Systematic Evaluation and Usability Analysis of Formal Methods Tools for Railway Signaling System Design

Alessio Ferrari, Franco Mazzanti, Davide Basile, and Maurice H. ter Beek

Abstract—Formal methods and supporting tools have a long record of success in the development of safety-critical systems. However, no single tool has emerged as the dominant solution for system design. Each tool differs from the others in terms of the modeling language used, its verification capabilities and other complementary features, and each development context has peculiar needs that require different tools. This is particularly problematic for the railway industry, in which formal methods are highly recommended by the norms, but no actual guidance is provided for the selection of tools. To guide companies in the selection of the most appropriate formal methods tools to adopt in their contexts, a clear assessment of the features of the currently available tools is required.
To address this goal, this paper considers a set of 13 formal methods tools that have been used for the early design of railway systems, and it presents a systematic evaluation of such tools and a preliminary usability analysis of a subset of 7 tools, involving railway practitioners. The results are discussed considering the most desired aspects by industry and earlier related studies. While the focus is on the railway signaling domain, the overall methodology can be applied to similar contexts. Our study thus contributes with a systematic evaluation of formal methods tools and it shows that despite the poor graphical interfaces, *usability* and *maturity* of the tools are not major problems, as claimed by contributions from the literature. Instead, support for *process integration* is the most relevant obstacle for the adoption of most of the tools. Our contribution can be useful to R&D engineers from railway signaling companies and infrastructure managers, but also to tool developers and academic researchers alike.

1 Introduction

The development of railway safety-critical systems, such as platforms for on-board automatic train control [1], [2] or computer-based interlocking infrastructures to route the trains [3], [4], has to follow strict process guidelines to deliver products that are highly dependable and trustworthy [5], [6]. Formal methods are mathematics-based techniques for the specification, development and (manual or automated) verification of software and hardware systems [7], [8], and are particularly indicated when rigor is a main concern. They have a long history of over 30 years of success stories in railway applications [5], [9], [10], with several support tools available in the market, including commercial and free solutions [11], [12], [13], [14]. Furthermore, the CENELEC EN 50128 norm [15], which is the standard for the development of railway software in Europe, highly recommends the usage of formal methods for the design and verification of those products that need to meet the highest safety integrity levels.

Despite these premises, the adoption of formal methods and their supporting tools by companies is rather limited [16], [17], and railway practitioners ask for more guidance to select the most adequate formal tool, or set of tools, for their development contexts [18], [19], [20]. This is common also to other application domains. As observed by Steffen [20]: "Prospective users have a hard time to orient themselves in the current tool landscape, and even experts typically only have very partial knowledge. Thus, the need for a more systematic approach to establish the profiles of tools and methods is obvious".

Previous work on applications of formal methods to railway problems has mostly focused on reporting experiences [1], [2], [3], [4], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], [35], [36], [37], [38], [39], [40], [41], [42], [43], [44], [45], [46], [47], [48], [49]. Notable cases are the usage of the B method for developing the Line 14 of the Paris Mtro and the driverless ParisRoissy Airport shuttle [21], the use of Simulink for formal model-based development of the metro control system of Rio de Janeiro [2], and the application of NuSMV to the ERTMS/ETCS European standard for railway control and management [1].

Recently, a stream of literature also emerged on the evaluation and comparison of formal methods tools—which we abbreviate as "formal tools" in the remainder of the paper—in railways. Mazzanti et al. [50] replicates the same railway design with different tools, informally describing their peculiarities. Haxthausen et al. [51] compares two methods for the verification of an interlocking system. Basile et al. [52] consider the differences between two formalisms and their associated simulation-based tools by applying them to an industrial railway project. Within this set of comparative works, the H2020 European Project ASTRail (SAtellite-based signaling and Automation SysTems on Railways along with Formal Method and Moving Block Validation)[1], foresees a structured analysis of the formal tools' landscape and the selection of the most appropriate ones for railways [12]. To address this goal, the project includes three main empirical activities: a systematic literature review, a systematic evalu-

• *The authors are with the Institute of Information Science and Technologies (ISTI) of the Italian National Research Council (CNR), Pisa, Italy. E-mail: {alessio.ferrari,franco.mazzanti,davide.basile,maurice.terbeek}@isti.cnr.it*

Manuscript received April, 2021; revised . . .

[1] http://www.astrail.eu

ation of tools, and an experimental assessment on a railway case study. This endeavour is driven by industrial demands. Indeed, ASTRail is one of the projects of the Shift2Rail initiative[2], which is funded by the main public and private stakeholders of the European railway sector.

As part of the ASTRail project, and building on a previous exploratory judgment study [18], this work reports on the *systematic evaluation* of a representative selection of 13 formal tools that can be considered appropriate for the early design of railway signaling systems. We focus on railway signaling, given the long history of applications of formal methods in this field [9], [53]. We adapt the DESMET methodology for tool evaluation proposed by Kitchenham et al. [54]. We first select a comprehensive set of evaluation features based on a brainstorming involving both formal methods and railway experts. Then, three assessors evaluate the tools and assign values to the features. To this end, they access the documentation, run the tools and perform trials to have an informed judgment. Multiple iterations and triangulation sessions are performed to ensure homogeneity in the assessment. As usability of the tools is considered as particularly relevant by railway practitioners according to recent surveys [55], [56], a preliminary usability study is carried out involving subjects from a company. More specifically, the subjects assist to structured live demos of a subset of the tools, and fill in a usability test questionnaire. This evaluation setting ensures a trade-off between the need to capture the viewpoint of railway experts, and the time that is inherently required to learn formal tools [20]. The data of our study—including a detailed description of evaluation features, evaluation sheets for the analysed tools and data from the usability test—is shared to facilitate inspection, replication and extension [57].

Our work provides the following main contributions: i) we establish a set of features to systematically evaluate formal tools for railways, which can be adapted to other domains; ii) we assess the features on 13 different formal tools and share the tool evaluation sheets; iii) we perform an initial comparative usability study of 7 of these tools involving railway practitioners; iv) we debunk some common beliefs about formal tools, especially concerning usability and maturity.

This work can be useful to three main profiles, i.e., railway practitioners, tool developers and researchers. Through our work, these different readers can have a clear opinion about the landscape of available tools, and their salient characteristics. Target practitioners are mainly R&D engineers from railway signaling companies and infrastructure managers. Those practitioners who are interested in introducing formal tools for early design in their organization can overcome the paradox of choice highlighted by Steffen [20] by means of a structured and vendor-independent[3] overview of available solutions. This way, they can perform an informed selection without having to go through the tools' documentation. Tool developers can check their position with respect to other tools, and identify missing features that are desired by industry. Researchers can leverage our experience in terms of adopted methodology and feature set to evaluate other formal tools, also considering domains other than railways. We note that signaling engineers and managers working in production departments, and interested, e.g., in development and verification of specific interlocking installations, are not our main target audience. Indeed, this work is concerned with tools for early system design, and does not consider later development stages. In addition, our study does not address the issue of formal methods deployment, which is of primary interest for these profiles. Further investigation using context-dependent research strategies, e.g, case study or action research [58], is required to devise guidelines that can enable companies to successfully deploy formal tools in their processes.

The rest of the paper is structured as follows. Sect. 2 discusses background and related work. In Sect. 3, we illustrate the study design, while in Sect. 5 we report the results. Sect. 6 discusses implications. Sect. 4 reports threats to validity and Sect. 7 provides some final remarks.

2 Background and Related Work

2.1 The Railway Domain and the ASTRail Project

The railway domain is characterized by a stringent safety culture, supported by rigorous development practices that are oriented to prevent catastrophic system failures and consequent accidents. The CENELEC EN 50128 standard for the development of software for railway control and protection systems [15] considers formal methods as highly recommended practices when developing platforms of the highest Safety Integrity Levels (SIL-3/4). Given these premises, the railway domain has been a traditional playground for practical experimentation with formal methods and tools [5], [8], [9], [59].

The European Shift2Rail initiative stimulates the development of safe and reliable technological advances to allow the completion of a single European railway area with the ambitious aim to "double the capacity of the European rail system and increase its reliability and service quality by 50%, all while halving life-cycle costs." Shift2Rail funds several projects, among which ASTRail (SAtellite-based signaling and Automation SysTems on Railways along with Formal Method and Moving Block Validation). The project has the goal to introduce novel technologies within the railway industry, borrowing from the automotive and avionics domain. To this end, ASTRail studies the integration of satellite-based train positioning based on a Global Navigation Satellite System (GNSS), moving block distancing, and automatic train driving, with the support of formal methods. One of the objectives of the project is to identify the most suitable formal methods and tools to be used for railway system development. To this end, ASTRail considers three main steps: i) a systematic literature review on formal methods and tools for railways, complemented with surveys with practitioners, and a review of research projects; ii) a systematic analysis of the most promising formal tools, based on the results of the previous activity; iii) a case study in which a selection of tools is used for the development of formal models of a railway signaling system that includes the aforementioned technologies. Results about the literature review and the surveys are reported in

[2] http://www.shift2rail.org
[3] We use the term "vendor" in a broad sense, including organizations providing commercial and free solutions.

previous works [12], [55], [56]. A comprehensive systematic mapping study has also been carried out recently [53]. The current work presents the systematic analysis of a selection of tools. Before this analysis, we also carried out a judgment study, involving *formal methods experts*, in which 9 different tools were qualitatively analysed and specific strengths and weaknesses were identified [18].

2.2 Formal Methods and Tools

Formal methods are mathematics-based techniques useful for the specification, analysis and verification of systems [7]. These methods are normally oriented to the development of mission-critical software and hardware, and are supported by software tools that can be used to model the system, perform automated verification, and, in some cases, produce source code and tests. The different formal techniques and tools that we consider can be partitioned into four broad families, namely *theorem proving* [60], [61], [62], *model checking* [63], [64], [65], *refinement checking* [66] and *formal model-based development* [23] [4]. Here, we briefly mention some of the main characteristics of these different families and point to some representative tools.

Theorem proving is an automated reasoning or deduction technique that aims to formally verify that some property holds in a system specification by means of semi-automated proofs that require interaction with the user. Roughly speaking, a theorem prover assists the user in demonstrating theorems over a system specification. The theorems can be related to some invariant that must hold in the specification or to the consistency of some specification refinement. Well-known theorem provers are Atelier B [67], Coq [68], Isabelle [69] and PVS [70].

Model checking is a technique to verify that some desired property expressed with a declarative formalization, typically a temporal logic, is satisfied by a specification. It is common to distinguish between linear-time and branching-time model checking, depending on the type of temporal logic supported by the verification engine, and thus the way in which the desired properties can be expressed. The popularity of model checking is mainly due to its full automation. Traditionally, the model-checking problem is solved by generating and traversing the entire state space of the specification, composed with the property to be analyzed. This is the approach of tools such as ProB [22], SPIN [71] and UMC [72].

The full state-space exploration leads to the well-known state-space explosion problem, and several techniques have been developed to overcome this scalability issue, such as for example symbolic model checking [73], supported by tools like NuSMV/nuXmv [74], [75] and PRISM [76], which applies also probabilistic model checking; statistical model checking [77], [78], applied by tools like UPPAAL [79]; and automata minimization techniques, which are supported by tools like CADP [80] and mCRL2 [81], both based on the formalism of process algebras.

Other tools, like SAL [82], [83] and TLA+ [84], take a different approach. SAL supports high-level specifications with a type system similar to that of the PVS theorem prover and "for infinite state systems it provides an "infinite bounded" model checker that uses SMT solving", which "blurs the line between theorem proving and model checking" [83]. TLA+ instead explicitly combines model checking with theorem proving. Theorem proving does not depend on the problem space, and this avoids the state-space explosion problem, although at the cost of reduced automation.

Refinement checking [66] is a verification technique that consists in automatically assessing that a certain specification is a correct refinement of a higher-level specification. This is the main technique used by FDR4 [66], but it is supported also by other tools, such as ProB [22].

Formal model-based development tools, finally, focus on specification through graphical models, which can then be simulated, analyzed and verified by means of various techniques. Although these tools often embed model-checking capabilities to support verification, their main emphasis is on the modeling phase, and for this reason we classify them in a different family. Examples of such tools are Colored Petri Net (CPN) Tools [85], Modelica [86], SCADE [87] and Simulink [88].

2.3 Related Work

The different families of formal tools briefly listed above give an intuition of the wide landscape of choice available.

On the other hand, empirical comparisons between tools, both for railways and other fields, are limited [18], [19], [20]. This, together with other issues largely discussed in the literature, such as the prototypical level of the tools, the skills required, the psychological barriers, tool deficiences, etc. [8], [9], [16], [89] has seriously hampered the adoption of formal tools by industrial practitioners.

To address this issue, several efforts have been performed in the community of formal methods for system and software engineering. Early comparative works are those by Gerhart et al. [90] and Ardis et al. [91]. The former analyses the results of 12 case studies, and reports a lack of adequate tool support at the time of writing. The latter, instead, performs a critical evaluation of 7 formal specification languages, highlighting that maintainability of the models, especially in terms of modularity, is one of the most common issues to address. More recently, Abbassi et al. [92] qualitatively compares three specification languages, namely B, TLA+ and Dash—an extension of the Alloy language, and show specific language features that characterize each language. Overall, these works mainly focus on methods and specification languages, rather than tools.

Well-known initiatives targeting a comprehensive and comparable view of formal tools are: tool competitions (cf., e.g., [93]), mostly oriented to assess performance over shared benchmarks [11]; the ETI initiative [94], aimed at creating an online service to experiment with different formal tools; the comprehensive study by Garavel and Graf [10], a report of over 300 pages characterizing the landscape of available methods and associated tools; the recent expert survey on formal methods by Garavel et al. [8], a report of 67 pages on the past, present and future of formal methods and tools in research, industry and education based on a survey among 130 high-profile experts in formal methods.

In the railway domain, different surveys have been performed about formal methods and tools (cf., e.g., [5],

[4] As we focus on system *design*, we do not consider deductive verification or abstract interpretation, which are applied to source code.

[21], [95], [96], [97], [98]). Bjørner [95] presents a first, non-systematic survey of formal methods applied to railway software. With a focus on the B method, the book edited by Boulanger [98] discusses successful industrial usages, including railway experiences at Siemens and other companies. Aspects of railway system development are covered by the book edited by Flammini [97], where two entire chapters are dedicated to formal methods applications. New applications and future challenges related to the increasing complexity of railway systems are indicated in the reviews of Fantechi et al. [5], [96]. A special issue dedicated to formal methods for transport systems contains two experiences in the railway domain [59]. A systematic literature review [12], including 114 research papers on formal methods and railways, and two surveys with practitioners [55], [56], were also recently published, highlighting the importance given by the railway industry to *maturity* and *usability* of formal tools.

Efforts in comparing formal tools have been performed also in railways. Specifically, Haxthausen et al. [51] compares two formal methodologies for interlocking system development. Mazzanti et al. [50] replicates the same railway design with multiple formal methods, and qualitatively discusses the peculiarities of each tool, namely CADP, FDR4, NuSMV, SPIN, UMC, mCLR2 and CPN Tools. Basile et al. [52] compares two formalisms and their associated simulation-based tools, among which UPPAAL, by applying them to a case study from an industrial railway project. During the ABZ 2018 conference, a case study track was specifically dedicated to a railway problem [99]. The specification provided by the organizers was modeled by different authors through different tools, including SPIN, Atelier B and ProB, leading to independent publications (e.g., [42], [44], [45], [46]) in a dedicated special issue [100]. Finally, Ferrari et al. [18] performs a judgment study considering 9 tools (all those used by Mazzanti et al. [50], except for mCRL2 and CPN Tools), and produces a table with strengths and weaknesses, to guide the adoption of formal methods in railways. The study does not consider specific evaluation features, and takes more of a bird-eye view on the tools.

Contribution. Overall, none of the existing studies and initiatives, both in railways and in the other domains, performs a *systematic* evaluation of formal tools. The current work addresses this gap. Compared to previous contributions in railways [18], [50], [51], [99], our work: i) is systematic and considers a comprehensive set of evaluation features; ii) takes into account a larger set of tools; and iii) is the first that presents a usability study with railway practitioners.

3 Research Design

Our study is oriented to provide a catalogue of the characteristics of available formal tools for railway signaling system design, and achieve a more structured understanding of the field. To this end, we perform a systematic tool evaluation adapting the guidelines of the DESMET methodology by Kitchenham [54], according to the qualitative feature analysis paradigm. Specifically, we first select a set of relevant features, and then we consider existing documentation and perform tool trials to qualitatively evaluate the features with a sufficient degree of objectivity. Usability in terms of ease-of-use is evaluated through a usability study. The produced comparison table and the results of the usability study are used as a basis to compare the different tools and discuss research gaps as well as mismatches between railway designers' expectations as identified by previous work [55], [56], and tools' functionalities and qualities. Although the DESMET methodology normally suggests a numerical scoring scheme for the evaluation features (i.e., qualitative values are mapped to numbers), we did not adopt this approach, as our goal is not to rank the tools, but to provide a comprehensive comparison without the need to establish winners and losers. It is worth noting that our research goal is analytic and informative, and not prescriptive. The adopted methodology does not allow us to provide a set of guidelines for the *deployment* of formal methods in companies, and address relevant issues such as cost-benefit analysis, or process adaptation [9]. Different types of context-dependent research strategies, such as, e.g., action research or case studies [58], should be considered to address this goal.

Our research focuses on the railway signaling domain, including interlocking, train separation and train supervision, given the long history of applications of formal methods in this field, and the demand for a more widespread knowledge dissemination towards practitioners and system designers [18], [19]. In the following, when we use the term "railway", we generally intend "railway signaling". Our work is concerned with the early design phases, which is also the development stage normally considered in applications of formal methods in railways [53]. This is common to other fields, as formal verification techniques are considered to be at their best during early design [8].

3.1 Research Questions

Our overall research goal is to *systematically evaluate formal tools for system design in the railway signaling domain*. The goal is decomposed into the following research questions (RQs):

- RQ1: *Which are the features that should be considered to evaluate a formal tool?* This question aims at creating a taxonomy of features to evaluate the tools. To address this goal, we perform a brainstorming session of feature elicitation, followed by consolidation activities to come to a well-defined set of features and associated values. This activity involves researchers and railway practitioners.
- RQ2: *How do different tools compare with respect to the different features?* With this question, we want to highlight strengths and weaknesses of the tools according to the selected features. The 13 tools are individually evaluated by three assessors with the support of the tools' documentation, research papers, and through brief trials with simple draft models defined by the assessors themselves. The evaluation enables the identification of individual as well as collective weaknesses and strengths of the tools. For each tool, an evaluation sheet is produced.
- RQ3: *How do different tools compare with respect to usability?* Usability is a particularly relevant macro-feature that needs to be assessed by different subjects

to be properly evaluated. To compare the ease-of-use of the tools we perform a preliminary usability assessment with railway practitioners. The practitioners did not directly use the tools, but assisted to structured tools' live demos, and used the System Usability Score (SUS) test to evaluate them. This unorthodox usability assessment compensates for the inherent complexity of learning formal methods, and give a first intuition of the potential *ease-of-use* the tools from the viewpoint of practitioners.

3.2 Tools

The 13 selected tools for the evaluation are nuXmv (1.1.1), ProB (1.9.3), Atelier B (4.5.1), Simulink (R2020a), SPIN (6.4.9), UMC (4.8), UPPAAL (4.1.4), CADP (2020-g), FDR4 (4.2.7), CPN Tools (4.0), mCLR2 (202006.0), SAL (3.3) and TLA+ (2). The first nine were included as they are evaluated among the most mature formal tools for railways, according to Ferrari et al. [18]. Specifically, Ferrari et al. [18] selected the first seven tools considering the results of a literature review and a survey with stakeholders [12], while CADP and FDR4 were included as representative tools for process algebras, since these are explicitly mentioned in the railway norms [15]. The last four were added here to consider a more representative set of the different flavors of formal system modeling and verification available. Specifically, CPN Tools was included to consider Petri Nets, a widely used graphical formalism also for railway modeling [39], [56]. mCLR2 is based on the algebra of communicating processes and supports minimization techniques, as does CADP, but mCRL2 is open source [81]. Finally, SAL and TLA+ are particular tools. SAL is a model checker whose models have a syntax that is partly derived from that of the PVS theorem prover, while TLA+, which is also used at Amazon [101], explicitly integrates theorem proving and model checking. More information about the peculiarities of each tool is reported in the evaluation sheets that were produced for each tool [57]. Our recent systematic mapping study on formal methods applications in railways [53] generally confirms the selected tools among the most frequently used in the literature[5]. It should be noted that "used in the literature" does not imply that a certain tool was used for the development of real-world railway products. Our goal is not to assess tools that have already been extensively used in railway companies, but rather to have a representative set of tools that can be applied for early design of railway signaling solutions. The landscape is vast and fragmented and several industry-relevant tools, such as SCADE, SafeCap, Systerel Smart Solver (S3), Prover, Ovado, etc., are not considered here. An exhaustive analysis is not the objective of this work. Furthermore, this would hardly be feasible, since, according to our mapping study [53], over 200 different tools had some experimental use in railways, and 36 of them appear in at least three papers. Further work can update the list of tools, using the features proposed in this paper as reference for assessment. It should be noted that the selected tools include full-fledged model-based frameworks as well as verification engines, which inherently have different goals and supported features. However, a list of hard-to-compare tools is often what is available to railway practitioners who first approach formal tools—this is confirmed by our surveys with stakeholders [55], [56], in which a mixture of formal tools emerged as most used in railways. Thus, we wanted to make an effort to put these tools on the same scale, so that useful evidence could be derived for these prospective users.

We remark that our evaluation is based on direct hands-on experience with the tools. The tools were tried in their free or academic license, depending on the available options. The commercial license was not purchased for any of the tools.

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 to 10	0

3.3 Study Participants

Table 1 summarizes the study participants' characteristics and expertise.

The main participants are the first three authors of the paper, who were involved in all the phases of the study. They are referred to as *assessors*. The assessors are male academics, with 13 years (1st author), 9 years (2nd author) and 4 years (3rd author) of experience in applications of formal methods to the railway industry, but with complementary expertise: semi-formal methods, classical model checking, and probabilistic and statistical methods and tools, respectively.

The association between tools and assessors was based on their expertise, to have an authoritative analysis. It is worth noting that the 2nd author has over 25 years of experience in formal methods, as well as hands-on experience on formal tool development, while the 1st author has 3 years of experience with applying semi-formal methods in the railway industry. The adopted DESMET methodology is inspired by systematic literature reviews [102], in which the authors are normally involved in the evaluation, and, as in our case, aims to reach objectivity through the selection

[5] The mapping study focuses on peer-reviewed literature from recognized search engines of scientific relevance, as recommended by Kitchenham [102]. It does not consider grey literature, and industry-oriented venues, such as IRSE (Institution of Railway Signal Engineers) publications, the Safety-critical Software Systems Symposium, and the Computers in Railway (COMPRAIL) conference. Different, and possibly complementary, results could be obtained with a multi-vocal literature review.

of appropriate evaluation features, triangulation and cross-checking between the authors.

The additional participants included three researchers (referred to as *academic experts*) and 9 railway practitioners (*industry experts*). One of the researchers is the last author, a male academic that has more than 15 years of experience with multiple formal methods, with a specific focus on the application of model-checking tools, among which probabilistic and statistical variants, recently also to railway problems. The other two academics (one male and one female) each have more than 25 years of experience in the application of formal methods to railways. Finally, the 9 practitioners, of which one female, have in general more than 10 years of experience in railways, but no prior experience in the application of formal methods.

3.4 RQ1: Feature Selection

The features were elicited with a collaborative approach inspired by the KJ method [103]. A 3 hour workshop was organized involving 8 participants: the assessors, the academic experts and two of the industry experts (ID 8 and 9 in Table 1). Participants were given 5 minutes to think about relevant features that should be considered when evaluating a formal tool for railway system design. Each participant wrote the features that they considered relevant in a sheet of paper that was not made visible to the others. Then, the moderator asked the participants to list their features and briefly discuss them. When the explanation given by the feature proponent was not clear, the others could ask additional questions to clarify the meaning of the feature. The moderator reported each feature in a list that was made visible to all participants through a projector. When a feature was already mentioned, it was not added to the list. At the end of the meeting, the assessors homogenized the feature names. For each feature, the assessors jointly defined the possible values, as well as systematic assessment criteria. Features that could not be evaluated with a sufficient degree of objectivity, and that required specific experimental evaluations (i.e., notation readability, resource consumption, learning curve and *scalability* [20]) were excluded from the analysis.

The process of consolidation of the systematic assessment criteria went through multiple updates and refinements also during the feature evaluation itself (cf. Sect. 3.5). This iterative process was oriented to improve objectivity without compromising clarity, so to facilitate replication and preserve the initial spirit of the identified features. The final list of features and possible values was later cross-checked by the other participants. A reference evaluation document was redacted with all the information to be included for each tool.

3.5 RQ2: Feature Evaluation

The three assessors performed the systematic feature evaluation. The assessors worked independently on a subset of the considered tools, and they produced an evaluation sheet for each tool, based on the reference document. The assessors worked on the tools that they were more familiar with, according to their previous experiences, so to give a more informed judgment. Specifically, one assessor (3rd author) used UPPAAL and Atelier B; one used Simulink (1st author); the assessor with more experience in formal methods (2nd author) used the remaining tools. To perform the evaluation, the assessors followed a structured procedure: 1) install and run the tool; 2) consult the website of the tool, to check the official documentation; 3) opportunistically search for additional documentation to identify useful information to fill the evaluation sheet; 4) refer to the structured list of papers on formal methods and railways published in the previously referred literature review [12] [6] (cf. Sect. 2.1), to check for tools' applications in railways; 5) perform some trials with the tools to confirm claims reported in the documentation, and assign the value to those features that required some hands-on activity to be evaluated; 6) report the evaluation on the sheet, together with the links to the consulted documents and papers, and appropriate notes when the motivation of some assignment needed clarification.

In subsequent face-to-face meetings, the assessors challenged each others' choices, and asked to provide motivations and evidence for the assigned values. The evaluation sheets were further revised by all three assessors to align visions and balance judgments. The process was incremental, and carried out across several months. In total, six meetings of 1 to 2 hours were carried out, and the evaluation sheets were used as living documents during these meetings. Finally, a summarizing table was produced to systematically compare the tools. The final evaluation sheets were part of a deliverable of the ASTRail project, and were thus subject to external review. Finally, the first and last author of this paper jointly cross-checked the table reported in this paper with respect to the evaluation sheets. The sheets, together with a detailed description of features and assessment criteria, are shared in our repository [50].

3.6 RQ3: Usability Evaluation

This section outlines the methodology adopted for the usability evaluation of the tools performed with railway experts. First, a set of models of the same sample system was developed by the assessors using 7 different tools, namely Atelier B, nuXmv, ProB, Simulink, SPIN, UMC and UPPAAL. The other tools were not included as, from the feature evaluation, they were considered to require *advanced* mathematical background to be understood (i.e., CADP, FDR4, mCLR2, SAL, or TLA+, cf. Fig. 1) or they were known from the literature to be inadequate to handle industry-size problems (this is the case of CPN Tools [50]). One exception is Atelier B, which, although requiring advanced background, is one of the few tools already used in the development of real-world railway products (cf. "Integration in the CENELEC process", Fig. 1).

Each assessor worked only on a subset of the tools, depending on their skills, as in the feature evaluation phase. The models were used to showcase the different tools, and evaluate their usability from the point of view of the 9 industry experts. A usability assessment in which the experts would directly interact with the tools was not considered reasonable, due to the skills required to master the tools,

[6] The list of 114 papers, and associated categories, can be accessed at https://goo.gl/TqGQx5.

and due to time constraints. Therefore, the three assessors showed the different characteristics of the tools in a 3 hour meeting with the experts, using the developed models as a reference. The experts were already familiar with the sample system considered, i.e., the moving block system, which was also used as reference by other works in the literature [18], [99], [100]. The assessors asked the experts to evaluate the usability of each tool based on their first impression. It should be noticed that, although unorthodox, this approach is meaningful. In practical cases, formal tool users need to be tool experts [20], [50], and railway engineers will have to interact with them. This requires the engineers to be able to understand the models and make sense of the results, but not to be proficient with the formal tools.

The meeting was performed as follows:

1) **Introduction:** an introduction was given to recall the main principles of the considered system. This served to provide all participants a uniform perspective on the system that they were going to see modeled.
2) **Tool Showcase:** each tool was presented by an assessor in a 15 minutes demo, covering the following aspects: 1) *General structure of the tool:* the presenter opens the tool, and provides a description of the graphical user interface (if available); 2) *Elements of the model:* the presenter opens the model, describes its architecture, and navigates it; 3) *Elements of the language:* minimal description of the modeling language constructs, based on the model shown; 4) *Simulation features:* a guided simulation is performed (if supported); 5) *Verification features:* description of the language used for formal verification, and presentation of a formal verification session with counter-example (if supported).
3) **Usability Evaluation:** after the presentation of each tool, a usability questionnaire for the tool (see below) is filled by the experts.

To evaluate the usability of the tool, we use a widely adopted usability questionnaire, the System Usability Scale (SUS) developed by Brooke [104], [105]. We preferred SUS over other questionnaires like QSUC, QUIS and USE (cf. [106] for pointers), since the questions were considered appropriate for our context. Some questions from the original SUS had to be tailored to our evaluation. The final questionnaire was:

1) I think that I would like to use this tool frequently.
2) I found the tool unnecessarily complex.
3) I thought the tool was easy to use.
4) I think that I would need the support of a technical person to be able to use this tool.
5) I found the various functions in this tool were well integrated.
6) I thought there was too much inconsistency in this tool.
7) I would imagine that most people with industrial railway background would learn to use this tool very quickly.
8) I imagine that the tool would be very cumbersome to use.
9) I imagine that I would feel very confident using the tool.
10) I imagine I would need to learn a lot of things before I could get going with this tool.

Answers are given in a 5-points Likert Scale, where 0 = Completely Disagree; 1 = Partially Disagree; 2 = Undecided; 3 = Partially Agree; 4 = Agree. To calculate the SUS score, we follow the guidelines of Brooke [104]. Overall, the SUS Score varies between 0 and 100, with the following interpretations for the scores, based on the work of Bangor et al. [107]: 100 = Best Imaginable; 85 = Excellent; 73 = Good; 52 = OK; 39 = Poor; 25 = Worst Imaginable.

4 Threats to Validity

Construct Validity. The set of considered tools does not represent the complete universe of formal tools. However, the selection rationale was motivated by other works in railways [18], and by the need to have representatives of different families (cf. Sect. 3). Overall, there is a bias towards model-checking tools with respect to theorem proving ones, but this also occurs in the railway literature, where the preference for this technique is rather common [12], as also confirmed by our recent mapping study [53]. It is worth noting that we did not purchase commercial licenses for those products offering them. This should not affect our evaluation, as academic licenses appear to support the same features. At the same time, the licensing cost did not allow us to experiment with relevant commercial tools such as SCADE, Systerel products, or Prover, which is an inherent limit of our evaluation.

The different features, as well as the possible values, were defined by a limited group of persons. However, the group represents both academic and railway industry viewpoints, and several triangulation activities were carried out to ensure clarity, a sufficient degree of completeness and a uniform interpretation. We moreover made an effort to define sufficiently objective features. In fact, we argue that the partially subjective features are a minority (namely, easy to install, quality of documentation and complexity of license management).

The means that was adopted to evaluate usability, i.e., the SUS questionnaire, is widely used and has been proven effective [107]. On the other hand, the presented usability evaluation is based on a demo showcase of the tools and different results may be obtained with a direct evaluation by railway experts. In particular, a more complete usability analysis would have to use methods such as think aloud, heuristic evaluation or cognitive walk-through [108], which can provide more fine-grained information on the usability issues of a tool. Given the relatively high number of tools and the resources required for such a direct evaluation by railway experts, we opted for a pragmatic approach that provides an indication of the *potential* usability of the tools. This type of evaluation is therefore inherently biased towards novice usage, and does not account for expert usage. Another limitation is the size of the showcased models, which are not detailed real-world designs, and cannot exercise all the features of a certain tool, especially in terms of language expressiveness. This choice is guided by the need to reach a trade-off between the number tools that could be showcased, and the time required by the experts for the assessment—also considering possible fatigue effects.

Internal Validity. The evaluation of the tools may have suffered from subjectivity. To limit this problem, triangulation activities were performed to align the evaluation, and the assessors were required to report appropriate evidence in the evaluation sheet, when some judgment required explicit justification. For some features, the value also depends on the available evidence that comes from the literature and from the websites of the tool, and may not reflect reality. To mitigate these aspects, different sources of information, and practical tool trials have been considered. Full objectivity of our results cannot be claimed, also due to the inherent limitations of the qualitative analysis carried out. However, we argue that the reported process, and the iterations performed to come to a consolidated judgment represent an acceptable approximation of a valid evaluation.

In the usability study, the researchers could have biased the audience towards a certain tool, based on their preferences. To mitigate this threat, the researchers rehearsed the tool showcase before the evaluation with the participation of one of the academic experts, and provided mutual recommendations. Therefore, we argue that the tools were presented in a quite uniform manner.

Another issue is related to the number of subjects involved, i.e. 9, which may be regarded as a limited sample. However, for different methods it has been shown that a group of 10±2 evaluators can identify 80% of the usability problems and that "optimal sample sizes of '10±2 can be applied to a general or basic evaluation situation, for example, just basic training provided to evaluators and a limited evaluation time allowed" [108][7]. Therefore, we argue that, although limited, the sample size is in line with the samples typically used in usability testing.

External Validity. The usability evaluation, as well as the preceding activities, involved railway experts with multiple roles, i.e., system engineers, developers and managers, and with different degrees of experience in railways, although more than 10 years in the vast majority of the cases. This covers a large spectrum of perspectives. On the other hand, the participants were all from the same railway supply company, and this may limit the results. However, it is worth noticing that railway companies follow standardized processes and, in addition, railway systems are well-established and even standardized in some cases (e.g., ERTMS [99], [109]). Moreover, the railway industry is an oligopoly and according to experts from the Shift2Rail project NEAR2050 (Future challenges of the railway sector)[8] "the rail supply industry will continue to be an oligopoly, but core rail system knowledge will have moved to IT departments or IT companies" [110]. Therefore, we argue that the perspectives of companies may be similar to each other and backgrounds and practices are comparable between companies, thus suggesting a sufficient degree of external validity of our results.

5 Execution and Results

5.1 RQ1: Feature Selection

The feature selection process led to the identification of 33 features, which have been hierarchically grouped into *functional*, *language expressiveness*, and *quality* features, with 8 categories in total. Below, we report categories, features and possible values with evaluation criteria. Features that could not be evaluated with a sufficient degree of objectivity (e.g., notation readability, resource consumption, learning curve and *scalability* [20]) were excluded from the analysis. It should be noted that, though we did not quantitatively assess scalability, we give a qualitative indication of the techniques supported by the tools to verify large-scale models (cf. Sect. 5.1, "Large-scale Verification Technique"). As acknowledged by previous literature [11], [20], [50], the wide range of verification and optimization options made available by the tools to address the scalability issue makes a systematic comparison from the quantitative standpoint an inherently hard task. The choice of the more appropriate options depend on the problem at hand, and providing general conclusions on a single design would not be in line with the scope of this work. We refer to our previous research that studies the scalability of model checkers for railway interlocking [111] and automatic train supervision systems [50].

5.1.0.1 Functional Features: Functional aspects have been organized in two categories: development functionalities and verification functionalities.

Development Functionalities.

- *Specification, modeling:* specifies whether the model can be edited graphically (GRAPH) or in some textual representation (TEXT) inside the tool, or whether the model can only be imported as a textual file (TEXTIM).
- *Code generation:* indicates if the tool supports automated code generation from model specifications (YES, NO).
- *Documentation and report generation:* the tool supports the automated generation of readable reports and documents (YES), allows the user to produce diagrams or partial reports that can in principle be included in official documentation (PARTIAL) or does not generate any usable documentation (NO).
- *Requirements traceability:* indicates if it is possible to trace requirements to the artifacts produced with the tool (YES, NO).
- *Project management:* specifies if the tool supports the management of a project, and the GUI-based navigation of its conceptual components such as models, verification results, tests, etc. (YES, NO).

Verification Functionalities.

- *Simulation:* indicates whether a model is executable, such that simulation of the evolution of the system models is possible. Such simulation could be either graphical (GRAPH) or textual (TEXT), a mix of the two (MIX) or absent (NO).
- *Formal verification:* the type of formal verification supported can be linear-time model checking (MC-L),

[7] The study by Hwang and Salvendy [108] considers three usability evaluation methods, namely think aloud, heuristic evaluation and cognitive walk-through. Here, due to pragmatic decisions discussed earlier in this section, we use a more lightweight approach based on a tool showcase, which arguably falls in the "basic evaluation situation" identified in the reported quote from [108].

[8] http://www.near2050.eu

branching-time model checking (MC-B), observer-based model checking (MC-O), theorem proving (TP) or refinement checking (RF) (cf. Sect. 2.2).

- *Large-scale Verification Technique:* indicates the type of approach adopted to verify large-scale models. This can be on-the-fly model checking (FLY): the state is generated on demand; Partial order reduction (POR): exploitation of symmetries in the state space; Parallel computation (PAR): parallel computation distributed on more hosts; Bounded Model Checking (BMC): state-space exploration up to a certain depth; Symbolic Model Checking (SYM): compact state space representation; SAT/SMT constraint solving and theorem proving (SCT): avoid explicit reasoning on the state space; Statistical Model Checking (SMC): avoid full state-space generation using simulations and provide an approximate solution; Compositionality and minimization (COM): divide the problem into smaller subproblems; No technique (NO).
- *Model-based testing:* indicates whether the tool provides support for automatically derived testing scenarios (YES, NO).

5.1.0.2 Language expressiveness: This group of features collects technical aspects related to the main modeling language made available by the tool.

- *Non-determinism:* evaluates if non-determinism is expressible. In particular, whether the language allows internal non-deterministic system evolution (INT) or external choices associated to inputs or trigger-events allow to express non-determinism (EXT).
- *Concurrency:* evaluates if and how concurrency aspects can be modeled. The model can be constituted by a set of asynchronously interacting elements (ASYNCH), by synchronous elements (SYNCH), by both (A/SYNCH) or by just one element (NO).
- *Timing aspects:* considers if the modeling language supports the notion of time (YES, NO).
- *Stochastic or probabilistic aspects:* evaluates if it is possible to model aspects related to randomness, such as for example stochastic delays or probabilities (YES, NO).
- *Modularity of the language:* evaluates how the architecture of the model can be structured in the form of different hierarchically linked modules. The cases are: the tool allows the user to model in a hierarchical way, and partitioning the model into modules (HIGH); the tool allows partitioning the model into modules, but without a notion of hierarchy (MEDIUM); modules are supported, but without a means to interact, neither by messages nor by shared memory (LOW); modules are not supported (NO).
- *Supported data structures:* the language supports numeric types, but no composite expressions (BASIC) or it has complex expressions like sequences, sets and array values (COMPLEX).
- *Float support:* indicates if floating-point numbers are supported as basic data types (YES, NO).

5.1.0.3 Quality Features: Quality aspects are organized into six categories: tool flexibility, maturity, usability, company constraints and domain-specific criteria.

Tool Flexibility.

- *Backward compatibility:* indicates to which extent models developed with previous versions of the tool can be used in the current version. Cases are: the vendor guarantees that legacy versions of the models can be used in the current version of the tool or the future availability of legacy versions of the tool (YES); the tool is open source, the input language is stable and *de facto* standard or there is evidence of interest in preserving backward compatibility (LIKELY); the tool is not open source and the provider does not show evidence regarding backward compatibility, even if the language is rather stable and a *de facto* standard (MODERATE); source code is not available, input format is not stable, and no information is available from the vendor (UNCERTAIN).
- *Standard input format:* evaluates the following cases: the input language is based on a language standardized by an international organization (STANDARD); the input language is open, public and documented (OPEN); the structure of the model specifications is easily accessible, but not publicly documented (PARTIAL); the internal structure of the model specification is hidden (NO).
- *Import from or export to other tools:* evaluates the following cases: the tool provides several import/export functionalities (HIGH); the tool has a standard format used by other tools or exports to some other formats (MEDIUM); the tool does not have import/export functionalities (LOW).
- *Modularity of the tool:* evaluates if the tool includes different modules and packages. Values are: the tool is composed of many modules that can be loaded to address different phases of the development process (HIGH); the tool offers multiple functionalities, but not in the form of modules that can be loaded and combined (MEDIUM); the tool offers a limited number of functionalities in a monolithic environment (LOW).
- *Team support:* specifies support for collaborative model development (YES, NO).

Maturity.

- *Industrial diffusion:* this is determined by whether the website of the tool reports multiple (HIGH), a few (MEDIUM) or no (LOW) cases of industrial usage;
- *Stage of development:* the cases are: the tool is a stable product with a long history of versions (MATURE); the tool is recent, but with a solid infrastructure (PARTIAL); the tool is a prototype (PROTOTYPE).

Usability—in the broad sense of ISO 9241-11:2018 [112].

- *Availability of customer support:* considers if reliable customer support can be purchased for maintenance and training (YES), free support is available in the form of bug reports, mailing lists or online user groups or discussion forums (PARTIAL) or communication channels need to be established between producers and users to have support (NO).

- *Graphical User Interface:* the cases are: the tool has a well-designed and powerful GUI (YES); a user-friendly GUI exists, but it does not cover all the tool functionalities in a graphical form (PARTIAL); a GUI exists, but not particularly powerful (LIMITED); the tool is command line (NO).
- *Mathematical background:* this feature aims at giving an idea of how easy it is to learn to use the tool for an electronic or computer engineer (i.e., the typical railway practitioner). Cases are: the tool does not require particular logical/mathematical skills (BASIC); the tool requires some knowledge of temporal logic (MEDIUM); the tool requires knowledge of theorem proving or process algebras (ADVANCED).
- *Quality of documentation:* cases are: the documentation is extensive, updated and clear, it includes examples that can be used by domain experts, and it is accessible and navigable in an easy way (EXCELLENT); the documentation is complete but offline, and requires some effort to be navigated (GOOD); the documentation is not sufficient or not easily accessible (LIMITED).

Company Constraints.

- *Cost:* cases are: the tool is available under payment only (PAY); the tool is free under certain conditions (e.g., academic license) and with moderate cost for industrial use (MIX); the tool is free (FREE).
- *Supported platforms:* indicates possible platforms supported by the tool (e.g., Windows, MacOS, Linux, or ALL three).
- *Complexity of license management:* cases are: the tool is free for commercial use, and no license management system is required (EASY); the tool offers academic and commercial licenses, both upon payment, limited effort was required to handle the academic license, and adequate information is provided for the commercial one (ADEQUATE); the tool has both a free and a purchasable version, limited problems were encountered when trying the free version, but limited information is provided on the website concerning the licensing system of the commercial one (MODERATE); several problems were encountered with the license management system (COMPLEX).
- *Easy to install:* cases are: the tool requires little or no external components (YES); the tool installation depends on external components or the installation process is not smooth (PARTIAL); the installation can interfere with the customer development environment (NO).

Domain-specific Criteria.

- *CENELEC certification:* cases are: the tool is certified according to the CENELEC norm (YES); the tool includes a CENELEC certification kit or it is certified according to other safety-related norms like DO178C [113] (PARTIAL); none of the above (NO).
- *Integration in the CENELEC process:* estimates how easy it is to integrate the tool in the existing railway life-cycle as described by the CENELEC norms. Cases are: in the literature review from [12] or in the tool documentation, evidence was found of tool usage for the development of railway products according to the CENELEC norms (YES); evidence was found of the usage of the tool in railways, but no CENELEC products developed with the tool (MEDIUM); no usage in railways was found (LOW).

5.2 RQ2: Feature Evaluation

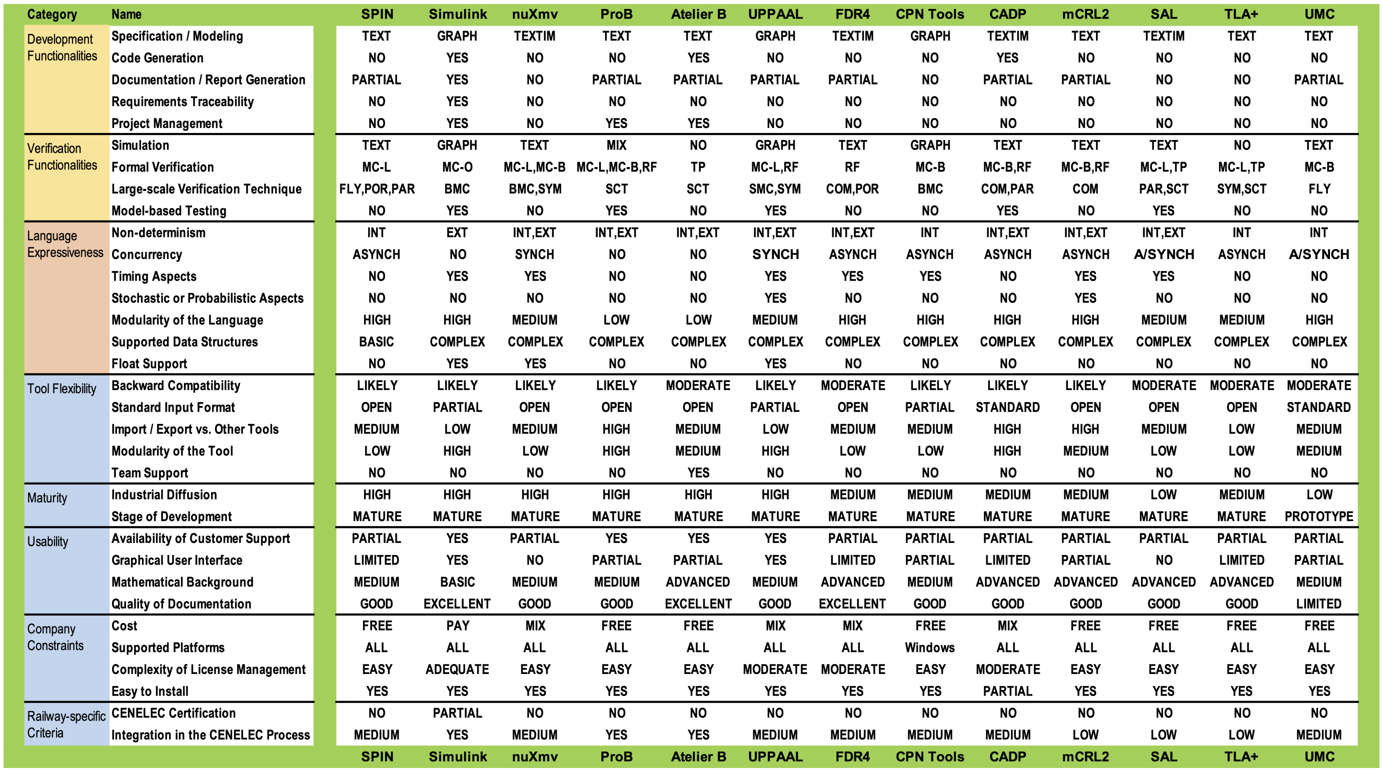
Figure 1 reports the table resulting from the feature evaluation activity. The reader is invited to consult the sheet of each tool to have clarifications on the judgments provided. Here we summarise the most evident trends, and contrast them with existing literature on desired features of formal tools in railway as indicated by recent surveys [12], [55], [56], and limitations and barriers for formal methods adoption (cf. [11], [16], [17], [19], [20], [114], [115], [116]).

Development Functionalities. Concerning development functionalities, we observe that the majority of the tools are based on textual specifications of the models (TEXT and TEXTIM) and, with the exception of Simulink, most of them support only a limited subset of the other complementary functionalities, such as code generation and requirements traceability. Interestingly, the project management feature, common in any IDE for software development, is only available in Simulink, ProB and Atelier B.

Observations. The tools appear to give limited relevance to development functionalities. This, to our knowledge, was not observed by other authors discussing limitations of formal tools (cf. [11], [16], [17], [20]). Traceability in particular is regarded as one of the most relevant functionalities by railway stakeholders [12], who need to ensure that all the artifacts of the process are explicitly linked. The scarce support for traceability is therefore a relevant pain point. Concerning the type of specification language supported (graphical vs. textual), some railway stakeholders may in principle prefer graphical languages [23], while few tools support them. However, Ottensooser et al. [117] have empirically shown that a pictorial specification is not necessarily more understandable than a textual one, and code-like models may be easier to maintain. Therefore, the need for graphical languages—and the understanding of specifications by all stakeholders—could be conflicting with the need for flexibility. Regardless of the language format, a project management feature is needed to manage complex industrial models.

Verification Functionalities. Verification functionalities, including model-based testing and simulation, are supported by a larger number of tools. However, specific strategies in terms of formal verification and scalability approaches are adopted by each platform, and this suggests a rather wide difference in terms of types of properties that can be verified on the models by each tool.

Observations. The difference between approaches is justified by the academic origin of most of the tools, which were primarily used to implement novel verification techniques, as observed by Garavel and Mateescu [11]. This has an impact on the degree of experience required to master a tool. Several authors (cf. [11], [20], [50]) highlighted that the differences in terms of verification strategies, together with the wide range of optimization options made available



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. 1. Evaluation table

to address scalability issues, require users to be experts in the tool to successfully verify large designs. The presence of several tools supporting model-based testing suggest that tool developers appear to be aware of the need for *complementarity* between testing and formal verification, which are not interchangeable activities.

Language Expressiveness. In terms of language expressiveness, the variety of feature value combinations, and therefore the specificity of each modeling language, is also quite evident, as for verification functionalities. In other terms, each tool is somewhat *unique*, both in terms of specification and in terms of verification strategies. Probabilistic aspects and floats have in general limited support. Only UPPAAL and mCLR2 allow the expression of probability, and floats are native types only for Simulink, nuXmv and UPPAAL.

Observations. The uniqueness of each language shows that after twenty years, the issues of *isolation* of formal specification languages pointed out by Van Lamsweerde [116] have yet to be solved. The limited support for floats suggests that, in most of the cases, tools are oriented to designing models that do not go down to the expressiveness of source code in terms of numerical data representation. Abstracting from details is nevertheless one of the principles of systems modeling, and other tools oriented to static analysis should be used to deal with errors arising from floating point numerical types [118].

We note that probabilistic features are traditionally associated to specialized tools for the analysis of performances and other dependability aspects [13], [14]. In this paper, we consider tools oriented towards system design, and this is the reason why this feature has limited support.

Tool Flexibility. Tool flexibility sees a major weakness in the team support feature, with Atelier B as the only tool including it. Only Simulink, ProB, UPPAAL and CADP can be regarded as flexible toolboxes (cf. “Modularity of the Tool” = HIGH), and only ProB, CADP and mCLR2 are open to different formats with several import/export functionalities. This suggests that in many cases the tools are independent ecosystems, and their integrated usage may be complicated.

Observations. This issue also observed by other authors [11], [20], who pointed out the high degree of specialization of languages and tools. Combining at least a subset of the tools and integrating them in a coherent process is extremely relevant. Indeed, as observed by previous work [18], [119], and given the “uniqueness” of each tool, the needs of an industrial process cannot be fulfilled by one platform only.

Maturity. The majority of the tools has a mature stage of development, and diffusion appears to be medium-high.

Observations. The high degree of industry-readiness observed is in contrast with the observation of previous work (cf., e.g., [9], [115]) reporting that the low maturity and prototypical level of many tools are among the main obstacles for formal methods adoption. As maturity is, by far, recognized as the most relevant quality attribute that a formal tool should have to be applied in the conservative world of railways [55], [56], we argue that low maturity can be considered as a false barrier for formal methods adoption. If we consider the Technology Readiness Level (TRL) scale [120], the tools that fulfill both maturity features (“Industrial Diffusion” = HIGH and “Stage of Development” = MATURE) can in principle be considered to have a TRL equal to 9, i.e., *tool proven in operational environment*. However, to assess TRL in a sound manner, Technology Readiness Assessment should be conducted [120], which is not within our scope. It should also be noted that tool maturity does not imply that the tool has actually been used in the production of railway software. Railway-specific criteria—discussed later in this section—provide more appropriate indications in this sense.

Usability. Usability aspects appear, at first glance, as major pain points, with limited customer support, limited GUI and the need for a medium to advanced mathematical background.

Observations. Ease of learning, which is the second-most desired feature [55], [56], is notoriously a problem for formal tools [16], [121], and it is complicated by the decreasing mathematical competence of engineers over the years [121]. In this sense our work confirms the literature. On the other hand, it also highlights that, to address this problem, not only engineering curricula should be enriched, and better GUIs should be developed, but also more effort should be dedicated to customer support, which is acceptable only for Simulink, ProB, Atelier B and UPPAAL. Some tools offer community-based support (e.g., the public discussion forum of SPIN[9], the user groups for TLA+ in Google Groups[10], and Reddit, [11]), and mailing lists, as in the case of nuXmv, FDR4 and SAL—more information is made available in our supplementary material as part of the individual evaluation sheets of each tool [57]. While these solutions are effective in an academic or open source environment, we argue that they may not be sufficient in a traditional development context, as the railway one. Indeed, previous works [2], [50] have observed that railway companies prefer to interact with other companies for customer support, to guarantee maintenance of legacy tool versions, and quick resolution of possible issues.

Company Constraints. Company constraints are in general fulfilled, with several platforms supported and ease of install and license management. This confirms the maturity and industry-readiness of the majority of the tools—cf. "Maturity" for related observations.

Railway-specific Criteria. Railway-specific criteria are not fulfilled, with MEDIUM/LOW easiness in integrating a tool in the CENELEC process, and no certified tool. However, three tools, namely Atelier B, ProB and Simulink have been used to develop railway products, indicating that their integration in the CENELEC process is at least possible.

Observations. The issue of CENELEC integration is regarded as the third most desired quality feature by railway practitioners [55], [56]. It appears that a major pain point that industry may face is that practitioners do not have guidance to accommodate these tools within their industrial processes, and how to combine the tools together in a flexible way. This was also observed in other domains, such as aerospace [16]. Concerning CENELEC certification, an analysis of commercial tools can provide a different viewpoint. The SCADE[12] platform includes a CENELEC certified code generator. The tool Prover[13], which is specialized for interlocking development, includes Prover Certifier, a package that produces safety evidence for the CENELEC norms. Another tool, OVADO[14], based on ProB, distributed by Systerel and specialized in the verification of configuration data, is also CENELEC certified. This suggests that, while certification requirements are not fulfilled by the general-purpose platforms for early design considered in our evaluation, the certification issue is instead addressed by problem-specific (Prover and OVADO), or task-specific (code generation in SCADE) commercial tools.

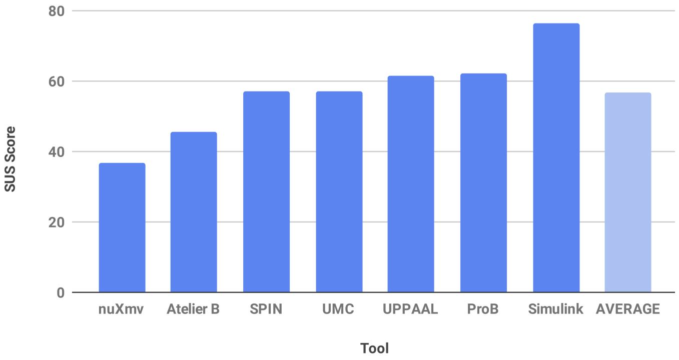


Fig. 2. SUS scores for the different tools

5.3 RQ3: Usability Evaluation

Figure 2 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). This is a formal model-based development tool, with appealing, effective GUIs, powerful languages and simulation capabilities. Simulink is followed by three other tools with acceptable GUIs, but with widely different capabilities, namely ProB (SUS Score = 62.22), UPPAAL (61.67) and UMC (57.22). ProB and UMC allow the user to model in textual form, but present the results of the simulation also in graphical form. Instead, the UPPAAL language is entirely graphical, and presents a graphical simulation style that recalls message sequence charts, which are well known by railway practitioners. Finally, SPIN (SUS Score = 56.94), Atelier B (45.56) and nuXmv (36.67), with some differences, are considered among the least usable tools. Although SPIN is a command-line tool, with only a limited GUI, its scores are higher than Atelier B. This can be explained considering the following observations discussed with the participants: i) SPIN uses a modeling language that is very similar to the C language, and therefore was considered familiar by the participants, who, in turn, gave higher scores; ii) Atelier B uses a refinement-based theorem-proving approach, which requires advanced skills to be mastered.

When computing the *average* SUS Score, we obtain 56.67, which is between OK and Good [107]. Hence, the general usability of the tools can be considered acceptable.

[9] http://spinroot.com/fluxbb/
[10] https://groups.google.com/g/tlaplus
[11] https://www.reddit.com/r/tlaplus/
[12] https://www.ansys.com/it-it/products/embedded-software/ansys-scade-suite
[13] https://www.prover.com
[14] https://ovado.net

6 Summary and Implications

Considering the inherent limitations of the DESMET methodology [54], our **take-away messages** are as follows:

1) Many of the formal tools lack support for development features and process-integration aspects;
2) Most of the formal tools are independent ecosystems, with unique, non-standard languages and specialized verification capabilities;
3) Formal tools are mature, as highly desired by railway industry [55], [56];
4) Most usability aspects appear to be low in principle, but, when the formal tools are assessed by practitioners, usability is considered acceptable.

In the following, we discuss the main implications of our results. Our goal is to foster the debate about the adoption of formal tools in safety-critical software engineering, and therefore we also indulge in expressing personal opinions.

Implications for Tool Developers. The majority of the considered formal tools lack support for process-related aspects, including development functionalities such as traceability and document/report generation, and no evidence of their integration in the railway process. With some differences, exceptions are Simulink, Atelier B and ProB. Tool developers are encouraged to give more relevance to process-related aspects and enrich their tools with additional development functionalities. Furthermore, they are encouraged to invest in customer support and consultancy: companies need to be accompanied in the introduction of novel tools, and tailored recommendations need to be provided on how to best adapt their processes. In this sense, the maturity of a tool is not enough if the tool vendor is not sufficiently mature from the commercial standpoint. Making their tool more interoperable/standardized is also suggested, as importing from known formats could in principle facilitate the transition of users to their tool. The vendors of those tools that are not considered in our evaluation can also benefit from this research. In particular, the grid of features can be useful to better position their products, and compare them with competitors.

Implications for Researchers. Tool certification, or qualification according to the norms, is regarded as a relevant problem by industrial practitioners, as also observed by Garavel [11] and Mazzanti [50]. However, none of the tools considered here is CENELEC certified. Some commercial tools, instead, are certified for specific tasks (SCADE), or specific problems (Prover and Ovado). This, in principle, appears to rule out all those tools stemming from academic research. Nevertheless, if a formal tool does not go down to code generation, we argue that tool qualification is not a radical issue. Railway systems get certified also without using formal design tools, as these are highly recommended but not *required* [17]. We argue that evidence should be provided in terms of *cost-benefit* analyses on the introduction of formal methods. Researchers should focus more on empirically showing that introducing a formal tool actually allows companies to reduce the cost of the testing phase, either by detecting errors beforehand or by facilitating the production of test cases.

Our work shows that the usability of the tools is in general *acceptable*. Our usability evaluation considers solely the first impact with a formal tool from the perspective of novices, while our feature evaluation checks general usability criteria (GUI, quality of documentation, customer support, *etc.*). Researchers are invited to confirm our findings in more complex contexts, in which the *learning curve* required to master a certain tool is also taken into account, because a truly usable tool is one that enables novices to quickly build expertise, but also provides appropriate features for more advanced users. At this regard, we encourage stratified usability studies on formal tools, in which usability is assessed from the viewpoint of novices and compared with the evaluation of experts in formal methods. We also observe that there are conflicting reports regarding the ease of learning of tools that require advanced mathematical background. For example, Newcombe et al. [122] reported that Amazon engineers with no prior knowledge of formal tools required only few weeks of training to utilize TLA+, and similar reports are available for the B method [123]. Therefore, circumstantial evidence seems to contradict common beliefs about the inherent complexity of learning formal methods, and we encourage more empirically grounded evaluations to answer the question: “Are formal methods truly difficult to learn?”.

Implications for Railway Practitioners. Practitioners are mainly intended here as R&D engineers working in railway companies or in infrastructure manager organizations. Several suggestions have been given to address the skill barriers of practitioners, and these are mostly oriented to improve education of engineers [16], [121]. However, we argue that this may be a workable problem. To be effectively used, most formal tools currently require experts in formal methods and in the chosen tool [20], [50]. Engineers need to understand the principles of a tool to interact with the experts, but do not necessarily need to be experts themselves. Our usability study suggests that current tools may be sufficiently acceptable by engineers, when they see the tools used by another subject. Considering that also the maturity prejudice appears to be not well founded, we invite practitioners to overcome their notorious skepticism [9], [89], and give formal tools a chance. Not in the least because Miller [124] provides *economical evidence* of the benefits of the application of formal methods and tools (including NuSMV / nuXmv, PVS, SAL, SCADE and Simulink) to industrial problems in avionics.

Tool maturity is the primary concern for railway practitioners [55], [56]. It should be noted that our analysis tackles the maturity of the tools in terms of two features, namely “Industrial Diffusion” and “Stage of Development”. Furthermore, railway-specific criteria (“CENELEC Certification” and “Integration in the CENELEC Process”) give an indication of suitability for railways. By checking these criteria, it is possible to identify the inherent maturity of the tools, as well as their degree of maturity for the railway field. In particular, “Industrial Diffusion” specifies whether the tool is used in industry—not limited to railways, but considering also avionics, automotive, etc. Instead, “Integration in the CENELEC Process" specifically considers whether there is documented evidence of a product developed with the tool according to the CENELEC norms. Given this information, a railway company can take the risk of selecting a mature tool possibly offering useful features, e.g., "Stochastic and Probabilistic Aspects", but being aware that it is not yet used in real-world railway projects. Conversely, the company can decide to play safe and adopt tools that are already proven in use by other railway companies.

Our selection of tools and our analysis can be particularly useful also to railway infrastructure managers. The importance of high quality standard interfaces is becoming more and more recognized by the organizations they work for, and it is widely acknowledged that formal methods are likely to play an important role in achieving the desired high quality of these interfaces (cf., e.g., the Eulynx initiative[15] [125]). Since the tools compared in our paper target high-level rigorous design, they are also relevant for the current attempts to enrich the standard interface definition process with formal analysis [49], [125], [126]. Therefore, though our work does not specifically focus on standard interfaces, we argue that our input can be beneficial also for this trending field of railway research.

Certain categories of practitioners, and in particular signaling engineers who need to verify specific installations, should also consider that, for some railway applications—mostly for wayside systems—domain-specific notations can be used (cf., e.g., Iliasov *et al.* [37]) and automatically verified, thereby facilitating the usage of formal tools by engineers. These practitioners should consider also other problem-specific tools not considered in our analysis, such as Prover, which is oriented to design and verification of interlocking products, and Ovado, targeting the verification of configuration data. Nevertheless, for many development cases, and especially high-level system design, the only (sub-optimal) option is to consider available formal languages and tools, and an interaction with experts in formal methods is therefore unavoidable. In this context, practitioners should keep in mind the limitations of formal tools; in particular, as confirmed by Clarke *et al.* [63] and Baier and Katoen [64], of model checking: i) it verifies a *system model*, and not the actual system (product or prototype) itself; any obtained result is thus as good as the system model; ii) it checks only *stated requirements*, i.e., there is no guarantee of completeness; iii) its usage requires some *expertise* in finding appropriate abstractions to obtain smaller system models and to state properties in the logical formalism used.

As a final remark for practitioners, and especially signaling engineers and managers, this paper does not tackle the problem of tool deployment in a railway development context. The informed choice of a certain tool, according to the information provided in this paper, is only a preliminary step. Practitioners should be warned that adopting a novel formal tool requires appropriate planning of its introduction, and will require adaptation of the existing process, as shown also by previous industrial cases [23], [127].

7 Conclusion

This paper presents a systematic evaluation of 13 formal tools for railway system design as well as a preliminary usability analysis of 7 of these formal tools. We show that the majority of the considered tools are mature and consolidated products, with a reasonably sufficient degree of industry readiness. Although many tools have limited GUIs, and require strong mathematical background, our usability study shows that, on average, the degree of usability of these 7 tools is between OK and Good, thus suggesting that usability may not be a strong barrier for formal tools' adoption. Main barriers are the limited support for development functionalities, such as traceability, and other process-integration features. We share our evaluation sheets [57], which include synthetic yet industry-relevant information about each single tool.

Our work provides a contribution that follows the recommendations of Huisman et al. [128], who ask different stakeholders—next to tool developers, researchers and practitioners, also policy makers and education staff—to join in an effort to reduce the gap between formal methods research and practice. Our study, and the shared data, can be useful to practitioners interested in introducing formal methods in their company, and to the research community of software engineers dealing with system development. Through our contribution, these two different profiles can have a clear and up-to-date opinion about the landscape of available tools, and their salient characteristics. Finally, tool developers can profit from our independent evaluation to identify the most suitable improvements for their tools. It is worth noting that our goal is not to provide a ranking between tools. Indeed, different industrial contexts—in terms of available expertise, project characteristics, *etc.*—may give more prominence to different features. Thus, one tool may be deemed more appropriate than another depending on the relative relevance of the features in the given context.

This work is part of a larger endeavour carried out within the ASTRail project, oriented to provide an overview of the landscape of formal tools in railways, and including surveys with stakeholders [55], [56], and a systematic mapping study [53]. This analytical activity is followed by an exploratory case study currently carried out within the 4SECURail project[16], which aims to provide a cost-benefit analysis of the adoption of formal methods, especially considering the point of view of infrastructure managers. Future work of our team will therefore focus on more context-dependent experiences, with the aim of providing empirical evidence, and possibly guidance, on formal tool adoption.

Acknowledgments

The authors would like to thank the ASTRail and 4SECURail projects, which received funding from the Shift2Rail Joint Undertaking under the European Union's Horizon 2020 research and innovation programme under grant agreements 777561 (ASTRail) and 881775 (4SECURail) in the context of the open call S2R-OC-IP2-01-2019, part of the "Annual Work Plan and Budget 2019", of the programme H2020-S2RJU-2019. The content of this paper reflects only the authors'

[15] https://www.eulynx.eu

[16] https://www.4securail.eu

view and the Shift2Rail Joint Undertaking is not responsible for any use that may be made of the included information.

References

[1] A. Chiappini, A. Cimatti, L. Macchi, O. Rebollo, M. Roveri, A. Susi, S. Tonetta, and B. Vittorini, "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. ACM, 2010, pp. 109–118.

[2] A. Ferrari, D. Grasso, G. Magnani, A. Fantechi, and M. Tempestini, "The Metrô Rio case study," *Sci. Comput. Program.*, vol. 78, no. 7, pp. 828–842, 2013.

[3] A. E. Haxthausen, J. Peleska, and S. Kinder, "A formal approach for the construction and verification of railway control systems," *Form. Asp. Comput.*, vol. 23, no. 2, pp. 191–219, 2011.

[4] K. Winter and N. J. Robinson, "Modelling large railway interlockings and model checking small ones," in *Proceedings of the 26th Australasian Computer Science Conference (ACSC'03)*, ser. CRPIT, M. J. Oudshoorn, Ed., vol. 16. Australian Computer Society, 2003, pp. 309–316.

[5] A. Fantechi, "Twenty-five years of formal methods and railways: What next?" in *Revised Selected Papers of the SEFM 2013 Collocated Workshops: BEAT2, WS-FMDS, FM-RAIL-Bok, MoKMaSD, and OpenCert*, ser. LNCS, S. Counsell and M. Núñez, Eds., vol. 8368. Springer, 2013, pp. 167–183.

[6] A. Fantechi, A. Ferrari, and S. Gnesi, "Formal methods and safety certification: Challenges in the railways domain," in *Proceedings of the 7th International Symposium on Leveraging Applications of Formal Methods, Verification and Validation: Discussion, Dissemination, Applications (ISoLA'16)*, ser. LNCS, T. Margaria and B. Steffen, Eds., vol. 9953. Springer, 2016, pp. 261–265.

[7] J. M. Wing, "A specifier's introduction to formal methods," *IEEE Comp.*, vol. 23, no. 9, pp. 8–22, 1990.

[8] H. Garavel, M. H. ter Beek, and J. van de Pol, "The 2020 Expert Survey on Formal Methods," in *Proceedings of the 25th International Conference on Formal Methods for Industrial Critical Systems (FMICS'20)*, ser. LNCS, M. H. ter Beek and D. Ničković, Eds., vol. 12327. Springer, 2020, pp. 3–69.

[9] J. Woodcock, P. G. Larsen, J. Bicarregui, and J. Fitzgerald, "Formal methods: Practice and experience," *ACM Comput. Surv.*, vol. 41, no. 4, pp. 19:1–19:36, 2009.

[10] H. Garavel and S. Graf, "Formal methods for safe and secure computer systems," Bundesamt für Sicherheit in der Informationstechnik, Bonn, Germany, BSI Study 875, December 2013.

[11] H. Garavel and R. Mateescu, "Reflections on Bernhard Steffens physics of software tools," in *Models, Mindsets, Meta: The What, the How, and the Why Not?*, ser. LNCS, T. Margaria, S. Graf, and K. G. Larsen, Eds. Springer, 2019, vol. 11200, pp. 186–207.

[12] A. Ferrari, M. H. ter Beek, F. Mazzanti, D. Basile, A. Fantechi, S. Gnesi, A. Piattino, and D. Trentini, "Survey on formal methods and tools in railways: The ASTRail approach," in *Proceedings of the 3rd International Conference on Reliability, Safety, and Security of Railway Systems: Modelling, Analysis, Verification, and Certification (RSSRail'19)*, ser. LNCS, S. Collart-Dutilleul, T. Lecomte, and A. Romanovsky, Eds., vol. 11495. Springer, 2019, pp. 226–241.

[13] A. Hartmanns and H. Hermanns, "In the quantitative automata zoo," *Sci. Comput. Program.*, vol. 112, pp. 3–23, 2015.

[14] J.-P. Katoen, "The probabilistic model checking landscape," in *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science (LICS'16)*. ACM, 2016, pp. 31–45.

[15] "EN 50128 – Railway applications – Communication, signalling and processing systems – Software for railway control and protection systems," CENELEC, June 2011.

[16] J. A. Davis, M. A. Clark, D. D. Cofer, A. Fifarek, J. Hinchman, J. A. Hoffman, B. W. Hulbert, S. P. Miller, and L. G. Wagner, "Study on the barriers to the industrial adoption of formal methods," in *Proceedings of the 18th International Workshop on Formal Methods for Industrial Critical Systems (FMICS'13)*, ser. LNCS, C. Pecheur and M. Dierkes, Eds., vol. 8187. Springer, 2013, pp. 63–77.

[17] M. Nyberg, D. Gurov, C. Lidström, A. Rasmusson, and J. Westman, "Formal verification in automotive industry: Enablers and obstacles," in *Proceedings of the 8th International Symposium on Leveraging Applications of Formal Methods, Verification and Validation: Industrial Practice (ISoLA'18)*, ser. LNCS, T. Margaria and B. Steffen, Eds., vol. 11247. Springer, 2018, pp. 139–158.

[18] A. Ferrari, F. Mazzanti, D. Basile, M. H. ter Beek, and A. Fantechi, "Comparing formal tools for system design: a judgment study," in *Proceedings of the 42nd ACM/IEEE International Conference on Software Engineering (ICSE'20)*. ACM, 2020, pp. 62–74.

[19] R. Schlick, M. Felderer, I. Majzik, R. Nardone, A. Raschke, C. F. Snook, and V. Vittorini, "A proposal of an example and experiments repository to foster industrial adoption of formal methods," in *Proceedings of the 8th International Symposium on Leveraging Applications of Formal Methods, Verification and Validation: Industrial Practice (ISoLA'18)*, ser. LNCS, T. Margaria and B. Steffen, Eds., vol. 11247. Springer, 2018, pp. 249–272.

[20] B. Steffen, "The physics of software tools: SWOT analysis and vision," *Int. J. Softw. Tools Technol. Transf.*, vol. 19, no. 1, pp. 1–7, 2017.

[21] J.-R. Abrial, "Formal methods: Theory becoming practice," *J. Univers. Comput. Sci.*, vol. 13, no. 5, pp. 619–628, 2007.

[22] M. Leuschel, J. Falampin, F. Fritz, and D. Plagge, "Automated property verification for large scale B models with ProB," *Form. Asp. Comput.*, vol. 23, no. 6, pp. 683–709, 2011.

[23] A. Ferrari, A. Fantechi, S. Gnesi, and G. Magnani, "Model-based development and formal methods in the railway industry," *IEEE Softw.*, vol. 30, no. 3, pp. 28–34, 2013.

[24] F. Moller, H. N. Nguyen, M. Roggenbach, S. Schneider, and H. Treharne, "Defining and model checking abstractions of complex railway models using CSP||B," in *Revised Selected Papers of the 8th International Haifa Verification Conference (HVC'12)*, ser. LNCS, A. Biere, A. Nahir, and T. Vos, Eds., vol. 7857. Springer, 2013, pp. 193–208.

[25] M. Ghazel, "Formalizing a subset of ERTMS/ETCS specifications for verification purposes," *Transport. Res. C-Emer.*, vol. 42, pp. 60–75, 2014.

[26] P. James, F. Moller, H. N. Nguyen, M. Roggenbach, S. Schneider, and H. Treharne, "Techniques for modelling and verifying railway interlockings," *Int. J. Softw. Tools Technol. Transf.*, vol. 16, no. 6, pp. 685–711, 2014.

[27] M. Bosschaart, E. Quaglietta, B. Janssen, and R. M. P. Goverde, "Efficient formalization of railway interlocking data in RailML," *Inf. Syst.*, vol. 49, pp. 126–141, 2015.

[28] R. Nardone, U. Gentile, M. Benerecetti, A. Peron, V. Vittorini, S. Marrone, and N. Mazzocca, "Modeling railway control systems in Promela," in *Revised Selected Papers of the 4th International Workshop on Formal Techniques for Safety-Critical Systems (FTSCS'15)*, ser. CCIS, C. Artho and P. C. Ölveczky, Eds., vol. 596. Springer, 2016, pp. 121–136.

[29] D. Basile, F. Di Giandomenico, and S. Gnesi, "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)*. ACM, 2017, pp. 1356–1363.

[30] Q. Cappart, C. Limbrée, P. Schaus, J. Quilbeuf, L. Traonouez, and A. Legay, "Verification of interlocking systems using statistical model checking," in *Proceedings of the 18th International Symposium on High Assurance Systems Engineering (HASE'17)*. IEEE, 2017, pp. 61–68.

[31] M. Ghazel, "A Control Scheme for Automatic Level Crossings Under the ERTMS/ETCS Level 2/3 Operation," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 10, pp. 2667–2680, 2017.

[32] T. Lecomte, D. Déharbe, É. Prun, and E. Mottin, "Applying a formal method in industry: A 25-year trajectory," in *Proceedings of the 20th Brazilian Symposium on Formal Methods: Foundations and Applications (SBMF'17)*, ser. LNCS, S. Cavalheiro and J. Fiadeiro, Eds., vol. 10623. Springer, 2017, pp. 70–87.

[33] L. H. Vu, A. E. Haxthausen, and J. Peleska, "Formal modelling and verification of interlocking systems featuring sequential release," *Sci. Comput. Program.*, vol. 133, pp. 91–115, 2017.

[34] M. Bartholomeus, B. Luttik, and T. Willemse, "Modeling and analysing ERTMS hybrid level 3 with the mCRL2 toolset," in *Proceedings of the 23rd International Conference on Formal Methods for Industrial Critical Systems (FMICS'18)*, ser. LNCS, F. Howar and J. Barnat, Eds., vol. 11119. Springer, 2018, pp. 98–114.

[35] D. Basile, M. H. ter Beek, and V. Ciancia, "Statistical model checking of a moving block railway signalling scenario with UPPAAL SMC: Experience and outlook," in *Proceedings of the 8th International Symposium on Leveraging Applications of Formal Methods, Verification and Validation: Verification (ISoLA'18)*, ser. LNCS, T. Margaria and B. Steffen, Eds., vol. 11245. Springer, 2018, pp. 372–391.

[36] U. Berger, P. James, A. Lawrence, M. Roggenbach, and M. Seisenberger, "Verification of the European Rail Traffic Management System in Real-Time Maude," *Sci. Comput. Program.*, vol. 154, pp. 61–88, 2018.

[37] A. Iliasov, D. Taylor, L. Laibinis, and A. B. Romanovsky, "Formal verification of signalling programs with SafeCap," in *Proceedings of the 37th International Conference on Computer Safety, Reliability, and Security (SAFECOMP 2018)*, ser. LNCS, B. Gallina, A. Skavhaug, and F. Bitsch, Eds., vol. 11093. Springer, 2018, pp. 91–106.

[38] F. Mazzanti and A. Ferrari, "Ten Diverse Formal Models for a CBTC Automatic Train Supervision System," in *Proceedings of the 3rd Workshop on Models for Formal Analysis of Real Systems and the 6th International Workshop on Verification and Program Transformation (MARS/VPT'18)*, ser. EPTCS, J. P. Gallagher, R. van Glabbeek, and W. Serwe, Eds., vol. 268, 2018, pp. 104–149.

[39] S. Vanit-Anunchai, "Modelling and simulating a Thai railway signalling system using Coloured Petri Nets," *Int. J. Softw. Tools Technol. Transf.*, vol. 20, no. 3, pp. 243–262, 2018.

[40] D. Basile, M. H. ter Beek, A. Ferrari, and A. Legay, "Modelling and analysing ERTMS L3 moving block railway signalling with Simulink and Uppaal SMC," in *Proceedings of the 24th International Conference on Formal Methods for Industrial Critical Systems (FMICS'19)*, ser. LNCS, K. G. Larsen and T. Willemse, Eds., vol. 11687. Springer, 2019, pp. 1–21.

[41] J. Abrial, "The ABZ-2018 case study with Event-B," *Int. J. Softw. Tools Technol. Transf.*, vol. 22, no. 3, pp. 257–264, 2020.

[42] P. Arcaini, J. Kofroň, and P. Ježek, "Validation of the hybrid ERTMS/ETCS level 3 using Spin," *Int. J. Softw. Tools Technol. Transf.*, vol. 22, no. 3, pp. 265–279, 2020.

[43] A. Cunha and N. Macedo, "Validating the hybrid ERTMS/ETCS level 3 concept with Electrum," *Int. J. Softw. Tools Technol. Transf.*, vol. 22, no. 3, pp. 281–296, 2020.

[44] D. Dghaym, M. Dalvandi, M. Poppleton, and C. Snook, "Formalising the hybrid ERTMS level 3 specification in iUML-B and Event-B," *Int. J. Softw. Tools Technol. Transf.*, vol. 22, no. 3, pp. 297–313, 2020.

[45] D. Hansen, M. Leuschel, P. Körner, S. Krings, T. Naulin, N. Nayeri, D. Schneider, and F. Skowron, "Validation and real-life demonstration of ETCS hybrid level 3 principles using a formal B model," *Int. J. Softw. Tools Technol. Transf.*, vol. 22, no. 3, pp. 315–332, 2020.

[46] A. Mammar, M. Frappier, S. J. Tueno Fotso, and R. Laleau, "A formal refinement-based analysis of the hybrid ERTMS/ETCS level 3 standard," *Int. J. Softw. Tools Technol. Transf.*, vol. 22, no. 3, pp. 333–347, 2020.

[47] S. J. Tueno Fotso, M. Frappier, R. Laleau, and A. Mammar, "Modeling the hybrid ERTMS/ETCS level 3 standard using a formal requirements engineering approach," *Int. J. Softw. Tools Technol. Transf.*, vol. 22, no. 3, pp. 349–363, 2020.

[48] D. Basile, M. H. ter Beek, and A. Legay, "Strategy synthesis for autonomous driving in a moving block railway system with Uppaal Stratego," in *Proceedings of the 40th IFIP WG 6.1 International Conference on Formal Techniques for Distributed Objects, Components, and Systems (FORTE'20)*, ser. LNCS, A. Gotsman and A. Sokolova, Eds., vol. 12136. Springer, 2020, pp. 3–21.

[49] D. Basile, M. H. ter Beek, A. Fantechi, A. Ferrari, S. Gnesi, L. Masullo, F. Mazzanti, A. Piattino, and D. Trentini, "Designing a demonstrator of formal methods for railways infrastructure managers," in *Proceedings of the 9th International Symposium on Leveraging Applications of Formal Methods: Applications (ISoLA'20)*, ser. LNCS, T. Margaria and B. Steffen, Eds., vol. 12478. Springer, 2020, pp. 467–485.

[50] F. Mazzanti, A. Ferrari, and G. O. Spagnolo, "Towards formal methods diversity in railways: an experience report with seven frameworks," *Int. J. Softw. Tools Technol. Transf.*, vol. 20, no. 3, pp. 263–288, 2018.

[51] A. E. Haxthausen, H. N. Nguyen, and M. Roggenbach, "Comparing formal verification approaches of interlocking systems," in *Proceedings of the 1st International Conference on Reliability, Safety, and Security of Railway Systems: Modelling, Analysis, Verification, and Certification (RSSRail'16)*, ser. LNCS, T. Lecomte, R. Pinger, and A. B. Romanovsky, Eds., vol. 9707. Springer, 2016, pp. 160–177.

[52] D. Basile, M. H. ter Beek, F. D. Giandomenico, A. Fantechi, S. Gnesi, and G. O. Spagnolo, "30 years of simulation-based quantitative analysis tools: A comparison experiment between Möbius and Uppaal SMC," in *Proceedings of the 9th International Symposium on Leveraging Applications of Formal Methods: Verification Principles (ISoLA'20)*, ser. LNCS, T. Margaria and B. Steffen, Eds., vol. 12476. Springer, 2020, pp. 368–384.

[53] A. Ferrari and M. H. ter Beek, "Formal methods in railways: a systematic mapping study," arXiv:2107.05413 **[cs.SE]**, July 2021.

[54] B. Kitchenham, S. Linkman, and D. Law, "DESMET: a methodology for evaluating software engineering methods and tools," *Comput. Control. Eng. J.*, vol. 8, no. 3, pp. 120–126, 1997.

[55] D. Basile, M. H. ter Beek, A. Fantechi, S. Gnesi, F. Mazzanti, A. Piattino, D. Trentini, and A. Ferrari, "On the industrial uptake of formal methods in the railway domain: A survey with stakeholders," in *Proceedings of the 14th International Conference on Integrated Formal Methods (iFM'18)*, ser. LNCS, C. A. Furia and K. Winter, Eds., vol. 11023. Springer, 2018, pp. 20–29.

[56] M. H. ter Beek, A. Borälv, A. Fantechi, A. Ferrari, S. Gnesi, C. Löfving, and F. Mazzanti, "Adopting formal methods in an industrial setting: The railways case," in *Proceedings of the 3rd World Congress on Formal Methods (FM'19)*, ser. LNCS, M. H. ter Beek, A. McIver, and J. N. Oliveira, Eds., vol. 11800. Springer, 2019, pp. 762–772.

[57] F. Mazzanti, A. Ferrari, D. Basile, and M. ter Beek, "Systematic Evaluation and Usability Analysis of Formal Tools for Railway System Design - Technical Annexes," Zenodo, August 2021. [Online]. Available: https://doi.org/10.5281/zenodo.5171939

[58] K.-J. Stol and B. Fitzgerald, "The ABC of software engineering research," *ACM Trans. Softw. Eng. Methodol.*, vol. 27, no. 3, pp. 11:1–11:51, 2018.

[59] M. H. ter Beek, S. Gnesi, and A. Knapp, "Formal methods for transport systems," *Int. J. Softw. Tools Technol. Transf.*, vol. 20, no. 3, pp. 237–241, 2018.

[60] D. Duffy, *Principles of Automated Theorem Proving*. John Wiley & Sons, 1991.

[61] J. A. Robinson and A. Voronkov, Eds., *Handbook of Automated Reasoning*. Elsevier, 2001.

[62] Y. Bertot and P. Castéran, *Interactive Theorem Proving and Program Development*. Springer, 2004.

[63] E. M. Clarke, O. Grumberg, and D. A. Peled, *Model Checking*. MIT Press, 1999.

[64] C. Baier and J.-P. Katoen, *Principles of Model Checking*. MIT Press, 2008.

[65] E. M. Clarke, T. A. Henzinger, H. Veith, and R. Bloem, Eds., *Handbook of Model Checking*. Springer, 2018.

[66] T. Gibson-Robinson, P. J. Armstrong, A. Boulgakov, and A. W. Roscoe, "FDR3: a parallel refinement checker for CSP," *Int. J. Softw. Tools Technol. Transf.*, vol. 18, no. 2, pp. 149–167, 2016.

[67] *Atelier B User Manual Version 4.0*, ClearSy, 2002.

[68] Y. Bertot, "A short presentation of Coq," in *Proceedings of the 21st International Conference on Theorem Proving in Higher Order Logics (TPHOLs'08)*, ser. LNCS, O. A. Mohamed, C. A. Muñoz, and S. Tahar, Eds., vol. 5170. Springer, 2008, pp. 12–16.

[69] L. C. Paulson, *Isabelle: A Generic Theorem Prover*, ser. LNCS. Springer, 1994, vol. 828.

[70] S. Owre and N. Shankar, "A brief overview of PVS," in *Proceedings of the 21st International Conference on Theorem Proving in Higher Order Logics (TPHOLs'08)*, ser. LNCS, O. A. Mohamed, C. A. Muñoz, and S. Tahar, Eds., vol. 5170. Springer, 2008, pp. 22–27.

[71] G. J. Holzmann, *The Spin Model Checker: Primer and Reference Manual*. Addison-Wesley, 2003.

[72] M. H. ter Beek, A. Fantechi, S. Gnesi, and F. Mazzanti, "A state/event-based model-checking approach for the analysis of abstract system properties," *Sci. Comput. Program.*, vol. 76, no. 2, pp. 119–135, 2011.

[73] K. L. McMillan, *Symbolic Model Checking*. Kluwer, 1993.

[74] A. Cimatti, E. M. Clarke, F. Giunchiglia, and M. Roveri, "NuSMV: A new symbolic model checker," *Int. J. Softw. Tools Technol. Transf.*, vol. 2, no. 4, pp. 410–425, 2000.

[75] R. Cavada, A. Cimatti, M. Dorigatti, A. Griggio, A. Mariotti, A. Micheli, S. Mover, M. Roveri, and S. Tonetta, "The nuXmv symbolic model checker," in *Proceedings of the 26th International Conference on Computer Aided Verification (CAV'14)*, ser. LNCS, A. Biere and R. Bloem, Eds., vol. 8559. Springer, 2014, pp. 334–342.

[76] M. Kwiatkowska, G. Norman, and D. Parker, "PRISM 4.0: Verification of probabilistic real-time systems," in *Proceedings of the 23rd International Conference on Computer Aided Verification (CAV'11)*,

ser. LNCS, G. Gopalakrishnan and S. Qadeer, Eds., vol. 6806. Springer, 2011, pp. 585–591.

[77] G. Agha and K. Palmskog, "A survey of statistical model checking," *ACM Trans. Model. Comput. Simul.*, vol. 28, no. 1, pp. 6:1–6:39, 2018.

[78] A. Legay, A. Lukina, L. Traonouez, J. Yang, S. A. Smolka, and R. Grosu, "Statistical model checking," in *Computing and Software Science: State of the Art and Perspectives*, ser. LNCS, B. Steffen and G. J. Woeginger, Eds. Springer, 2019, vol. 10000, pp. 478–504.

[79] A. David, K. G. Larsen, A. Legay, M. Mikučionis, and D. B. Poulsen, "UPPAAL SMC tutorial," *Int. J. Softw. Tools Technol. Transf.*, vol. 17, no. 4, pp. 397–415, 2015.

[80] H. Garavel, F. Lang, R. Mateescu, and W. Serwe, "CADP 2011: a toolbox for the construction and analysis of distributed processes," *Int. J. Softw. Tools Technol. Transf.*, vol. 15, no. 2, pp. 89–107, 2013.

[81] J. F. Groote and M. R. Mousavi, *Modeling and Analysis of Communicating Systems*. MIT Press, 2014.

[82] L. M. de Moura, S. Owre, H. Rueß, J. M. Rushby, N. Shankar, M. Sorea, and A. Tiwari, "SAL 2," in *Proceedings of the 16th International Conference on Computer Aided Verification (CAV'04)*, ser. LNCS, R. Alur and D. A. Peled, Eds., vol. 3114. Springer, 2004, pp. 496–500.

[83] J. M. Rushby, "Tutorial: Automated Formal Methods with PVS, SAL, and Yices," in *Proceedings of the 4th International Conference on Software Engineering and Formal Methods (SEFM'06)*. IEEE, 2006, p. 262.

[84] L. Lamport, *Specifying Systems: The TLA+ Language and Tools for Hardware and Software Engineers*. Addison-Wesley, 2002.

[85] K. Jensen, L. M. Kristensen, and L. Wells, "Coloured Petri Nets and CPN Tools for modelling and validation of concurrent systems," *Int. J. Softw. Tools Technol. Transf.*, vol. 9, no. 3-4, pp. 213–254, 2007.

[86] P. Fritzson, *Principles of Object Oriented Modeling and Simulation with Modelica 3.3: A CyberPhysical Approach*. IEEE/Wiley, 2014.

[87] G. Berry, "SCADE: Synchronous design and validation of embedded control software," in *Proceedings of the General Motors R&D Workshop*, S. Ramesh and P. Sampath, Eds. Springer, 2007, pp. 19–33.

[88] J. B. Dabney and T. L. Harman, *Mastering Simulink*. Pearson, 2003.

[89] M. Gleirscher and D. Marmsoler, "Formal methods in dependable systems engineering: A survey of professionals from Europe and North America," *Empir. Softw. Eng.*, vol. 25, no. 6, pp. 4473–4546, 2020.

[90] S. L. Gerhart, D. Craigen, and T. Ralston, "Experience with formal methods in critical systems," *IEEE Softw.*, vol. 11, no. 1, pp. 21–28, January 1994.

[91] M. A. Ardis, J. A. Chaves, L. J. Jagadeesan, P. Mataga, C. Puchol, M. G. Staskauskas, and J. V. Olnhausen, "A framework for evaluating specification methods for reactive systems: Experience report," *IEEE Trans. Softw. Eng.*, vol. 22, no. 6, pp. 378–389, 1996.

[92] A. Abbassi, A. Bandali, N. A. Day, and J. Serna, "A comparison of the declarative modelling languages B, Dash, and TLA$^+$," in *Proceedings of the 8th IEEE International Model-Driven Requirements Engineering Workshop (MoDRE@RE'18)*. IEEE, 2018, pp. 11–20.

[93] E. Bartocci, D. Beyer, P. E. Black, G. Fedyukovich, H. Garavel, A. Hartmanns, M. Huisman, F. Kordon, J. Nagele, M. Sighireanu, B. Steffen, M. Suda, G. Sutcliffe, T. Weber, and A. Yamada, "TOOLympics 2019: An overview of competitions in formal methods," in *Proceedings of the 25th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'19)*, ser. LNCS, D. Beyer, M. Huisman, F. Kordon, and B. Steffen, Eds., vol. 11429. Springer, 2019, pp. 3–24.

[94] B. Steffen, T. Margaria, and V. Braun, "The Electronic Tool Integration platform: concepts and design," *Int. J. Softw. Tools Technol. Transf.*, vol. 1, no. 1-2, pp. 9–30, 1997.

[95] D. Bjørner, "New results and trends in formal techniques & tools for the development of software for transportation systems: A review," in *Proceedings of the 4th Symposium on Formal Methods for Railway Operation and Control Systems (FORMS'03)*, G. Tarnai and E. Schnieder, Eds. L'Harmattan, 2003.

[96] A. Fantechi, W. Fokkink, and A. Morzenti, "Some trends in formal methods applications to railway signaling," in *FMICS: A Survey of Applications*, S. Gnesi and T. Margaria, Eds. IEEE/Wiley, 2013, ch. 4, pp. 61–84.

[97] F. Flammini, Ed., *Railway Safety, Reliability, and Security: Technologies and Systems Engineering*. IGI Global, 2012.

[98] J.-L. Boulanger, Ed., *Formal Methods Applied to Industrial Complex Systems: Implementation of the B Method*. John Wiley & Sons, 2014.

[99] T. S. Hoang, M. J. Butler, and K. Reichl, "The hybrid ERTMS/ETCS level 3 case study," in *Proceedings of the 6th International Conference on Abstract State Machines, Alloy, B, TLA, VDM, and Z (ABZ'18)*, ser. LNCS, M. J. Butler, A. Raschke, T. S. Hoang, and K. Reichl, Eds., vol. 10817. Springer, 2018, pp. 251–261.

[100] M. Butler, T. S. Hoang, A. Raschke, and K. Reichl, "Introduction to the special section on the ABZ 2018 case study: Hybrid ERTMS/ETCS level 3," *Int. J. Softw. Tools Technol. Transf.*, vol. 22, no. 3, pp. 249–255, 2020.

[101] C. Newcombe, "Why Amazon chose TLA$^+$," in *Proceedings of the 4th International Conference on Abstract State Machines, Alloy, B, TLA, VDM, and Z (ABZ'14)*, ser. LNCS, Y. A. Ameur and K. Schewe, Eds., vol. 8477. Springer, 2014, pp. 25–39.

[102] B. Kitchenham, "Procedures for performing systematic reviews," Keele University, Tech. Rep. TR/SE-0401, July 2004, and National ICT Australia, Tech. Rep. 0400011T.1.

[103] R. Scupin, "The KJ method: A technique for analyzing data derived from Japanese ethnology," *Hum. Organ.*, vol. 56, no. 2, pp. 233–237, 1997.

[104] J. Brooke, "SUS: A 'quick and dirty' usability scale," in *Usability Evaluation in Industry*, P. W. Jordan, B. Thomas, B. A. Weerdmeester, and I. L. McClelland, Eds. CRC press, 1996, ch. 21, pp. 189–194.

[105] ——, "SUS: A retrospective," *J. Usability Stud.*, vol. 8, no. 2, pp. 29–40, 2013.

[106] P. Pourali and J. M. Atlee, "An empirical investigation to understand the difficulties and challenges of software modellers when using modelling tools," in *Proceedings of the 21th ACM/IEEE International Conference on Model Driven Engineering Languages and Systems (MODELS'18)*. ACM, 2018, pp. 224–234.

[107] A. Bangor, P. T. Kortum, and J. T. Miller, "An empirical evaluation of the system usability scale," *Int. J. Hum. Comput. Interact.*, vol. 24, no. 6, pp. 574–594, 2008.

[108] W. Hwang and G. Salvendy, "Number of people required for usability evaluation: The 10±2 rule," *Commun. ACM*, vol. 53, no. 5, pp. 130–133, 2010.

[109] N. Furness, H. van Houten, L. Arenas, and M. Bartholomeus, "ERTMS level 3: the game-changer," *IRSE News*, vol. 232, pp. 2–9, April 2017.

[110] P. Stroh, P. Melia, and S. Bilgici, "Deliverable D2.1: Determine the long-term changes in future needs," NEAR2050, January 2018.

[111] A. Ferrari, G. Magnani, D. Grasso, and A. Fantechi, "Model checking interlocking control tables," in *Proceedings of the 8th Symposium on Formal Methods for Automation and Safety in Railway and Automotive Systems (FORMS/FORMAT'10)*, E. Schnieder and G. Tarnai, Eds. Springer, 2010, pp. 107–115.

[112] "ISO 9241-11:2018 – Ergonomics of human-system interaction – Part 11: Usability: Definitions and concepts," ISO/TC 159/SC 4, March 2018.

[113] L. Rierson, *Developing Safety-Critical Software: A Practical Guide for Aviation Software and DO-178C Compliance*. CRC Press, 2017.

[114] Y. A. Ameur, F. Boniol, and V. Wiels, "Toward a wider use of formal methods for aerospace systems design and verification," *Int. J. Softw. Tools Technol. Transf.*, vol. 12, no. 1, pp. 1–7, 2010.

[115] J. M. Atlee, S. Beidu, N. A. Day, F. Faghih, and P. Shaker, "Recommendations for improving the usability of formal methods for product lines," in *Proceedings of the 1st FME Workshop on Formal Methods in Software Engineering (FormaliSE'13)*. IEEE, 2013, pp. 43–49.

[116] A. van Lamsweerde, "Formal specification: a roadmap," in *Proceedings of the Future of Software Engineering Track of the 22nd International Conference on Software Engineering (ICSE'00)*. ACM, 2000, pp. 147–159.

[117] A. Ottensooser, A. D. Fekete, H. A. Reijers, J. Mendling, and C. Menictas, "Making sense of business process descriptions: An experimental comparison of graphical and textual notations," *J. Syst. Softw.*, vol. 85, no. 3, pp. 596–606, 2012.

[118] M. M. Moscato, L. Titolo, A. Dutle, and C. A. Muñoz, "Automatic estimation of verified floating-point round-off errors via static analysis," in *Proceedings of the 36th International Conference on Computer Safety, Reliability, and Security (SAFECOMP'17)*, ser. LNCS, S. Tonetta, E. Schoitsch, and F. Bitsch, Eds., vol. 10488. Springer, 2017, pp. 213–229.

[119] M. Gleirscher, S. Foster, and J. Woodcock, “New opportunities for integrated formal methods,” *ACM Comput. Surv.*, vol. 52, no. 6, pp. 117:1–117:36, 2020.

[120] “Technology readiness assessment guide: Best practices for evaluating the readiness of technology for use in acquisition programs and projects,” Tech. Rep., January 2020.

[121] D. Bjørner and K. Havelund, “40 years of formal methods: Some obstacles and some possibilities?” in *Proceedings of the 19th International Symposium on Formal Methods (FM’14)*, ser. LNCS, C. Jones, P. Pihlajasaari, and J. Sun, Eds., vol. 8442. Springer, 2014, pp. 42–61.

[122] C. Newcombe, T. Rath, F. Zhang, B. Munteanu, M. Brooker, and M. Deardeuff, “How Amazon Web Services uses formal methods,” *Commun. ACM*, vol. 58, no. 4, pp. 66–73, 2015.

[123] A. G. Russo Jr., “Formal methods as an improvement tool,” in *Industrial Deployment of System Engineering Methods*, A. B. Romanovsky and M. Thomas, Eds. Springer, 2013, pp. 81–95.

[124] S. P. Miller, “Lessons from twenty years of industrial formal methods,” in *Proceedings of the 20th High Confidence Software and Systems Conference (HCSS’12)*, 2012.

[125] M. van der Werff, B. Elsweiler, B. Luttik, and P. Hendriks, “The use of formal methods in standardisation of interfaces of signalling systems,” *IRSE News*, vol. 256, pp. 1–3, June 2019.

[126] M. Bartholomeus, B. Luttik, T. Willemse, D. Hansen, M. Leuschel, P. Hendriks, and P. Hendriks, “The use of formal methods in specification and demonstration of ERTMS Hybrid Level 3,” *IRSE News*, vol. 260, pp. 14–17, November 2019.

[127] A. Ferrari, A. Fantechi, and S. Gnesi, “Lessons learnt from the adoption of formal model-based development,” in *Proceedings of the 4th International Symposium NASA Formal Methods (NFM’12)*, ser. LNCS, A. Goodloe and S. Person, Eds., vol. 7226. Springer, 2012, pp. 24–38.

[128] M. Huisman, D. Gurov, and A. Malkis, “Formal methods: From academia to industrial practice. A travel guide,” arXiv:2002.07279 **[cs.SE]**, February 2020.

Alessio Ferrari is a research scientist at ISTI-CNR (Pisa, Italy) and member of the Formal Methods and Tools lab. His main research interests are requirements engineering, empirical software engineering, and applications of formal methods. In these fields, he has published over 80 peer-reviewed papers in conferences and journals. He has been working as system engineer at General Electric Transportation Systems s.p.a., a world leading railway signaling manufacturer, where he used formal methods and code generation for the development of railway systems. He has been workpackage leader of the ASTRail project, and he is currently involved in the 4SECURail project, funded under the Shift2Rail initiative.

Franco Mazzanti Franco Mazzanti is a senior researcher at ISTI–CNR (Pisa, Italy) and member of the Formal Methods and Tools lab. He is the main designer and author of the KandISTI formal verification framework, which includes the explicit, on-the-fly model-checking tools CMC, FMC, UMC and VMC. He has authored more than 40 papers in the field of formal methods. His research focuses on i) the design and support of state- and event-based branching-time temporal logics for the specification and evaluation of system requirements, and ii) the exploitation of formal methods and tools diversity for the analysis of concurrent, asynchronous systems. He has applied his research in many EU projects, among which ASTRail and 4SECURail.

Davide Basile Davide Basile is a researcher at ISTICNR (Pisa, Italy) and a member of the Formal Methods and Tools lab. He has authored about 40 peer-reviewed papers in conferences and journals in the field of formal methods, software engineering and dependable computing. His research focuses on developing both novel formalisms for emerging computational paradigms and supporting tools, exploring formal verification techniques and applying state-of-the-art formal methods to the design of real-world systems and emerging technologies in the railway domain. He participated in several projects in collaboration with railway companies, including SISTER, with the participation of Thales s.p.a., and ASTRail. He is currently involved in the 4SECURail project, funded under the Shift2Rail initiative.

Maurice H. ter Beek Maurice is a senior researcher at ISTI–CNR (Pisa, Italy) and head of the Formal Methods and Tools lab. He obtained a Ph.D. at Leiden University, The Netherlands. He has authored over 150 peer-reviewed papers, edited over 25 proceedings and special issues of journals, and he serves on the editorial board of the Journal of Logical and Algebraic Methods in Programming, Science of Computer Programming, PeerJ Computer Science and ERCIM News. He works on formal methods and model-checking tools for the specification and verification of safety-critical software systems and communication protocols, focusing in particular on applications in service-oriented computing, software product line engineering and railway systems.