-3-319-98938-9_2
63. ter Beek, M.H., Borälv, A., Fantechi, A., Ferrari, A., Gnesi, S., Löfving, C., Mazzanti, F.: Adopting formal methods in an industrial setting: the railways case. In: ter Beek, M.H., McIver, A., Oliveira, J.N. (eds.) Proceedings of the 3rd World Congress on Formal Methods: The Next 30 Years (FM'19). LNCS, vol. 11800, pp. 762–772. Springer, (2019). https://doi.org/10.1007/978-3-030-30942-8_46
64. Mazzanti, F., Ferrari, A., Spagnolo, G.O.: Towards formal methods diversity in railways: an experience report with seven frameworks. Int. J. Softw. Tools Technol. Transf. **20**(3), 263–288 (2018). https://doi.org/10.1007/s10009-018-0488-3
65. Ferrari, A., Mazzanti, F., Basile, D., ter Beek, M.H., Fantechi, A.: Comparing formal tools for system design: a judgment study. In: Proceedings of the 42nd ACM/IEEE International Conference on Software Engineering (ICSE'20), pp. 62–74. ACM, (2020). https://doi.org/10.1145/3377811.3380373
66. Ferrari, A., Mazzanti, F., Basile, D., ter Beek, M.H.: Systematic evaluation and usability analysis of formal methods tools for railway signaling system design. IEEE Trans. Softw. Eng. **48**(11), 4675–4691 (2022). https://doi.org/10.1109/TSE.2021.3124677
67. Lano, K.: The B Language and Method: a Guide to Practical Formal Development. FACIT. Springer, (1996). https://link.springer.com/book/10.1007/978-1-4471-1494-9
68. Abrial, J.-R., Hallerstede, S.: Refinement, decomposition, and instantiation of discrete models: application to Event-B. Fundam. Inform. **77**(1–2), 1–28 (2007)
69. Abrial, J.-R.: Modeling in Event-B: System and Software Engineering. Cambridge University Press, Cambridge (2010). https://doi.org/10.1017/CBO9781139195881
70. Chiappini, A., Cimatti, A., Macchi, L., Rebollo, O., Roveri, M., Susi, A., Tonetta, S., Vittorini, B.: Formalization and validation of a subset of the European Train Control System. In: Proceedings of the 32nd International Conference on Software Engineering (ICSE'10), pp. 109–118. ACM, (2010). https://doi.org/10.1145/1810295.1810312
71. Cimatti, A., Roveri, M., Susi, A., Tonetta, S.: Formalizing requirements with object models and temporal constraints. Softw. Syst. Model. **10**(2), 147–160 (2011). https://doi.org/10.1007/s10270-009-0130-7
72. Cimatti, A., Roveri, M., Susi, A., Tonetta, S.: Validation of requirements for hybrid systems: a formal approach. ACM Trans. Softw. Eng. Methodol. **21**(4), 22:1–22:34 (2012). https://doi.org/10.1145/2377656.2377659
73. Bosschaart, M., Quaglietta, E., Janssen, B., Goverde, R.M.P.: Efficient formalization of railway interlocking data in RailML. Inf. Syst. **49**, 126–141 (2015). https://doi.org/10.1016/j.is.2014.11.007
74. Hamid, B., Pérez, J.: Supporting pattern-based dependability engineering via model-driven development: approach, tool-support and empirical validation. J. Syst. Softw. **122**, 239–273 (2016). https://doi.org/10.1016/j.jss.2016.09.027
75. Wu, D., Schnieder, E.: Scenario-based system design with colored Petri nets: an application to train control systems. Softw. Syst. Model. **17**(1), 295–317 (2018). https://doi.org/10.1007/s10270-016-0517-1
76. Snook, C.F., Hoang, T.S., Dghaym, D., Salehi Fathabadi, A., Butler, M.J.: Domain-specific scenarios for refinement-based methods. J. Syst. Archit. (2021). https://doi.org/10.1016/j.sysarc.2020.101833
77. Badeau, F., Amelot, A.: Using B as a high level programming language in an industrial project: Roissy VAL. In: Treharne, H., King, S., Henson, M.C., Schneider, S.A. (eds.) Proceedings of the 4th International Conference of B and Z Users (ZB'05). LNCS, vol. 3455, pp. 334–354. Springer, (2005). https://doi.org/10.1007/11415787_20
78. Leuschel, M., Falampin, J., Fritz, F., Plagge, D.: Automated property verification for large scale B models with ProB. Form. Asp. Comput. **23**(6), 683–709 (2011). https://doi.org/10.1007/s00165-010-0172-1
79. Abo, R., Voisin, L.: Formal Implementation of data validation for railway safety-related systems with OVADO. In: Counsell, S., Núñez, M. (eds.) Proceedings of the SEFM 2013 Collocated Workshops: BEAT2, WS-FMDS, FM-RAIL-Bok, MoKMaSD, and OpenCert. LNCS, vol. 8368, pp. 221–236. Springer, (2013). https://doi.org/10.1007/978-3-319-05032-4_17
80. James, P., Moller, F., Nga, N.H., Roggenbach, M., Schneider, S.A., Treharne, H.: Techniques for modelling and verifying railway interlockings. Int. J. Softw. Tools Technol. Transf. **16**(6), 685–711 (2014). https://doi.org/10.1007/s10009-014-0304-7
81. Fürst, A., Hoang, T.S., Basin, D.A., Sato, N., Miyazaki, K.: Large-scale system development using abstract data types and refinement. Sci. Comput. Program. **131**, 59–75 (2016). https://doi.org/10.1016/j.scico.2016.04.010
82. Reichl, K., Fischer, T., Tummeltshammer, P.: Using formal methods for verification and validation in railway. In: Aichernig, B.K., Furia, C.A. (eds.) Proceedings of the 10th International Conference on Tests and Proofs (TAP'16). LNCS, vol. 9762, pp. 3–13. Springer, (2016). https://doi.org/10.1007/978-3-319-41135-4_1
83. Comptier, M., Déharbe, D., Molinero Perez, J., Mussat, L., Thibaut, P., Sabatier, D.: Safety analysis of a CBTC system: a rigorous approach with Event-B. In: Fantechi, A., Lecomte, T., Romanovsky, A.B. (eds.) Proceedings of the 2nd International Conference on Reliability, Safety, and Security of Railway Systems: Modelling, Analysis, Verification, and Certification (RSSRail'17). LNCS, vol. 10598, pp. 148–159. Springer, (2017). https://doi.org/10.1007/978-3-319-68499-4_10
84. Idani, A., Ledru, Y., Ait Wakrime, A., Ben Ayed, R., Collart Dutilleul, S.: Incremental development of a safety critical system combining formal methods and DSMLs: application to a railway system. In: Larsen, K.G., Willemse, T.A.C. (eds.) Proceedings of the 24th International Conference on Formal Methods for Industrial Critical Systems (FMICS'19). LNCS, vol. 11687, pp. 93–109. Springer, (2019). https://doi.org/10.1007/978-3-030-27008-7_6
85. Berger, U., James, P., Lawrence, A., Roggenbach, M., Seisenberger, M.: Verification of the European rail traffic management system in real-time Maude. Sci. Comput. Program. **154**, 61–88 (2018). https://doi.org/10.1016/j.scico.2017.10.011
86. Bao, Y., Chen, M., Zhu, Q., Wei, T., Mallet, F., Zhou, T.: Quantitative performance evaluation of uncertainty-aware hybrid AADL designs using statistical model checking. IEEE Trans. Comput. Aided. Des. Integr. Circuits Syst. **36**(12), 1989–2002 (2017). https://doi.org/10.1109/TCAD.2017.2681076

87. Wang, Y., Chen, L., Kirkwood, D., Fu, P., Lv, J., Roberts, C.: Hybrid online model-based testing for communication-based train control systems. IEEE Intell. Transp. Syst. Mag. **10**(3), 35–47 (2018). https://doi.org/10.1109/MITS.2018.2842230
88. Kitchenham, B., Linkman, S., Law, D.: DESMET: a methodology for evaluating software engineering methods and tools. Comput. Control. Eng. J. **8**(3), 120–126 (1997). https://doi.org/10.1049/cce:19970304
89. Ferrari, A., ter Beek, M.H., Mazzanti, F., Basile, D., Fantechi, A., Gnesi, S., Piattino, A., Trentini, D.: Survey on formal methods and tools in railways: the ASTRail approach. In: Collart Dutilleul, S., Lecomte, T., Romanovsky, A. (eds.) Proceedings of the 3rd International Conference on Reliability, Safety, and Security of Railway Systems: Modelling, Analysis, Verification, and Certification (RSSRail'19). LNCS, vol. 11495, pp. 226–241. Springer, (2019). https://doi.org/10.1007/978-3-030-18744-6_15
90. Dabney, J.B., Harman, T.L.: Mastering Simulink. Pearson, (2003). https://www.pearson.com/en-us/subject-catalog/p/mastering-simulink/P200000003257/9780131424777
91. Jensen, K., Kristensen, L.M., Wells, L.: Coloured Petri Nets and CPN Tools for modelling and validation of concurrent systems. Int. J. Softw. Tools Technol. Transf. **9**(3–4), 213–254 (2007). https://doi.org/10.1007/s10009-007-0038-x
92. de Moura, L.M., Owre, S., Rueß, H., Rushby, J.M., Shankar, N., Sorea, M., Tiwari, A.: SAL 2. In: Alur, R., Peled, D.A. (eds.) Proceedings of the 16th International Conference on Computer Aided Verification (CAV'04). LNCS, vol. 3114, pp. 496–500. Springer, (2004). https://doi.org/10.1007/978-3-540-27813-9_45
93. Lamport, L.: Specifying Systems: The TLA+ Language and Tools for Hardware and Software Engineers. Addison-Wesley, (2002). https://www.pearson.com/en-us/subject-catalog/p/specifying-systems-the-tla-language-and-tools-for-hardware-and-software-engineers/P200000009296/9780321143068
94. ter Beek, M.H., Fantechi, A., Gnesi, S., Mazzanti, F.: A state/event-based model-checking approach for the analysis of abstract system properties. Sci. Comput. Program. **76**(2), 119–135 (2011). https://doi.org/10.1016/j.scico.2010.07.002
95. Pohl, K.: Requirements Engineering: Fundamentals, Principles, and Techniques. Springer, Berlin (2010)
96. Scupin, R.: The KJ method: A technique for analyzing data derived from Japanese ethnology. Hum. Organ. **56**(2), 233–237 (1997). https://doi.org/10.17730/humo.56.2.x335923511444655
97. Brooke, J.: SUS: A 'quick and dirty' usability scale. In: Jordan, P.W., Thomas, B., Weerdmeester, B.A., McClelland, I.L. (eds.) Usability Evaluation in Industry, pp. 189–194. CRC, (1996). Chap. 21. https://doi.org/10.1201/9781498710411
98. Brooke, J.: SUS: A retrospective. J. Usability Stud. **8**(2), 29–40 (2013). https://doi.org/10.5555/2817912.2817913
99. Bangor, A., Kortum, P.T., Miller, J.T.: An empirical evaluation of the system usability scale. Int. J. Hum. Comput. Interact. **24**(6), 574–594 (2008). https://doi.org/10.1080/10447310802205776
100. Furness, N., van Houten, H., Arenas, L., Bartholomeus, M.: ERTMS level 3: the game-changer. IRSE News **232**, 2–9 (2017). https://www.irse.nl/resources/170314-ERTMS-L3-The-gamechanger-from-IRSE-News-Issue-232.pdf
101. Marais, J., Beugin, J., Berbineau, M.: A survey of GNSS-based research and developments for the European railway signaling. IEEE Trans. Intell. Transp. Syst. **18**(10), 2602–2618 (2017). https://doi.org/10.1109/TITS.2017.2658179
102. Basile, D., ter Beek, M.H., Ferrari, A., Legay, A.: Exploring the ERTMS/ETCS full moving block specification: an experience with formal methods. Int. J. Softw. Tools Technol. Transf. **24**(3), 351–370 (2022). https://doi.org/10.1007/S10009-022-00653-3
103. Biagi, M., Carnevali, L., Paolieri, M., Vicario, E.: Performability evaluation of the ERTMS/ETCS - Level 3. Transp. Res. C-Emer. **82**, 314–336 (2017). https://doi.org/10.1016/j.trc.2017.07.002
104. Bartholomeus, M., Luttik, B., Willemse, T.: Modeling and analysing ERTMS hybrid level 3 with the mCRL2 toolset. In: Howar, F., Barnat, J. (eds.) Proceedings of the 23rd International Conference on Formal Methods for Industrial Critical Systems (FMICS'18). LNCS, vol. 11119, pp. 98–114. Springer, (2018). https://doi.org/10.1007/978-3-030-00244-2_7
105. Basile, D., ter Beek, M.H., Ciancia, V.: Statistical model checking of a moving block railway signalling scenario with Uppaal SMC: experience and outlook. In: Margaria, T., Steffen, B. (eds.) Proceedings of the 8th International Symposium on Leveraging Applications of Formal Methods, Verification and Validation: Verification (ISoLA'18). LNCS, vol. 11245, pp. 372–391. Springer, (2018). https://doi.org/10.1007/978-3-030-03421-4_24
106. Basile, D., ter Beek, M.H., Ferrari, A., Legay, A.: Modelling and analysing ERTMS L3 moving block railway signalling with Simulink and Uppaal SMC. In: Larsen, K.G., Willemse, T. (eds.) Proceedings of the 24th International Conference on Formal Methods for Industrial Critical Systems (FMICS'19). LNCS, vol. 11687, pp. 1–21. Springer, (2019). https://doi.org/10.1007/978-3-030-27008-7_1
107. Douglass, B.P.: Real-Time UML. In: Damm, W., Olderog, E.-R. (eds.) Proceedings of the 7th International Symposium on Formal Techniques in Real-Time and Fault-Tolerant Systems (FTRTFT'02). LNCS, vol. 2469, pp. 53–70. Springer, (2002). https://doi.org/10.1007/3-540-45739-9_4
108. Selic, B.: The real-time UML standard: definition and application. In: Proceedings of the Design, Automation and Test in Europe Conference and Exhibition (DATE'02), pp. 770–772 (2002). https://doi.org/10.1109/DATE.2002.998385
109. Harel, D.: Statecharts: a visual formalism for complex systems. Sci. Comput. Program. **8**(3), 231–274 (1987). https://doi.org/10.1016/0167-6423(87)90035-9
110. Filipovikj, P., Mahmud, N., Marinescu, R., Seceleanu, C., Ljungkrantz, O., Lönn, H.: Simulink to UPPAAL statistical model checker: Analyzing Automotive Industrial Systems. In: Fitzgerald, J., Heitmeyer, C., Gnesi, S., Philippou, A. (eds.) Proceedings of the 21st International Symposium on Formal Methods (FM'16). LNCS, vol. 9995, pp. 748–756. Springer, (2016). https://doi.org/10.1007/978-3-319-48989-6_46
111. Jin, Y., Xie, G., Chen, P., Hei, X., Ji, W., Zhao, J.: High-speed train emergency brake modeling and online identification of time-varying parameters. Math. Probl. Eng. 2020 (2020). https://doi.org/10.1155/2020/3872852
112. Basile, D., ter Beek, M.H., Legay, A.: Strategy synthesis for autonomous driving in a moving block railway system with Uppaal Stratego. In: Gotsman, A., Sokolova, A. (eds.) Proceedings of the 40th IFIP WG 6.1 International Conference on Formal Techniques for Distributed Objects, Components, and Systems (FORTE'20). LNCS, vol. 12136, pp. 3–21. Springer, (2020). https://doi.org/10.1007/978-3-030-50086-3_1
113. Jaeger, M., Jensen, P.G., Larsen, K.G., Legay, A., Sedwards, S., Taankvist, J.H.: Teaching Stratego to play ball: optimal synthesis for continuous space MDPs. In: Chen, Y.-F., Cheng, C.-H., Esparza, J. (eds.) Proceedings of the 17th International Symposium on Automated Technology for Verification and Analysis (ATVA'19). LNCS, vol. 11781, pp. 81–97. Springer, (2019).https://doi.org/10.1007/978-3-030-31784-3_5
114. Gu, R., Jensen, P.G., Poulsen, D.B., Seceleanu, C., Enoiu, E., Lundqvist, K.: Verifiable strategy synthesis for multiple autonomous agents: a scalable approach. Int. J. Softw. Tools Technol. Transf. **24**(3), 395–414 (2022). https://doi.org/10.1007/s10009-022-00657-z
115. Gu, R., Jensen, P.G., Seceleanu, C., Enoiu, E., Lundqvist, K.: Correctness-guaranteed strategy synthesis and compression for multi-agent autonomous systems. Sci. Comput. Program. 224 (2022). https://doi.org/10.1016/J.SCICO.2022.102894

116. Gu, R.: Formal methods for scalable synthesis and verification of autonomous systems: mission planning and collision avoidance. PhD thesis, Mälardalen University, (2022). https://urn.kb.se/resolve?urn=urn:nbn:se:mdh:diva-58086
117. Basile, D., ter Beek, M.H., Di Giandomenico, F., Fantechi, A., Gnesi, S., Spagnolo, G.O.: 30 years of simulation-based quantitative analysis tools: a comparison experiment between Möbius and Uppaal SMC. In: Margaria, T., Steffen, B. (eds.) Proceedings of the 9th International Symposium on Leveraging Applications of Formal Methods, Verification and Validation: Verification (ISoLA'20). LNCS, vol. 12476, pp. 368–384. Springer, (2020). https://doi.org/10.1007/978-3-030-61362-4_21
118. Clark, G., Courtney, T., Daly, D., Deavours, D., Derisavi, S., Doyle, J.M., Sanders, W.H., Webster, P.: The Möbius modeling tool. In: Proceedings of the 9th International Workshop on Petri Nets and Performance Models (PNPM'01), pp. 241–250. IEEE, (2001). https://doi.org/10.1109/PNPM.2001.953373
119. Basile, D., Chiaradonna, S., Di Giandomenico, F., Gnesi, S.: A stochastic model-based approach to analyse reliable energy-saving rail road switch heating systems. J. Rail Transp. Plan. Manag. **6**(2), 163–181 (2016). https://doi.org/10.1016/j.jrtpm.2016.03.003
120. Basile, D., Di Giandomenico, F., Gnesi, S.: Statistical model checking of an energy-saving cyber-physical system in the railway domain. In: Proceedings of the 32nd Symposium on Applied Computing (SAC'17), pp. 1356–1363. ACM, (2017). https://doi.org/10.1145/3019612.3019824
121. ter Beek, M.H., Ciancia, V., Latella, D., Massink, M., Spagnolo, G.O.: Spatial model checking for smart stations: research challenges. In: Lluch Lafuente, A., Mavridou, A. (eds.) Proceedings of the 26th International Conference on Formal Methods for Industrial Critical Systems (FMICS'21). LNCS, vol. 12863, pp. 39–47. Springer, (2021). https://doi.org/10.1007/978-3-030-85248-1_3
122. Belmonte, G., Ciancia, V., Latella, D., Massink, M.: VoxLogicA: A spatial model checker for declarative image analysis. In: Vojnar, T., Zhang, L. (eds.) Proceedings of the 25th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'19). LNCS, vol. 11427, pp. 281–298. Springer, (2019). https://doi.org/10.1007/978-3-030-17462-0_16
123. Ciancia, V., Belmonte, G., Latella, D., Massink, M.: A hands-on introduction to spatial model checking using VoxLogicA. In: Laarman, A., Sokolova, A. (eds.) Proceedings of the 27th International Symposium on Model Checking Software (SPIN'21). LNCS, vol. 12864, pp. 22–41. Springer, (2021). https://doi.org/10.1007/978-3-030-84629-9_2
124. Basile, D., ter Beek, M.H.: Contract automata library. Sci. Comput. Program. 221 (2022). https://doi.org/10.1016/j.scico.2022.102841
125. Basile, D., ter Beek, M.H., Bussi, L., Ciancia, V.: A toolchain for strategy synthesis with spatial properties. Int. J. Softw. Tools Technol. Transf. **25**(5), 641–658 (2023). https://doi.org/10.1007/S10009-023-00730-1
126. Basile, D., ter Beek, M.H., Carnevali, L., Chiaradonna, S., Di Giandomenico, F., Fantechi, A., Gori, G.: An integrated perspective on the evaluation of complex railway systems. In: Margaria, T., Steffen, B. (eds.) Proceedings of the 12th International Symposium on Leveraging Applications of Formal Methods, Verification and Validation (ISoLA'24). LNCS, vol. 15223, pp. 190–207. Springer, (2024). https://doi.org/10.1007/978-3-031-75390-9_13
127. Tang, R., De Donato, L., Bešinović, N., Flammini, F., Goverde, R.M.P., Lin, Z., Liu, R., Tang, T., Vittorini, V., Wang, Z.: A literature review of artificial intelligence applications in railway systems. Transp. Res. C-Emer. 140 (2022). https://doi.org/10.1016/j.trc.2022.103679
128. Binder, M., Mezhuyev, V., Tschandl, M.: Predictive maintenance for railway domain: a systematic literature review. IEEE Eng. Manag. Rev. **51**(2), 120–140 (2023). https://doi.org/10.1109/EMR.2023.3262282
129. Fidma, M.A., Hamour, N., Ouchani, S., Benslimane, S.M.: Predictive maintenance approaches in Industry 4.0: a systematic literature review. In: Proceedings of the 31st IEEE International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE'23), pp. 1–6. IEEE, (2023). https://doi.org/10.1109/WETICE57085.2023.10477802
130. Hafsi, M., Hamour, N., Ouchani, S.: Predictive maintenance for smart industrial systems: a roadmap. In: Procedia Comput. Sci. **220**, 645–650 (2023). Proceedings of the 6th International Conference on Emerging Data and Industry 4.0 (EDI40'23). https://doi.org/10.1016/J.PROCS.2023.03.082
131. Longo, N., Serpi, V., Jacazio, G., Sorli, M.: Model-based predictive maintenance techniques applied to automotive industry. In: Proceedings of the 4th European Conference of the Prognostics and Health Management Society (PHME'18), vol. 4. PHM Society, (2018). https://doi.org/10.36001/phme.2018.v4i1.353
132. Meghoe, A.A.: Physical model-based predictive maintenance for rail infrastructure. PhD thesis, University of Twente, (2019). https://doi.org/10.3990/1.9789036548786
133. Zhang, W., Yang, D., Wang, H.: Data-driven methods for predictive maintenance of industrial equipment: a survey. IEEE Syst. J. **13**(3), 2213–2227 (2019). https://doi.org/10.1109/JSYST.2019.2905565
134. Cao, Q., Zanni-Merk, C., Samet, A., Reich, C., Beckmann, A., Giannetti, C., de Beuvron, F.: KSPMI: A knowledge-based system for predictive maintenance in Industry 4.0. Robot. Comput.-Integr Manuf. 74 (2022). https://doi.org/10.1016/J.RCIM.2021.102281
135. van Dinter, R., Tekinerdogan, B., Catal, C.: Predictive maintenance using digital twins: a systematic literature review. Inf. Softw. Technol. 151 (2022). https://doi.org/10.1016/J.INFSOF.2022.107008
136. Ferdous, R., Spagnolo, G., Borselli, A., Rota, L., Ferrari, A.: Identifying maintenance needs with machine learning: a case study in railways. In: Proceedings of the 32nd International Requirements Engineering Conference Workshops (REW'24), pp. 22–25. IEEE, (2024). https://doi.org/10.1109/REW61692.2024.00008
137. Cauffriez, L., Loslever, P., Caouder, N., Turgis, F., Copin, R.: Robustness study and reliability growth based on exploratory design of experiments and statistical analysis: a case study using a train door test bench. J. Adv. Manuf. Technol. **66**(1–4), 27–44 (2013). https://doi.org/10.1007/s00170-012-4303-0
138. Parri, J., Sampietro, S., Vicario, E.: FaultFlow: a tool supporting an MDE approach for timed failure logic analysis. In: Proceedings of the 17th European Dependable Computing Conference (EDCC'21), pp. 25–32. IEEE, (2021). https://doi.org/10.1109/EDCC53658.2021.00011
139. Seisenberger, M., ter Beek, M.H., Fan, X., Ferrari, A., Haxthausen, A.E., James, P., Lawrence, A., Luttik, B., van de Pol, J., Wimmer, S.: Safe and secure future AI-driven railway technologies: challenges for formal methods in railway. In: Margaria, T., Steffen, B. (eds.) Proceedings of the 11th International Symposium on Leveraging Applications of Formal Methods, Verification and Validation: Practice (ISoLA'22). LNCS, vol. 13704, pp. 246–268. Springer, (2022). https://doi.org/10.1007/978-3-031-19762-8_20
140. Gleirscher, M., van de Pol, J., Woodcock, J.: A manifesto for applicable formal methods. Softw. Syst. Model. **22**(6), 1737–1749 (2023). https://doi.org/10.1007/S10270-023-01124-2
141. Cook, B., Khazem, K., Kroening, D., Tasiran, S., Tautschnig, M., Tuttle, M.R.: Model checking boot code from AWS data centers. In: Chockler, H., Weissenbacher, G. (eds.) Proceedings of the 30th International Conference on Computer Aided Verifica-

tion (CAV'18). LNCS, vol. 10982, pp. 467–486. Springer, (2018). https://doi.org/10.1007/978-3-319-96142-2_28
142. Backes, J., Bolignano, P., Cook, B., Dodge, C., Gacek, A., Luckow, K.S., Rungta, N., Tkachuk, O., Varming, C.: Semantic-based automated reasoning for AWS access policies using SMT. In: Bjørner, N.S., Gurfinkel, A. (eds.) Proceedings of the 18th Conference on Formal Methods in Computer-Aided Design (FMCAD'18), pp. 1–9. IEEE, (2018). https://doi.org/10.23919/FMCAD.2018.8602994
143. Rungta, N.: A Billion SMT Queries a Day. In: Shoham, S., Vizel, Y. (eds.) Proceedings of the 34th International Conference on Computer Aided Verification (CAV'22). LNCS, vol. 13371, pp. 3–18. Springer, (2022). https://doi.org/10.1007/978-3-031-13185-1_1
144. Cook, B.: Automated reasoning's scientific frontiers (2022). https://www.amazon.science/blog/automated-reasonings-scientific-frontiers
145. Binns, L.: By computers, for computers: Improving scanner metrology software with generated code (2023). https://www.linkedin.com/pulse/computers-improving-scanner-metrology-software-code-lewis/
146. ter Beek, M., Broy, M., Dongol, B.: The Role of Formal Methods in Computer Science Education. Inroads 15(4), 58–66 (2024). https://doi.org/10.1145/3702231
147. Dongol, B., Dubois, C., Hallerstede, S., Hehner, E., Morgan, C., Müller, P., Ribeiro, L., Silva, A., Smith, G., de Vink, E.: On formal methods thinking in computer science education. Form. Asp. Comput. 37(1), 8:1–8:23 (2025). https://doi.org/10.1145/3670419
148. Kramer, J.: Is abstraction the Key to computing? Commun. ACM **50**(4), 36–42 (2007). https://doi.org/10.1145/1232743.1232745
149. Khelladi, D.E., Combemale, B., Acher, M., Barais, O.: On the power of abstraction: a model-driven co-evolution approach of software code. In: Proceedings of the 42nd International Conference on Software Engineering Track on New Ideas and Emerging Results (ICSE-NIER'20), pp. 85–88. ACM, (2020). https://doi.org/10.1145/3377816.3381727
150. Benjamin, P.C., Erraguntla, M., Delen, D., Mayer, R.J.: Simulation modeling at multiple levels of abstraction. In: Proceedings of the 30th Winter Simulation Conference (WSC'98), pp. 391–398. IEEE, (1998). https://doi.org/10.1109/WSC.1998.745013
151. Franceschini, R., Challenger, M., Cicchetti, A., Denil, J., Vangheluwe, H.: Challenges for automation in adaptive abstraction. In: Companion Proceedings of the 22nd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems (MODELS-C'19), pp. 443–448. IEEE, (2019). https://doi.org/10.1109/MODELS-C.2019.00071
152. Broy, M., Brucker, A.D., Fantechi, A., Gleirscher, M., Havelund, K., Kuppe, M.A., Mendes, A., Platzer, A., Ringert, J.O., Sullivan, A.: Does every computer scientist need to know formal methods? Form. Asp. Comput. 37(1), 6:1–6:17 (2025). https://doi.org/10.1145/3670795



**Maurice H. ter Beek** Maurice ter Beek is a Director of Research at CNR-ISTI (Pisa, Italy) and head of the Formal Methods and Tools lab. He obtained a Ph.D. at Leiden University (The Netherlands). He authored over 150 peer-reviewed papers, edited over 30 special issues of journals and proceedings, and serves on the editorial board of Formal Aspects of Computing, International Journal on Software Tools for Technology Transfer, Journal of Logical and Algebraic Methods in Programming, Science of Computer Programming, PeerJ Computer Science and ERCIM News. His research interests are formal methods and model-checking tools for the specification and verification of safety-critical software systems and communication protocols, focussing in particular on applications in service-oriented computing, software product line engineering and railway systems. He regularly serves on the PC of relevant formal methods and railway conferences like FM, FMICS, iFM, all of which he also chaired, and ABZ, FASE, FormaliSE, RSSRail, SEFM and SPIN. He is a board member of Formal Methods Europe (FME).