Formal Methods in Railways: a Systematic Mapping Study

ALESSIO FERRARI and MAURICE H. TER BEEK, Istituto di Scienza e Tecnologie dell'Informazione, Consiglio Nazionale delle Ricerche (ISTI–CNR), Italy

Formal methods are mathematically based techniques for the rigorous development of software-intensive systems. The railway signaling domain is a field in which formal methods have traditionally been applied, with several success stories. This article reports on a mapping study that surveys the landscape of research on applications of formal methods to the development of railway systems. Following the guidelines of systematic reviews, we identify 328 relevant primary studies, and extract information about their demographics, the characteristics of formal methods used and railway-specific aspects. Our main results are as follows: (i) we identify a total of 328 primary studies relevant to our scope published between 1989 and 2020, of which 44% published during the last 5 years and 24% involving industry; (ii) the majority of studies are evaluated through Examples (41%) and Experience Reports (38%), while full-fledged Case Studies are limited (1.5%); (iii) Model checking is the most commonly adopted technique (47%), followed by simulation (27%) and theorem proving (19.5%); (iv) the dominant languages are UML (18%) and B (15%), while frequently used tools are ProB (9%), NuSMV (8%) and UPPAAL (7%); however, a diverse landscape of languages and tools is employed; (v) the majority of systems are interlocking products (40%), followed by models of high-level control logic (27%); (vi) most of the studies focus on the Architecture (66%) and Detailed Design (45%) development phases. Based on these findings, we highlight current research gaps and expected actions. In particular, the need to focus on more empirically sound research methods, such as Case Studies and Controlled Experiments, and to lower the degree of abstraction, by applying formal methods and tools to development phases that are closer to software development. Our study contributes with an empirically based perspective on the future of research and practice in formal methods applications for railways. It can be used by formal methods researchers to better focus their scientific inquiries, and by railway practitioners for an improved understanding of the interplay between formal methods and their specific application domain.

CCS Concepts: • **General and reference → Surveys and overviews**; • **Software and its engineering → Formal methods**; *Model checking*; *Automated static analysis*; *Software verification*; *Software design engineering*; **Software development methods**; *V-model*; *Software verification and validation*; *Formal software verification*; *Software reliability*; *Software fault tolerance*; *Software safety*; **Model-driven software engineering**; **System modeling languages**; *Specification languages*; *Petri nets*; *Unified Modeling Language (UML)*; *Domain specific languages*; • **Computing methodologies → Modeling and simulation**; **Model verification and validation**; • **Applied computing**;

Additional Key Words and Phrases: formal methods, semi-formal methods, model-based development, model checking, theorem proving, static analysis, railway systems, railway signaling, interlocking.

ACM Reference Format:
Alessio Ferrari and Maurice H. ter Beek. 0000. Formal Methods in Railways: a Systematic Mapping Study. *ACM Comput. Surv.* 0, 0, Article 0 (0000), 35 pages. https://doi.org/0000001.0000001

Authors' address: Alessio Ferrari, alessio.ferrari@isti.cnr.it; Maurice H. ter Beek, maurice.terbeek@isti.cnr.it, Istituto di Scienza e Tecnologie dell'Informazione, Consiglio Nazionale delle Ricerche (ISTI–CNR), Via G. Moruzzi, 1, Pisa, 56124, Italy.

0360-0300/0000/0-ART0 $15.00
https://doi.org/0000001.0000001

1 INTRODUCTION

The railway signaling domain has traditionally been a fruitful playground for formal methods. The extensive survey of Woodcock *et al.* [97] recognised transportation, including railways, as a primary field in which formal methods have been applied, also for the development of real-world railway platforms. Well-known projects are Line 14 of the Paris Metro and the driverless Paris–Roissy Airport shuttle, developed with the B method [2], the metro control system of Rio de Janeiro, developed with the support of Simulink/Stateflow [50], and the verification of the ERTMS/ETCS European standard for railway control and management with NuSMV [28]. A set of international joint projects has also been funded on formal methods for railways starting from 1998 (14 projects in total were counted until 2018, cf. [53]). Notable cases include OpenETCS (http://openetcs.org) and the more recent 4SECURail (https://www.4securail.eu) and X2Rail-1 (https://cordis.europa.eu/project/id/730640). Projects in the field have recently seen a particular boost also thanks to the Shift2Rail (https://shift2rail.org/) initiative. This is a joint effort of railway stakeholders and the European Commission to advance the railway field through innovative research projects involving academia and industry. Shift2Rail considers formal methods to be fundamental to the provision of safe and reliable technological advances in railways.

Surveys on formal methods in industry, including railways, have already appeared in the literature. Some focus on providing personal overviews of past experiences [1, 2, 30] or on collecting viewpoints of experts [56, 58]; others target the railway field specifically, with questionnaires [9, 10], discussion of future challenges [18, 46], and comparison of tools in the domain [13, 51, 52, 76]. However, despite the interest of the industry and research communities, there is no systematic study aimed at collecting and analyzing the existing literature in formal methods for railways to provide a framework to move forward in research and practice. This is particularly needed, as the world of formal methods is vast, and practitioners often face a paradox of choice in selecting formal techniques [52].

This paper presents the first systematic mapping study on formal methods in the railway domain. We retrieve and select 328 high-quality research papers from the literature in the time span 1989–2020, and we categorize them according to three different facets: *demographic and empirical*, identifying years, publication venues and research methods used; *formal methods*, categorizing techniques, tools and languages; *railway*, concerning systems and development phases addressed by the research. Furthermore, we perform a stratified analysis to understand which are the characteristics of the studies concerned with industrial applications, and what are the main trends of the last years.

Our results show that formal methods for railways is a thriving research field with a strong industrial bound, since 143 studies were published solely in the last five years (44% of the total), and 79 studies (24%) involve industry. Most of the studies focus on non-standard interlocking applications, and on high-level modeling and early development phases. In terms of languages and tools, the landscape is highly diversified. The dominant languages are UML (18%) and B (15%), while frequently used tools are ProB (9%), NuSMV (8%) and UPPAAL (7%), but a long tail exists in the statistics. The empirical maturity of the field is still limited, as many papers present only examples or experience reports. Our work thus calls for more empirical rigor in the field, with case studies, which can leverage the strong link with industries, and controlled experiments, which can address issues related to the learnability of formal methods and aspects related to human factors. Furthermore, we encourage applications that operate on later railway development phases, and on lower-level models and code, which received less attention so far. Finally, based on our findings, and in line with the needs for interoperability, we also support focusing more on modeling and verifying standard systems.

The remainder of the paper is structured as follows. In Sect. 2 we present background on formal methods and railways, and we discuss related reviews to motivate the current study. Sect. 3 describes the review method. Sect. 4 presents the results, and Sect. 5 discusses the empirical findings. Sect. 6 reports threats to validity, and Sect. 7 concludes the paper.

2 BACKGROUND AND MOTIVATION

2.1 Formal Techniques

Formal methods are rigorous mathematics-based techniques and tools for the specification (*modeling*) and manual or automated verification (analysis) of software or hardware systems or system designs [56, 97]. *Semi-formal methods* refer to techniques and tools that are not fully formal, i.e., lacking a precise and unambiguously defined syntax and semantics; a prominent example is the Unified modeling Language (UML).

Model-based development relies on rigorous techniques (e.g., the B family [25]) to derive a concrete (low-level) implementation from an abstract (high-level) specification by successive *refinement* steps based on *model transformation*. During refinement, a specification is complemented with details that are unnecessary in the higher-level specifications. Model-based development usually involves some semi-formal methods and it is typically complemented with (automatic) *code generation*, which generates source code that is by definition consistent with the model it is generated from.

Formal verification concerns (exhaustive) verification that critical system properties related to safety and security are satisfied, i.e., verifying correctness of the system (model) which dynamic analysis methods based on *simulation* or *model-based testing* cannot. Since neither of the latter two techniques explore all possible system behavior (state space), a counterexample found by either testing or simulation demonstrates an error but the lack of counterexamples does not prove absence of errors. The success of testing moreover depends on the quality of *test generation*, which generates appropriate sequences of input values that guarantee the models to satisfy specific testing criteria (e.g., model coverage).

Formal verification is often (but not always) automated, and the resulting tools can reduce the effort and time needed to prove the correctness of systems considerably. Formal verification is supported by several families of techniques, the most important ones being *theorem proving* [85] and *model checking* [29], including probabilistic [7] and statistical [3] approaches.

Model checking verifies whether a system model meets a given specification, typically formulated in (temporal) logic; model checkers automate this process. Model checking explores all possible system behavior in the form of a (reachability) graph, possibly constructed on the fly. Model checking thus allows to determine the absence of errors in a system model and it moreover produces a counterexample that demonstrates how the error can be produced. *Reachability analysis* is used to determine which states of a model can be reached, thus suffering from the state-space explosion problem inherent to model checking: as the number of state variables of a system increases, the size of the system's state space grows exponentially. Advanced techniques such as *symbolic* or *bounded* model checking alleviate this problem, sometimes in combination with SAT/SMT solving [12].

Constraint satisfaction problems are decision problems stated in the form of a set of constraints that can be solved with *SAT solving*, *SMT solving* or (integer) *linear programming solving* techniques [93]. *Static analysis* is another abstract interpretation technique to detect erroneous run-time behavior at compile-time, typically by computing over-approximations, i.e., including behavior that cannot actually occur. *Static checking* is another compile-time technique, which catches syntax or typing errors, i.e., errors that are independent of specific variable values.

Theorem proving uses deductive reasoning to provide a proof in symbolic logic by inference; theorem provers automate much of this process [89]. Contrary to model checking, which is largely limited to finite models and propositional logic, theorem proving can handle infinite state spaces and many theorem provers moreover support automatic code generation.

Correct-by-construction approaches such as (supervisory control) *synthesis* [26] concern the creation of (program or system) models that provably satisfy a high-level formal specification. In supervisory control synthesis, starting from a model of the uncontrolled system and a model of the behavioral requirements, a supervisory controller model is synthesized; a few tools to do so exist. Such a supervisory controller thus influences system behavior by disabling controllable events to guarantee system correctness with respect to the requirements (e.g., safety properties).

2.2 Railway Signaling Systems

Railway signaling systems are complex, dependable cyber-physical platforms, composed of interacting subsystems with different safety-critical levels. These systems also have diverse applications, from traditional heavy rails, to light rapid rails and to metro lines.

Signaling subsystems can be distinguished between those that mainly control the transit of trains at the stations, and those that mainly ensure safety along the lines. In a station, the most important subsystem is the so-called *interlocking*, a safety critical platform that controls points and signals, and all the wayside entities, as, e.g., the elements to identify the presence of a train in a specific portion of the line (Axle Counters, or Track Circuits [5]). By monitoring and setting the status of the entities, the interlocking can enable safe train routing. In advanced metro systems, train routing is commanded by so the called Automatic Train Supervision (ATS) platform, while in more traditional systems a human command is issued.

Once a train is routed, preservation of the safety distance from other trains needs to be ensured. This is supported by Automatic Train Protection (ATP) platforms, which are composed of a wayside subsystem and an on-board subsystem. The wayside ATP monitors the position of the train, and makes preceding trains aware that the portion of the line in front of them is occupied. The on-board ATP receives information from the wayside one, and monitors the so-called braking curve, issuing an emergency brake in case there is a risk of collision. Along the lines, a reduced form of interlocking, namely the Railway Crossing Controller, ensures the safety of level crossings. Advanced metro systems also include an automatic driver called Automatic Train Operation (ATO).

Given the need to ensure interoperability between the different subsystems described, product standards have been defined by international organizations. The most important product standard for heavy rail is the European Rail Traffic Management System/European Train Control System (ERTMS/ETCS)[1]. The standard foresees four levels of automation (0 to 3), and from level 2 it includes the so-called Radio Block Center (RBC), a radio-based wayside ATP. The standard provides very detailed requirements specifications, as its goal is to ensure that subsystems developed by different vendors are able to seamlessly communicate, so to encourage competition. Another known standard for heavy rails is the Chinese Train Control System (CTCS), which is analogous to the ERTMS/ETCS in terms of goals. A well-known product standard, oriented to metro lines, is the Communications-based Train Control (CBTC) system, also known as Urban Guided Transport Management and Command/Control System (UGTMS). Two international standards provide general requirements for CBTC systems, IEEE 1474.1-2004 [67] and IEC 62290 [68]. The main characteristic of CBTC, shared also with ERTMS/ETCS Level 3, is the concept of *moving block*. In a nutshell, this concept consists of computing the safety distance between trains considering the exact position each train, instead of considering as its position the segment of the line occupied by the train. The wayside ATP for CBTC systems is frequently called Zone Controller (ZC), though its name might depend on the vendor.

[1] https://www.era.europa.eu/activities/european-rail-traffic-management-system-ertms_en

The railway field is particularly lively in terms of innovation efforts, especially thanks to the increased sensitivity of the global community towards green transportation. At this regard, the Shift2Rail program (https://shift2rail.org/) is an unprecedented joint effort by the European rail sector and the European Union (EU), tripling EU-funding to nearly €1 Billion for rail research, innovation, and demonstration across the 7-year lifespan of the initiative, to move European Railway forward. In November 2015, the Shift2Rail Multi-Annual Action Plan was adopted and in June 2016 Shift2Rail awarded the first grants. Over 100 projects have been financed so far, including projects in which formal methods play a prominent role (ASTRail, X2Rail-1, X2Rail-2, 4SecuRail, *etc.*[2]). A successor to the Shift2Rail joint undertaking, Europe's Rail, has recently been announced[3].

2.3 Related Reviews

Formal methods have been studied in academia and applied in industry for quite some time now, as witnessed by introductions from the early '90s to the use of formal methods in developing safety-critical software systems from academia [83, 87, 95] and industry [91, 92]. Further historical references from long-time advocates of formal methods reflect on the industrial application of formal methods through the metaphors of seven (and seven more) myths [20, 61] and ten commandments [21, 22] to eventually realize their benefits [62]. Also worth mentioning are early experiences and perspectives on industrial applications of formal methods [24, 57], as well as the first systematic survey and analysis of the use of formal methods in the development of industrial applications [33], publicized by several works [31, 34, 35]. This extensive survey is based on twelve case studies from industry, including one from the railway signaling domain, namely the development of ATP systems for the subways of Paris and Calcutta [36, 60]. The mid-'90s were also characterized by panels and round-table discussions, involving academics and practitioners alike, on the (future) use of formal methods in industry [19, 63].

The classical 1996 survey on formal methods [30] illustrates a number of case studies in specification and verification, including the one from the railway signaling domain described in [33] plus an additional one on specifying the signaling rules of railway interlocking systems [70].

At the turn of the century, several personal, non-systematic surveys, often based on previous surveys and involving practitioners, were published [15, 32, 65]; Dietrich and Hubaux [39] present a more extensive survey of the use of formal methods for communication services both in academia and in industry. Shortly after, the first non-systematic surveys of formal methods in the railway domain were published [13, 84]; these are both very personal, informal reviews of formal techniques and tools and exemplary applications to railway systems. Also worth mentioning are a tutorial introduction to the B method [2] and a brief description and discussion of two of its best-known applications in industry [1]: the development of safety-critical parts of the subway line 14 and the Roissy airport shuttle of Paris [6, 11].

The classical 2009 survey on formal methods [97], "perhaps the most comprehensive review ever made of formal methods application in industry", reviews the application of formal methods in 62 different industrial projects world-wide, in all but 6 cases by collecting data directly from individuals who had been involved in the projects. One of the eight highlighted projects is from the railway domain. The paper also provides an overview of 20 years of surveys on formal methods in industry, including all surveys mentioned above.

The last decade has seen several introductions and accounts of trends and experiences concerning the role of formal methods in the development of safety-critical applications [23, 59, 64], in particular in the railway domain and from an academic viewpoint by Fantechi et al. [45–47, 49] and in the

[2] https://projects.shift2rail.org/s2r_projects.aspx
[3] https://ec.europa.eu/commission/presscorner/detail/en/ip_21_702

general transportation domain and from an industrial viewpoint, focusing mostly on SCADE and/or the B method, by Boulanger [16–18]. Further studies are lessons learned and obstacles found with respect to decades of integrating formal methods in research, education and industrial practice, in particular in the transportation domain [14, 25, 38, 73, 79]. We also mention a number of recent surveys in the railway domain at large [44, 66, 74, 75, 81, 94, 98], neither of which involve formal methods. The extensive report from Garavel and Graf [54] provides a state-of-the-art account on the use of formal methods in academia and industry, and a large number of success stories: a carefully selected list of 30, well-documented case studies from three decades (one per year from the period 1982–2011), including a large number of railway cases. Finally, we mention three recent questionnaire surveys on the use of formal methods in the railway domain [9, 10, 53], conducted with both academic and industrial stakeholders in the context of Shift2Rail, which identifies the main formal methods and tools used in the railway domain as well as their most relevant functionalities and features.

The recent 2020 survey on formal methods [56], "an unprecedented effort to gather the collective knowledge of the formal methods community", reviews the responses of 130 high-profile experts in formal methods to 30 questions on the past, present and future of formal methods in research, industry and education. The paper also presents 111 position statements by these experts about the challenges and benefits of formal methods. In parallel, [58], "the largest cross-sectional survey of formal methods use among software engineering researchers and practitioners to this date", surveys the academic and industrial use of formal methods in safety-critical software domains, identifying transportation as a typical application domain for formal methods.

Contribution. The evidence from previous reviews shows the interest of the research community in formal methods for railways, as well as several notable attempts of surveying literature and practitioners in the field. Nevertheless, the vast majority of the studies are either non-systematic, based on personal opinions, or collecting information from stakeholders or experts rather than from scientific literature. To our knowledge, this is the first systematic mapping study on the topic. We contribute to the body of knowledge with an empirically grounded overview of the current state of the research on formal methods in railways, to help researchers and practitioners identify the current gaps, and embrace future challenges.

3 REVIEW METHOD

This section describes the review method adopted, which follows the guidelines of Kitchenham [71] for conducting literature reviews. Accordingly, we first outline research questions (Sect. 3.1), and then we illustrate the search string (Sect. 3.2), the study search and selection strategy (Sect. 3.3), followed by data extraction (Sect. 3.4) and data synthesis procedures (Sect. 3.5).

3.1 Review Questions

The main goal of our SLR is as follows:

Goal: *Identify, understand and characterize studies on the application of formal methods to the development of railway systems, identifying recent trends and considering industrial applications, for the purpose of supporting formal methods practice and research.*

To address this goal, we aim to first provide a demographic characterization of studies concerning applications of formal methods in the railway domain. Then, we plan to classify which methods are used, in which phase of the development process, and for what types of systems. Within this classification, we want to identify which papers are concerned with industrial applications, either in the context of exploratory studies involving industrial partners, or for the development of real-world railway products. Finally, we want to focus on the trends of the last years to understand possible future directions. By 'recent' we intend studies published after 2015, as in our research we identified a relevant increase in the number of studies starting from 2016 (cf. Fig. 3). Therefore, from our goal, we derive the following research questions.

- RQ1: How is research demographically and empirically characterized in the field of applications of formal methods in the railway domain?
  - RQ1.1: What is the **time** distribution of primary studies?
  - RQ1.2: What is the **venue** distribution of primary studies?
  - RQ1.3: Which type of **evaluation** has been conducted in the primary studies?
  - RQ1-I: What is the degree of **industrial** involvement in the primary studies?
- RQ2: What formal methods are used in the railway domain?
  - RQ2.1: What is the degree of **formality** of the studies?
  - RQ2.2: What **formal techniques** are used?
  - RQ2.3: Which specification **languages**?
  - RQ2.4: Which **tools**?
- RQ3: In which way are formal methods applied to railway system development?
  - RQ3.1: To which category of **railway system**?
  - RQ3.2: To which category of **railway subsystem**?
  - RQ3.3: In which **phases** of the system development are formal methods applied?
- RQ-I: What are the characteristics of the studies reporting **industrial** applications?
- RQ-T: What are the emerging **trends** of the last years?

RQ1 aims to give a first overview of the time and venue distribution of the studies, to help identifying the evolution of the field across time, relevant journals and conferences, and the empirical maturity of the studies. RQ1 also includes a "service" question, namely RQ1-I, which serves to support the stratified analysis in relation to RQ-I. RQ2 focuses on the formal methods facet, identifying the core elements of any formal method, namely degree of formality, technique, language and tool. RQ3, instead, focuses on the railway facet, and aims to identify the most common phases, railway systems and subsystems in which formal methods are applied.

RQ-I and RQ-T, instead, aim to address *recent trends and industrial applications*, as specified in our overarching research goal. While RQ1 to RQ3 are independent questions, RQ-I and RQ-T are cross-cutting questions (e.g., certain methods identified in RQ2 may be more trendy, other more established and industrially validated). The paper is organized to primarily answer RQ1, RQ2, and RQ3 while RQ-T and RQ-I are answered in relation to the other questions to facilitate the interpretation of the data and have a more concise visualization of the statistics.

3.2 Databases and Search String

We selected the following scientific databases as data sources, which typically include papers in our considered scope: ACM Digital Library, IEEE Xplore, ScienceDirect and SpringerLink. For SpringerLink, we focus the search on the categories of "Computer Science > Software Engineering" and "Engineering > Software Engineering/Programming and Operating Systems", as a pilot search on the entire database—which supports full-text search only— conducted to 49,116 documents, a number that was considered unmanageable for the available resources. To define the search string, we use the major terms "formal methods" (representing the object of the research, or intervention) and "railways" (representing the domain of application, or context) as base terms. Then, we elaborate each base term with alternative words, keyphrases and wildcards, when appropriate. We then use the Boolean OR to incorporate alternatives into each base term set, and Boolean AND to link the two sets. The terms were initially selected based on brainstorming among the

Inclusion Criteria	
I1	The study presents an application of formal or semi-formal method, including model-based development methods, to the development of railway systems
I2	The study is mainly concerned with the development of railway systems for signaling and control
I3	The study comes from an acceptable source such as a peer-reviewed scientific journal, conference, symposium, or workshop
I4	The study is written in English language
Exclusion Criteria	
E1	The study does not use a formal or semi-formal method
E2	The study does not apply a method to the railway domain
E3	The study uses a railway problem as a part of a benchmark for performance evaluation
E4	The study is concerned with the quantitative assessment of reliability, availability and maintainability (RAM) requirements expressed in quantitative form
E5	The type of study is a secondary study
E6	The type of study is a book or a book section
E7	The study did not undergo a peer-review process (i.e. Festschrift contributions, etc.)
E8	The study has been published in another, extended form
E9	The study has the form of editorial, abstract, keynote, poster or a short paper (body less than or equal to 4 pages)

Table 1. Inclusion and exclusion criteria.

1	Is there a clear statement of the aims of the research?
2	Is there an adequate description of the context in which the research was carried out?
3	Is there a clear description of the adopted research methodology?
4	Is there a clear description of the task addressed with formal methods?
5	Is there a clear identification of the formal languages and tools used?
6	Was the data collected in a way that addressed the research issue?
7	Was the data analysis sufficiently rigorous?
8	Is there a clear statement of the findings and limitations?
9	Is the study of value for research?
10	Is the study of value for practice?

Table 2. Quality checklist adopted for the study.

participants and the search string was further refined through pilot searches. The final search string:

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

3.3 Search Strategy and Study Selection Procedures

Our **primary search** strategy consists of adapting the search string to each specific database, and then selecting relevant studies for data extraction. The selection is performed considering inclusion and exclusion criteria listed in Table 1 and the quality checklist listed in Table 2. The quality checklist is inspired by the work of Dybå and Dingsøyr [41] and Chen and Ali Babar [27], and adapted to our context during initial pilots. The selection procedure is carried out according to the following instructions:

(1) **Retrieval:** adapt the search to each specific search engine, considering its peculiarities. Perform the search on metadata only. If this is not supported, perform full-text search.
(2) **Screening:** read title and abstract of the papers and apply the inclusion criteria. The papers that fulfill the criteria are marked as *included*, they are downloaded and stored in Zenodo. Use the features of Zenodo to identify and discard duplicates.
(3) **Full-text Reading:** read the full-text of the included papers, and apply the exclusion criteria, plus the quality checklist. A ternary scale (Yes = 1, Partial = 0.5, No = 0) is used to grade the studies on each question reported in the checklist. The quality score of the paper is computed as the sum of the grades. If a paper does not not reach a quality score higher than 6 out of 10, exclude the paper from the selection. This threshold was identified during a pilot study to discard papers that would not allow appropriate data extraction, as relevant information is missing. The selected papers are given a unique identifier and are retained for data extraction.

The **secondary search** strategy consists of identifying additional studies by performing backward snowballing on high-quality papers, namely all those papers that received a quality score equal to 10—all quality criteria fulfilled.

Two researchers—first and second author—apply the study selection procedures outlined above on separate subsets of the retrieved studies. When a study includes the researcher among the authors, the selection is performed by the other researcher to reduce bias. The two researchers have backgrounds in formal methods and railways, but complementary competences. The first author has a more industrial background, having worked as system engineer in a railway company, and participated to several technology transfer studies. The second author has a strong academic background on different formal methods, formal languages and logic. Thus, possible deficiencies in terms of knowledge of formal methods of the first author are compensated by the second one, while knowledge gaps on railway-specific aspects by the second author are addressed by the first one. These considerations apply also to the data extraction and synthesis procedures (Sect. 3.4).

3.3.1 Performing Primary Search and Study Selection. The primary search is repeated four times on the following dates: October 30, 2017 (Pilot); December 7, 2017 (First, I); November 26, 2018 (Second, II); September 24, 2020 (Third, III).

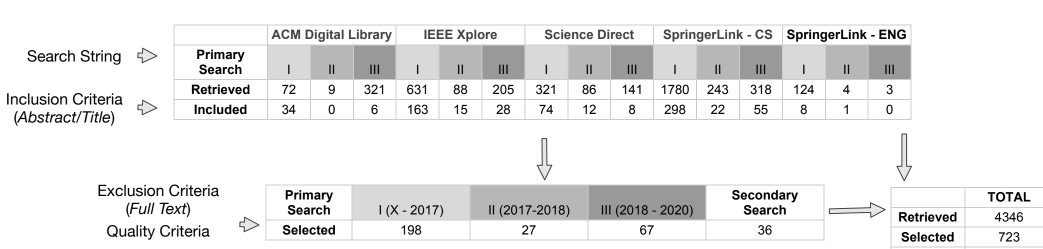

Fig. 1. Process of study selection and numerical results.

The search of October 2017 was conducted as part of a pilot study, in which also the data extraction criteria were piloted on a sample of the top-10 papers retrieved by the four search engines (40 papers in total). The pilot study allowed the involved assessors to align their judgments in data selection and extraction, and consolidate the procedures. Since the search, selection and analysis were focused on a subset of the papers, we do not report these results here.

Search I was unbounded, so all papers available before December 7, 2017 were considered. The other searches were restricted to the time interval between the last year of the preceding search, and the year of the search[4]. Therefore, Search II collected papers in the interval Jan 1, 2017—Nov 26, 2018, and Search III in Jan 1, 2018—Sep 24, 2020.

The secondary search was conducted based on the papers selected in the primary search. Fig. 1 reports the numerical results of the study selection procedure. Overall, 4,346 studies were initially retrieved, 724 were included, and **328** were finally selected.

Empirical Evaluation Information (RQ1.3)	
Type of Study and Evaluation (from Chen and Ali Babar [27])	Rigorous analysis (RA): Rigorous derivation and proof, suited for formal model
	Case study (CS): An empirical inquiry that investigates a contemporary phenomenon within its real-life context; when the boundaries between phenomenon and context are not clearly evident; and in which multiple sources of evidence are used
	Discussion (DC): Provided some qualitative, textual, opinion
	Example (EX): Authors describing an application and provide an example to assist in the description, but the example is used to 'validate' or 'evaluate' as far as the authors suggest
	Experience Report (ER): The result has been used on real examples, but not in the form of case studies or controlled experiments, the evidence of its use is collected informally or formally
	Field study (FS): Controlled experiment performed in industry settings
	Laboratory experiment with human subjects (LH): Identification of precise relationships between variables in a designed controlled environment using human subjects and quantitative techniques
	Laboratory experiment with software subjects (LS): A laboratory experiment to compare the performance of newly proposed system with other existing systems
	Simulation (SI): Execution of a system with artificial data, using a model of the real world

Table 3. Data extraction categories for evaluation information, which are used to answer RQ1.3. The types of study highlighted in bold are the ones that have actually been found in the selected papers.

3.4 Data Extraction Procedure

Data extraction is performed by the authors, referred in the following as *extractors*. Relevant publications are partitioned into two balanced sets, and each extractor extracts data from one set. Each extractor reviews also the extracted data for the other set. In case of disagreement, a third expert in railway applications to railway problems, namely Alessandro Fantechi from the

[4]This choice prevented from ignoring papers at the boundary between years. Duplicates were discarded with the support of Zenodo (cf. the study selection procedure in Sect. 3.3)

Industrial Study Information (RQ1-I)	
Industrial Evaluation (adapted from Chen and Ali Babar [27])	NO: not evaluated in industrial settings
	LAB: industrial problem treated in laboratory settings
	IND: industrial problem validated with railway experts
	DEV: development of an industrial product
Authorship	A: only academic authors
	I: only industrial authors
	AI: both academic and industrial authors

Table 4. Data extraction categories used to identify industrial studies, in relation to RQ1-I.

Formal Methods Facet (RQ2)	
Degree of formality of the method(s)	F: Formal
	SF: Semi-formal
	SFF: Semi-formal and Formal
Name(s) of the technique(s) applied	List of techniques applied in the paper (e.g., model-based development, model checking, theorem proving)
Name(s) of the language(s) used	Name of the languages used for modeling in the context of the paper (e.g., UML, State Machines)
Name(s) of the support tool(s) used	Name of the tools used in the paper (e.g., Atelier B, SCADE, Simulink, UPPAAL)

Table 5. Data extraction categories related to the formal methods facet, used to answer RQ2.

University of Florence[5], is consulted to guide towards a decision on the data to be recorded. When one of the extractor is author of a relevant study, the data extraction is conducted by the other. The data is recorded in the form of shared Google spreadsheets to ease analysis and cross-checking.

Data extraction is conducted by first extracting publication details—title, authors, type of venue (conference or journal), name of venue, publication year and doi. This information is used to answer RQ1.1 and RQ1.2. In addition, the information about the year of publication serves also RQ-T. To answer RQ.1.3, we extract evaluation information following the categorisation proposed by Chen and Ali Babar [27], reported in Table 3. Furthermore, to answer RQ1-I we extract the information from Table 4, also adapted from Chen and Ali Babar [27], and enriched with information about authorship. This information serves also RQ-I. Concerning the subquestions related to RQ2, about the formal method facet, we extract the data reported in the extraction scheme of Table 5, while for RQ3 the scheme in Table 6 is applied. The extraction schemes reported in the tables include two types of data: those for which pre-defined classes are considered (reported in **bold** in the tables); those for which the extractors can use free-text (reported in ***bold italic***). The free text is homogenized in the data synthesis procedure. The extraction schemes also allow the possibility to include more than one element in each extraction item, e.g., more than one language or more than one phase.

3.5 Data Synthesis Procedure

In the data synthesis procedure we consider the extracted data, we homogenize them and provide visual analytics to systematically answer the RQs. Part of the data (e.g., all evaluation information, as well as the degree of formality of the method, phases, type of study) are well-defined sets

[5]The author is the most cited in Scopus when searching for publications in "formal method" and "railway".

Railway Context (RQ3)	
Phase(s) of the system development addressed (from CENELEC)	P: Planning
	R: Requirements
	A: Architecture & Design
	D: Component Design
	I: Implementation
	T: Testing
	N: Integration
	V: Validation
	M: Maintenance
Type(s) of railway system considered	SA (Stand-alone system): if the system treated in the paper is not related with other systems, as for the case of Platform Screen Door Controllers, Railway Crossing Controllers, Axle Counters
	ERTMS-ETCS: if the system is part of an ERTMS/ETCS system
	CBTC: if the system is part of a CBTC system
	CTCS: if the system is part of a CTCS system
	NS (Non-standard Train Control and Management): if the system is part of a train control and management system that does not follow an international standard
Type(s) of railway subsystem considered	For Stand-alone systems (SA): name of the system (e.g., Platform Screen Door, Railway Crossing, Axle Counter, etc.)
	For ERTMS-ETCS, CBTC, CTCS, NS: • HLCL (High-level Control Logic and Communication): if the paper treats the high-level logic of the system, or the communication between two or more components • Name of the system (e.g., wayside ATP, onboard ATP, ATO, Radio Block Center, etc.): if the paper treats solely one subsystem

Table 6. Data extraction categories in relation to the railway context, used to answers RQ3.

of classes. For these cases, data synthesis is straightforward, and results are represented in the form of graphical diagrams, choosing the most appropriate for each case to ease visualization and analysis. Other data (e.g., name of tools, languages, techniques) are based on free-text entered by the extractors. This data needs to be homogenized, and, to this end, we adopt an open coding technique [88]. One author codes the free text, and produces a set of well-defined and finite set of tags that can be used to produce appropriate statistics and data visualization. The other author reviews and cross-checks the coding results also in relation to understandability and clarity of the tags. In this task, the first author primarily homogenizes data in relation to RQ3, given his greater expertise in railway systems. The second author, instead, primarily homogenizes data in relation to RQ2, given his broader knowledge of formal methods. This process is carried out throughout multiple iteration until an agreed set of classes is reached also for the free-text fields.

In general, we visually synthesize results by strictly following the categories identified, and by providing histograms that account for industrial studies and for recent trends. In some cases we considered it appropriate to provide refined data synthesis, guided by evidence from the extracted data. Specifically, the studies were observed to use multiple formal techniques, and thus we synthesize data also about combinations of techniques. Similarly, for railway development phases, we highlight combinations of multiple phases addressed by the same study. Concerning tools, we provide combined statistics with specification languages, to highlight relevant relationships, as we observed that often the adopted specification language does not match with the used tool. Finally, relationships between categories of systems and subsystems are also highlighted.

The final spreadsheet file, which has been used to produce the statistics in this paper, is publicly shared at https://doi.org/10.5281/zenodo.5084640.

4 RESULTS

In the following, we report the results of our analysis. Specifically, for each element of interest, we first plot and discuss the total number of studies in relation to industrial ones. Then, we plot the distribution for recent studies (published after 2015) without distinguishing between industrial and academic. We observed that industrial studies are scarce in recent years (cf. Sect. 4.2), thereby limiting the relevance of associated statistics, which will not be reported.

To support the reporting, we first answer the 'service' question RQ1-I, which allows us to perform a stratified analysis in relation to RQ-I. Then follows RQ1.1, to support stratified analysis for RQ-T.

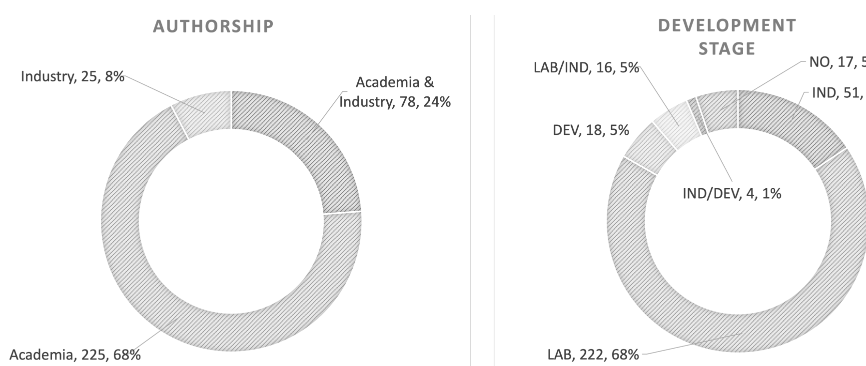

Fig. 2. Degree of industrial involvement, evaluated in terms of authorship of the paper and development stage of the study.

4.1 RQ1-I: Industrial Involvement

Fig. 2 reports the degree of industrial involvement identified in the studies, based on the type of authorship and the development stage of the application considered in the paper.

We see that about two thirds of the studies (222, 68%) have academic authors only, while the other third have some form of industrial involvement, either in conjunction with academic authors (78, 24%) or with authors coming exclusively from industry (25, 8%).

Looking at the development stage we see a similar situation, with the majority of the studies concerned with industrial problems treated in laboratory settings (LAB, 222, 68%). In a non-negligible amount of cases, however, a step forward was performed: part of the studies have been validated with railway experts (IND, 51, 16%), part of them document the development of railway products with formal methods (DEV, 18, 5%) and some studies were considered somewhat in between the different categories (LAB/IND, 16, 5%; IND/DEV, 4, 1%). Still, some studies did not have any form of contact with industry and its specific problems, but considered solely toy problems (NO, 17, 5%).

The statistics just described are used in the following to identify those studies that are concerned with industrial applications (industrial studies, for short) and perform a stratified analysis of the results. Specifically, we consider a study to be *industrial* in two main cases: (i) it is tagged as IND, DEV, or IND/DEV; (ii) it is tagged as LAB/IND and the authorship is AI or I. In this way, we exclude from industrial studies borderline cases for which the actual industrial involvement is not entirely clear (i.e., those marked as LAB/IND, but without industrial authors). According to this classification, 24% of the studies are industrial (79 in total), while 76% are not (249)—in the following we refer to these studies as *academic*.

The results show that studies are in general performed by academics in laboratory settings. Nevertheless, a relevant number of them (about one fourth) involve industry in some form, with formal methods applied also for the development of real products. This proportion suggests that research in the field is not self-referential and interaction with industry is present.

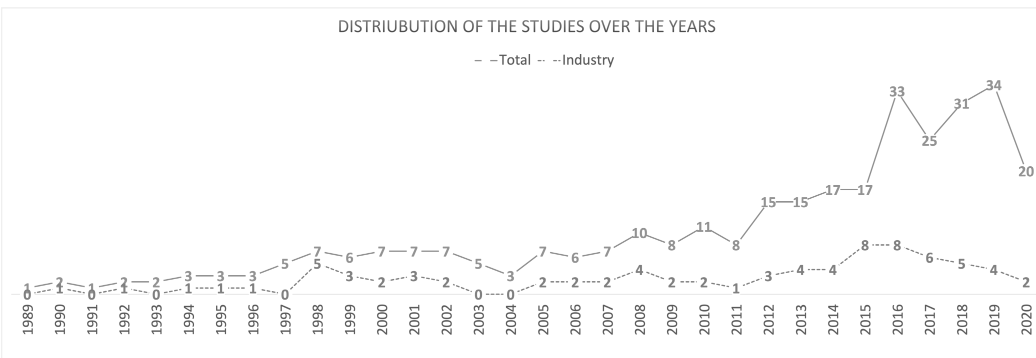

Fig. 3. Studies by year.

4.2 RQ1.1: Studies by Year

Fig. 3 shows the number of publications per year, also considering industrial studies. We see that a slowly but steadily increasing number of works is available starting from 1989. Formal methods for railways therefore span over 30 years of research and applications. The number of papers increases from 2016 onwards, with a peak in 2016 and in 2019. As one can see from Fig. 3, in 2016 publications almost double with respect to the previous two years, going from 17 to 33 papers.

The overall increase since 2016 can be associated to the boost given to research in the railway domain by the Shift2Rail program (https://shift2rail.org/), which started providing grants exactly in 2016. The specific peak in 2016 is linked to the occurrence in the same year of the two conferences RSSRail (Int. Conf. on Reliability, Safety, and Security of Railway Systems: Modeling, Analysis, Verification, and Certification) and ISoLA (Int. Symp. on Leveraging Applications of Formal Methods, Verification and Validation). The former had its first edition in 2016, and it is specialized in railways, therefore being a natural venue for these types of studies. The second one is a bi-annual symposium, and regularly has a track dedicated to formal methods for railway systems. The peak in 2019 is again due RSSRail, and its co-occurrence with FMICS (Int. Conf. on Formal Methods for Industrial Critical Systems), a venue in which railway applications are a common topic.

Concerning industrial studies, it is interesting to note that the first ones appear already in the early '90s. This indicates that formal methods for railways is born with a strong industrial focus, as industrially relevant problems have always been a primary concern. On the other hand, while for several years the proportion between industrial and academic studies is somewhat stable, we see that the radical increase in terms of papers observed in the last years is not followed by a corresponding abundance of industrial studies. On the contrary, industrial studies started decreasing after a peak in 2015–2016. This suggests that recent work mostly focuses on research problems, possibly exploiting new techniques, while industrial experimentation appears to be more limited[6].

Information about the year of publication is used in the following to identify recent trends. Specifically, we consider a study to be *recent* if it is published in the last 5 years, i.e., after 2015. This choice matches with the radical increase in 2016 in terms of number of publications in the field.

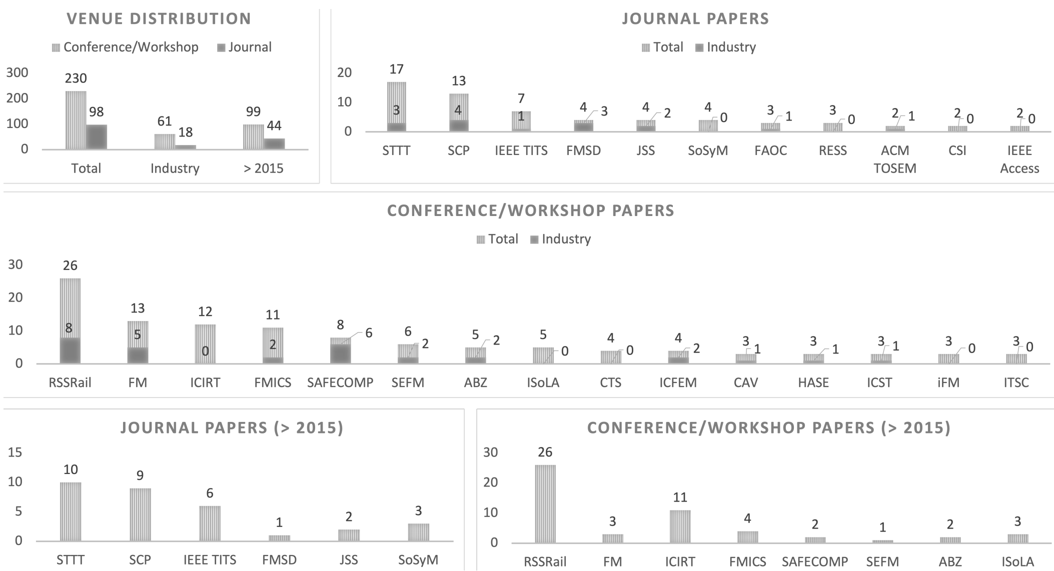

Fig. 4. Statistics on publication venues.

4.3 RQ1.2: Venue

Fig. 4 reports the statistics on the venues in which papers about applications of formal methods to railways are published. The majority of the works are published in conferences (230, 70%), but a relevant percentage appears in journals (98, 30%). The distribution is the same for recent works. This indicates a well-established research field, with solid journal publications. On the other hand, the field is subject to ongoing development, with many conference and workshop contributions. The proportion leans towards conferences in a more marked way when considering industrial papers (61, 77% vs 18, 23%). This can be linked to the tendency of companies to go for in-person dissemination venues, which can facilitate networking. Furthermore, journal publications may require disclosing more data, which is not always acceptable for company confidentiality policies.

Let us now look at the specific venues—for the sake of space, the plots report solely the most frequent ones. The acronyms of conferences/workshops and journals are described in Table 7.

Among **conference contributions**, RSSRail clearly dominates (26, 11% of the conferences). This is not surprising, as this venue is specialized in rigorous methods applied to railway development. RSSRail is followed by FM (13, 6%), ICIRT (12, 5%) FMICS (11, 5%), SAFECOMP (8, 3%) and SEFM (6, 3%). FM is the flagship conference on formal methods, showing that the railway domain is particularly important for the whole community, and it is not a niche field of experimentation. The other venues are also not strictly focused on railways, but on intelligent transport systems, in the case of

[6]This could indicate that some formal approaches are now consolidated in industry. However, this conclusion cannot be drawn from the analysis, which focuses on research papers. A multi-vocal literature review would be needed to explore this.

Journals	
STTT	International Journal on Software Tools for Technology Transfer
SCP	Science of Computer Programming
IEEE TITS	IEEE Transactions on Intelligent Transportation Systems
FMSD	Formal Methods in System Design
JSS	Journal of Systems and Software
SoSyM	Software and Systems Modeling
FAOC	Formal Aspects of Computing
RESS	Reliability Engineering & System Safety
ACM TOSEM	ACM Transactions on Software Engineering and Methodology
CSI	Computer Standards & Interfaces
IEEE Access	IEEE Access
Conferences and Workshops	
RSSRail	International Conference on Reliability, Safety and Security of Railway Systems
FM	International Symposium on Formal Methods
ICIRT	IEEE International Conference on Intelligent Rail Transportation
FMICS	international Conference on Formal Methods for Industrial Critical Systems
SAFECOMP	International Conference on Computer Safety, Reliability and Security
SEFM	International Conference on Software Engineering and Formal Methods
ABZ	International Conference on Rigorous State Based Methods
ISoLA	International Symposium on Leveraging Applications of Formal Methods, Verification and Validation
CTS	IFAC Symposium on Control in Transportation Systems
ICFEM	International Conference on Formal Engineering Methods
CAV	International Conference on Computer-Aided Verification
HASE	IEEE International Symposium on High Assurance Systems Engineering
ICST	IEEE International Conference on Software Testing, Verification and Validation
iFM	International Conference on integrated Formal Methods
ITSC	IEEE International Conference on Intelligent Transportation Systems

Table 7. Description of acronyms for publication venues.

ICIRT, and on formal methods and software engineering applied to safety-critical systems. This landscape indicates that applications of formal methods to railway systems are considered relevant and well-accepted both in application-centered venues, like RSSRail and ICIRT, and in formal methods ones, like FM, FMICS, SAFECOMP and SEFM. Industrial works are particularly welcome in RSSRail, FM, FMICS and SAFECOMP. FM has an Industry Day forum organized in conjunction with the main symposium, which targets industrial development and use of formal methods. A selection of contributions to the Industry Day is published in the symposium proceedings. Recent works confirm the historical landscape, although in this case specialized venues like RSSRail and ICIRT clearly outrank the others. This suggests that research efforts are now focused on applying existing formal methods, possibly tailoring them to specific railway applications.

Considering **journal papers**, STTT (17, 17% of the journals) and SCP (13, 13%) are the most common venues, followed by IEEE TITS (7, 7%), FMSD (4, 4%), JSS (4, 4%) and SoSym (4, 4%). Special issues dedicated to FM and FMICS were published in STTT, SCP, FMSD and FAOC. Also industrial works are published in all these venues, except for IEEE TITS, and recent works follow the general trend. The considered journals are rather diverse in terms of focus. STTT is concerned with tools for technology transfer and the interplay between technology and industry. It is traditionally focused on formal tools and structured methods in general, and it is thus appropriate for papers that wish to experiment novel formal tools on railway systems. SCP, JSS and SoSym have a broader scope, more oriented to system modeling. IEEE TITS is specialized in transport systems, while FMSD is the only pure formal methods journal among those considered. Overall, this landscape confirms the different interests of the research community towards the railway field, and that not only conferences but also a rather large spectrum of journals welcomes formal methods applied to railways.

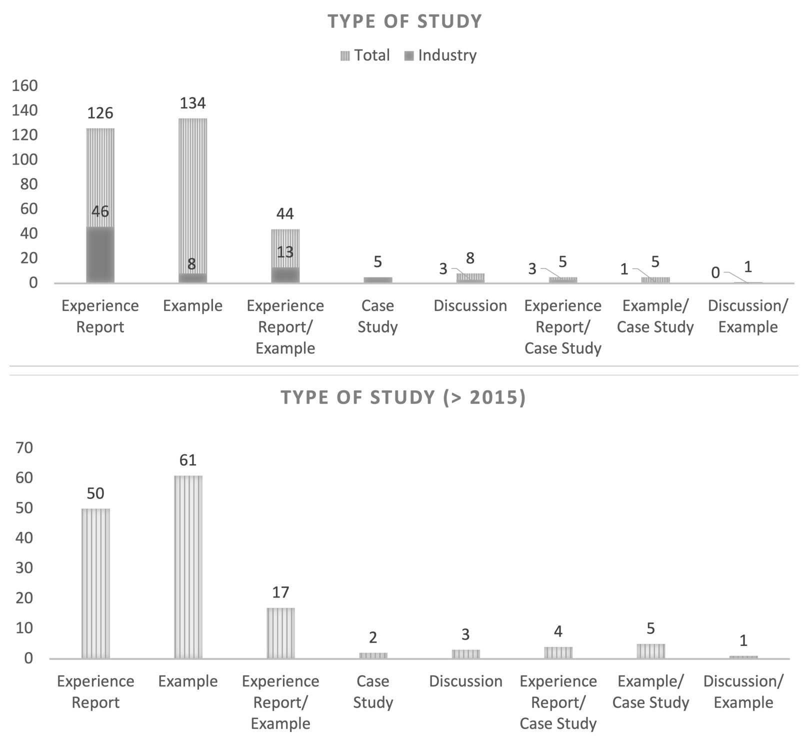

Fig. 5. Type of study.

4.4 RQ1.3: Empirical Evaluation

Fig. 5 reports the statistics concerning the type of evaluation, considering the comparison between industrial studies and the total number of studies (top), and the recent trends (bottom).

The large majority of the studies are Examples (134, 41%), followed by Experience reports (126, 38%) and borderline cases between the two categories (Experience Report/Example, 44, 13%). The remaining papers concern Discussions (8, 2.4%), Case Studies (5, 1.5%) and other cases with less clear-cutting characterization. A different balance is identified for industrial studies, which are mostly Experience Reports (46, 58% of industrial studies) or other borderline cases in the same category (Experience Report/Example, 13, 16%; Experience Report/Case Study, 3, 4%). All five Case Studies (6%) are industrial, as one expects from this type of research [86]. These numbers indicate that most of the academic studies present Examples, to demonstrate or illustrate some formal technique. Instead, industrial studies tend to make a step forward and present real experiences. However, these experiences are mostly retrospective (i.e., Experience Reports) and do not concern the more mature form of Case Studies, with structured research questions and a rigorous process of data collection and analysis.

The trends after 2015 (Fig. 5, bottom) match with the structure already observed for the whole set of studies. Hence, we argue that the focus on Examples and Experience Reports did not substantially change along the years.

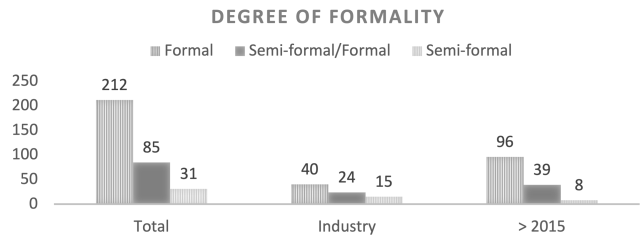

Fig. 6. Degree of formality.

4.5 RQ2.1: Degree of Formality

Fig. 6 reports the degree of formality of the techniques applied in the studies. Most of the works (212, 65%) are strictly Formal, part of them combine Formal and Semi-formal approaches (85, 26%) and the remaining ones are purely Semi-formal (31, 9%).

Interestingly, the proportion changes when considering solely industrial studies, for which half of the studies use exclusively Formal approaches (40, 51%), while the other half make use of Semi-formal techniques (Semi-formal/Formal, 24, 30%; Semi-formal, 15, 19%). This suggests that industrial works tend to take into higher consideration Semi-formal approaches, arguably since these can help to bridge the gap between researchers and practitioners.

Recent works, instead, basically follow the general trend, with 96 Formal (67%), 39 Semi-formal/Formal (27%) and 8 Semi-formal (6%) studies.

To summarize, Formal approaches dominate, with Semi-formal ones having a higher role in industrial studies. The trend did not substantially change in recent studies.

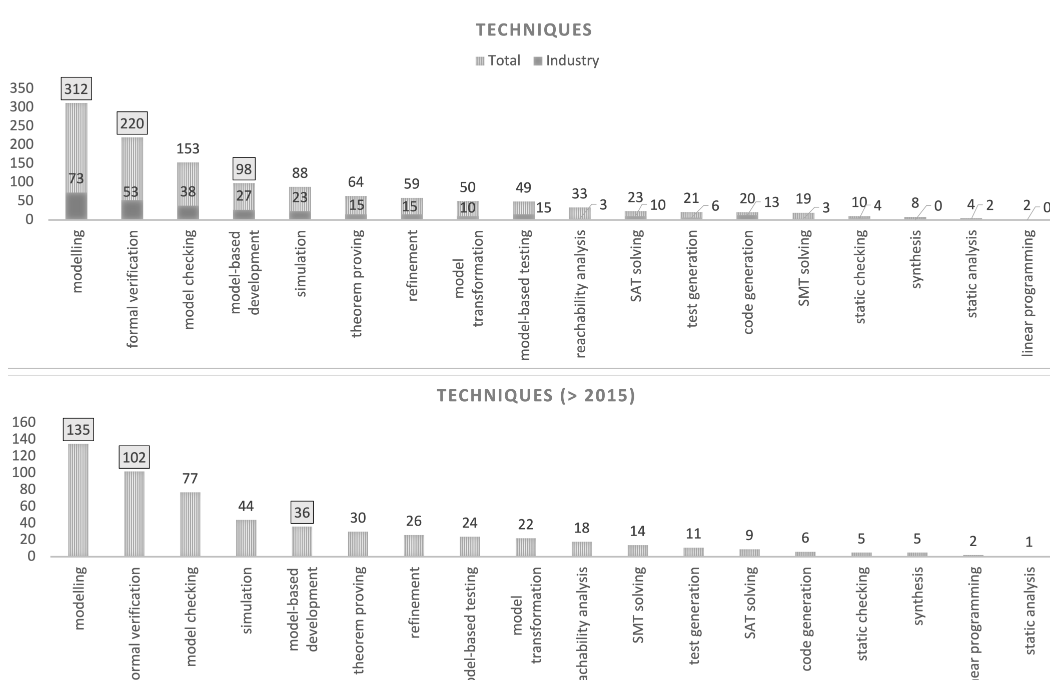

Fig. 7. Techniques.

4.6 RQ2.2: Techniques

Fig. 7 reports the techniques used in the studies, according to the thematic analysis carried out. We report general families of techniques, namely modeling, formal verification and model-based development with highlighted labels.

The vast majority of papers use the two fundamental techniques of formal methods, namely modeling (312, 95%) and formal verification (220, 67%). Though dominant, formal verification is not used in 33% of the studies, suggesting that other approaches, possibly non-formal, are used in combination with modeling. The most common technique for formal verification is model checking (153, 47%), used in about half of the works. The other classical verification techniques, namely theorem proving (64, 19.5%) and refinement (59, 18%) appear in a relevant, yet more limited number of studies. More frequent are other techniques such as the general family of model-based development (98, 30%) and simulation (88, 27%). The presence of other typical model-based techniques is also quite relevant, with model transformation (50, 15%), model-based testing (49, 15%) and reachability analysis (33, 10%) being frequently used.

Techniques that are strictly related to code, like test generation (21, 6%), code generation (20, 6%) and static analysis (4, 1%) appear in a more limited number of papers. On the one hand, this variety of techniques indicates that railways is a playground for a large number of different approaches. On the other hand, this suggests that formal methods are typically applied on abstract, high-level models, and source code is only marginally considered. Industrial studies seem to follow the same general trends, but with more attention to source code, as code generation and static analysis are used in over half of the studies (13 out of 20 for code generation; and 2 out of 4 for static analysis).

Recent studies, shown at the bottom of Fig. 7, indicate that the landscape is basically stable. However, some *hot* techniques exist. In particular for simulation (44, 31% of recent studies), SMT solving (14, 10%), model-based testing (24, 18%) and test generation (11, 8%), half of the studies were published in the last five years.

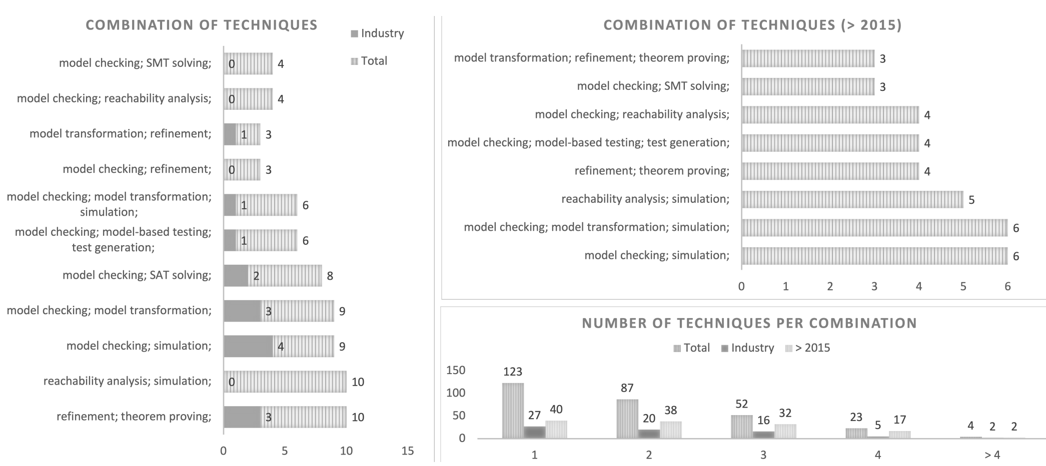

Fig. 8. Combination of techniques.

Fig. 8 shows the most frequent **combinations** of two or more techniques, and without considering the general families of modeling, formal verification and model-based development. Each combination is considered individually—subsets of combinations are not counted.

The bottom-right histogram indicates the number of techniques for each combination. We see that a large majority of the papers use only one technique (123 papers in total, 38%, 27 industrial, 34%, 40 recent, 28%)[7], but a relevant number of papers use two to three techniques. The trend is similar in both industrial and recent papers, though recent papers appear to use also richer combinations (e.g., 12% of the recent papers use 4 combinations, with respect to 7% of the total set).

We now consider the specific combinations of techniques, by looking at the left and top-right histograms of Fig. 8. Here, we consider only combinations occurring in four or more papers (three or more for recent papers), to ease readability. Historically, the most frequent combination of techniques are *refinement & theorem proving*, *reachability analysis & simulation*, followed by model checking with other techniques. These other techniques include model transformation, simulation, SAT solving, model-based testing and test generation. In industrial papers, model checking occurs more frequently, in combination with other techniques in the model-based family. Interestingly, *reachability analysis & simulation*, a combination typically associated with Petri Nets, is never used in industrial papers, although it is the second one in terms of frequency. The greater relevance of model checking with respect to theorem proving is also visible in recent papers, in which *model checking & simulation*, and *model checking & model transformation & simulation* are the most frequent combinations. Traditionally a combination of techniques matching the B-family (ProB in particular), we contribute this also to the recent popularity of applying statistical model checking (UPPAAL in particular) in the railway domain (cf. Sect. 4.8).

It is also worth noting that, out of 166 total papers that use two or more techniques, only 72 are represented in the plot (43%). This indicates that a large majority of the papers use uncommon combinations and that a long tail of variants exists in these plots, as we will observe also for language families and tools in the next sections.

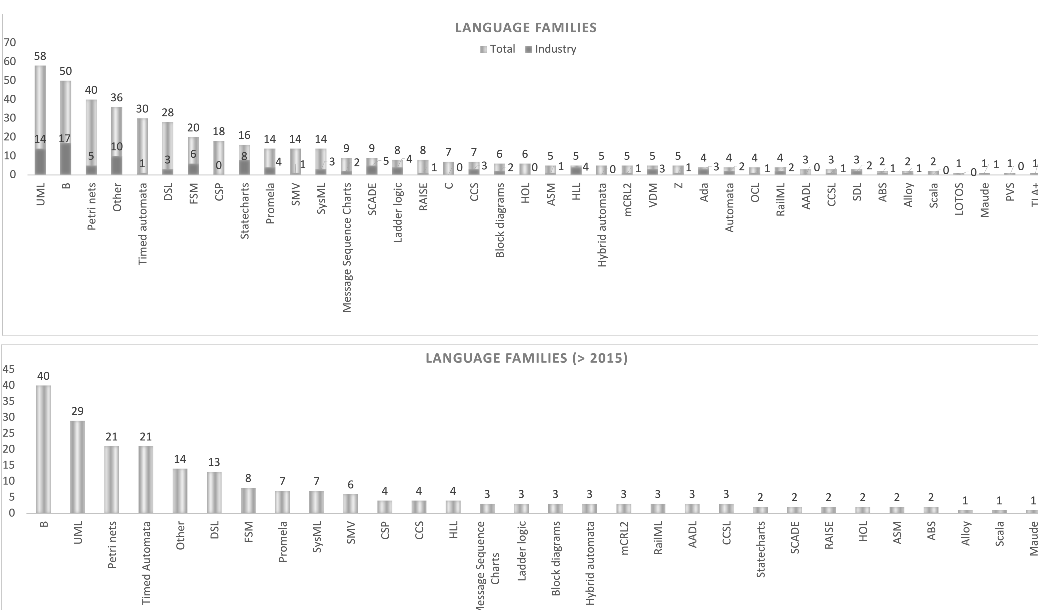

Fig. 9. Modeling language families considered in the studies.

[7]The sum of papers does not amount to the total number of papers, as some of them used only modeling or only formal verification, without reference to known techniques.

4.7 RQ2.3: Language Families

Fig. 9 reports the statistics on the language families used in the studies. Dominant modeling language families are UML (58, 18%), B (50, 15%), Petri nets (40, 12%), Timed automata (30, 9%), Domain-Specific Languages (DSL, 28, 9%), Finite State Machines (20, 6%), CSP (18, 5%) and Statecharts (16, 5%). Other studies use tool-specific languages, such as Promela (14, 4%) and SMV (14, 4%). Besides these well-known language families, the plot shows a large number of languages that are used only in a limited number of studies—yet in many cases above 4. Furthermore, the placeholder 'Other', used for less established languages, appears as fourth most frequent language family, confirming that many works tend to be somehow unique, in terms of the language used.

When looking at the number of industrial studies, the differences with respect to the general trend is rather evident. B appears to be the most frequent modeling language family used in industrial works (17, 22%), followed by UML (14, 18%) and by Statecharts (8, 10%). Some languages appear to be used almost exclusively in academic works. These include Petri nets, Timed automata and CSP. Others, instead, have a more industrial vocation, such as SCADE and High-level Language (HLL), the input language of the SAT-based model checker S3 (Systerel Smart Solver).

Recent works also differ from the historical trends, with the B language clearly occurring as the dominant one (40, 20%) and some modeling languages falling in the long tail, including industrially relevant languages like Statecharts (2, 1%).

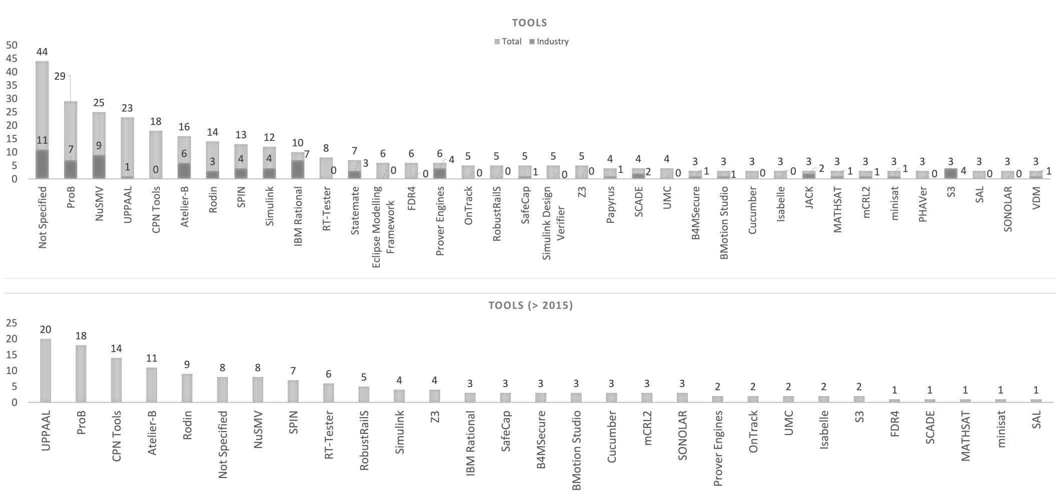

Fig. 10. Tools.

4.8 RQ2.4: Tools

Fig. 10 shows the results about the tools that are used in the papers. The majority of works do not indicate a specific tool (Not Specified, 44, 13% of the total). Frequently used tools are ProB (29, 9%), NuSMV (25, 8%), UPPAAL (23, 7%), CPN Tools (18, 5%), Atelier-B (16, 5%), Rodin (14, 4%), SPIN (13, 4%), Simulink (12, 4%), the IBM Rational family for UML and SysML (10, 3%), *etc.* Tools in the B-family, namely ProB, Atelier-B and the Rodin platform clearly dominate, when considered together, but many other other well-known platforms are considered and a long tail of other tools, used solely in a few papers, can clearly be observed in the plot.

Concerning industrial applications, we see that about the same percentage of papers does not specify a tool (11, 14%). The B-family still dominates, although NuSMV is the most frequently used tool in industrial works (9, 11%). Other tools in the long tail appear to have an industrial vocation, since the papers using them concern works with industry in more than half of the cases. These include Prover Engines, S3 and IBM Rational. Not surprisingly, these are closed-source tools that are not freely available, and experimentation in academia is naturally more oriented towards tools that have a free license and are extensible. Other tools seem to be used almost exclusively in academic works in railways, namely CPN Tools (0 industrial works out of 18) and UPPAAL (1 out of 23).

Recent works show a reduced tendency to have Not Specified tools. This suggests a greater attention in recent years to give importance to the tool used, and not only to the applied technique. The dominant toolset is UPPAAL (20, 14%), which includes UPPAAL SMC and UPPAAL Stratego, followed by ProB (18, 13%). Frequently used are also CPN Tools (14, 10%), Rodin (11, 8%), Atelier-B (9, 6%), NuSMV (8, 6%) and SPIN (7, 5%). This scenario indicates that UPPAAL is the hot tool in recent years, although its usage in the railway industry is still limited, while ProB combines industrial uptake and frequency of use in recent works. The long tail of tools remains also for recent works, suggesting that the field is still a playground for experimentation with tools. Interestingly, many of these tools are specialized for railways, in particular RobustRailS, SafeCap and OnTrack. This suggests that while general purpose formal tools are used in the domain, there is a strong interest to tailor formal tools to the peculiarities of the domain.

It is useful to look at the **relationships** between frequently used tools and modeling languages, reported in Fig. 11. Besides the expected relationships between languages and tools, such as Promela for SPIN, SMV for NuSMV, Timed Automata for UPPAAL and B for ProB, Atelier-B and Rodin, there are some peculiar cases. In particular, the UML language is used in combination with all main tools, including ProB and NuSMV. Interestingly, with the exception of IBM Rational, none of the tools is specifically oriented to support UML. Thus, we conclude that UML is the language commonly used to model the system, but then the model is translated into the input language of different formal tools, e.g., to apply formal verification. This is in line with the fact that UML is the de facto industrial standard for documentation and communication among stakeholders. Another peculiar case is Petri nets, for which rather frequently the authors do not specify the support tool used.

Finally, looking at Fig. 11, it is also worth noting that some tools are used in papers in which different languages are used in combination. In particular, for many tools (e.g., ProB, SPIN, NuSMV) the sum of papers using them is smaller than the sum of the values that appear in their row cells. This phenomenon is less prominent for languages, where the total of papers is generally lower than the sum of the column cells. This suggests that a typical paper in our scope considers a single formal tool, but multiple modeling languages. While for ProB this is somehow in line with the vocation of the tool, which is oriented to be open to different input formalisms, for SPIN and NuSMV this can be related to the vocation of the tools as verification engines rather than design platforms, with limited graphical interfaces [52], yet powerful formal verification capabilities.

4.9 RQ3.1: Category of Railway System

Fig. 12 reports the distribution of system categories. The large majority of papers does not refer to any railway product standard (229, 70%), indicating that most of the works focus on applications that either follow proprietary system specifications from some companies or are examples possibly inspired by real applications. A non-negligible number of works, however, is dedicated to the ERTMS-ETCS standard (61, 19%). This is followed by the CBTC (26, 8%) and the CTCS (15, 5%) standards. Industrial applications follow the same trends, with slightly more applications using ERTMS-ETCS (17, 22% vs 19%) and CBTC (10, 13% vs 8%). Papers based on CTCS, instead, are not concerned with industrial applications. Considering recent studies, the percentage of works that focus on standards increases. In particular, although the majority of works is still classified as

	B	UML	Petri Nets	Timed Automata	DSL	FSM	Promela	SysML	SMV	CSP	TOTAL
NotSpecified	5	4	8	3	3	6	1	1	0	0	44
ProB	27	10	2	0	8	0	0	1	0	5	30
NuSMV	0	7	0	0	1	2	1	0	13	1	25
UPPAAL	0	5	0	23	0	0	0	1	0	3	24
CPNTools	0	6	17	0	0	0	1	1	1	1	18
Atelier-B	16	5	3	0	1	0	0	2	0	0	16
Rodin	14	5	0	0	1	0	0	1	0	0	14
SPIN	0	3	1	0	1	3	12	0	1	1	13
Simulink	0	2	0	1	0	0	0	2	0	1	12
IBMRational	0	8	0	0	0	2	1	3	0	0	10
TOTAL	50	58	40	30	28	20	14	14	14	18	0

Fig. 11. Tools in relation to modeling languages

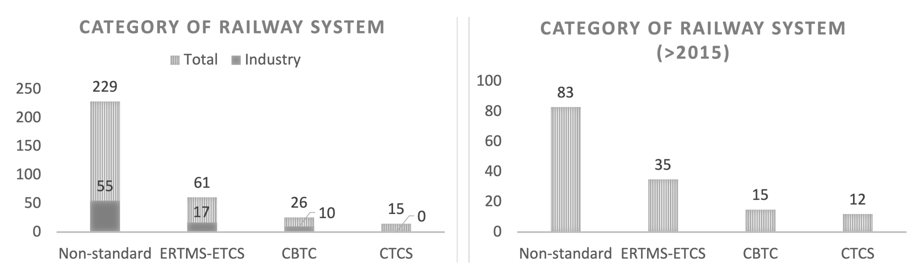

Fig. 12. Category of railway system.

Non-standard (83, 58%), a slight increment is observed on works considering ERTMS-ETCS (35, 24% vs 19%), CBTC (115, 0% vs 8%) and CTCS (12, 8% vs 5%).

4.10 RQ3.2: Category and Railway Subsystem

Fig, 13 reports the categories of subsystems, separated by category. The majority of non-standard subsystems are Interlocking systems (130, 40%), followed by Railway Crossing Controllers (29, 9%) and High-level Control Logic (29, 9%). Then a large set of different subsystems is covered, including ATO, ATP, Configuration Data, etc. This indicates that formal methods have been applied to a wide range of non-standard systems. The dominance of interlocking platforms is strictly linked to their equation-based tabular nature, which make them particularly amenable for formal verification by means of model checking or SMT solving. Interlocking platforms are also strongly present for industrial applications (29, 37%), confirming the prevalence of this type of subsystem in railway studies. High-level Control Logic, instead, is the typical (set of) subsystems considered in studies that use some standard: 42, 13% for ERTMS; 9, 3% for CBTC; 8, 2% for CTCS. Overall, less variety in terms of subsystems types is observed for standardized cases, with the exception of CBTC, for which there is a higher balance in terms of different systems considered as one can visually grasp.

Looking at recent work in Fig. 14, Interlocking still dominates, although in a less marked way (45, 31% vs 40% for Non-standard systems). For the other subsystems the statistics are basically

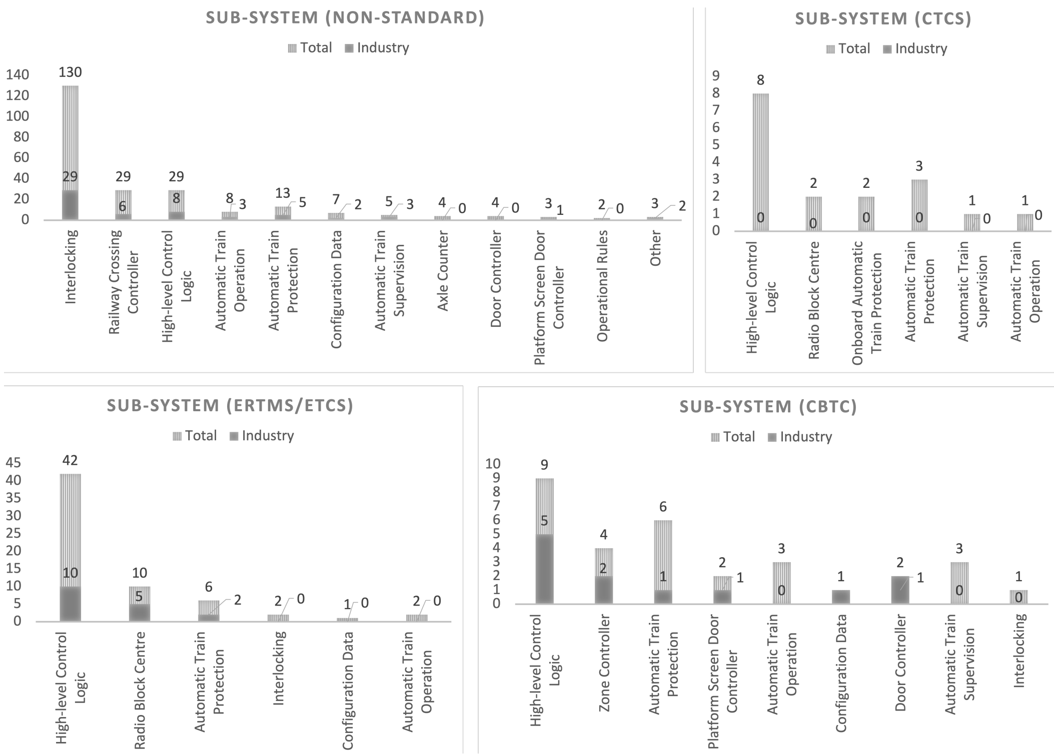

Fig. 13. Category of railway subsystem.

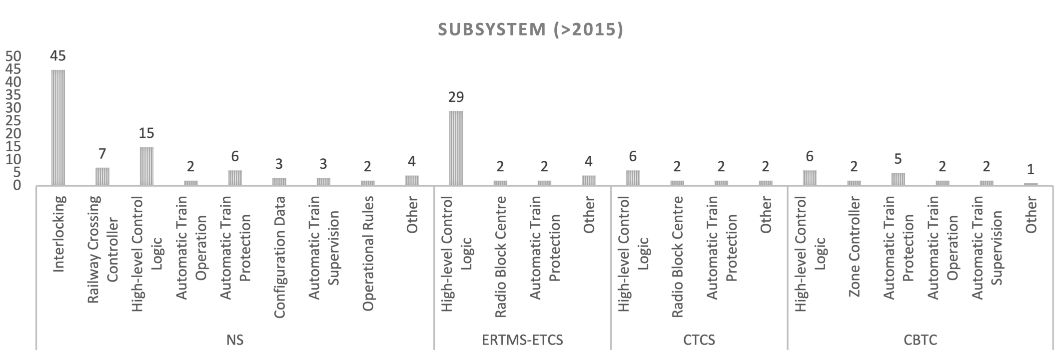

Fig. 14. Category of railway subsystem for recent papers.

comparable with the historical ones, with the exception of Railway Crossing Controller (7, 5% vs 9%). Apparently, the interest in this system, typically used in the past as an exemplary reference problem to experiment with new formal methods, tends to decrease in favor of other subsystems.

4.11 RQ3.3: Railway Development Phases

Fig. 15 reports the development phases considered in the studies. The majority of them is concerned with Architecture (218, 66%) and Detailed Design (149, 45%), followed by Requirements (42, 13%), Testing (41, 13%) and Validation (30, 9%). This trend is followed by industrial studies, although

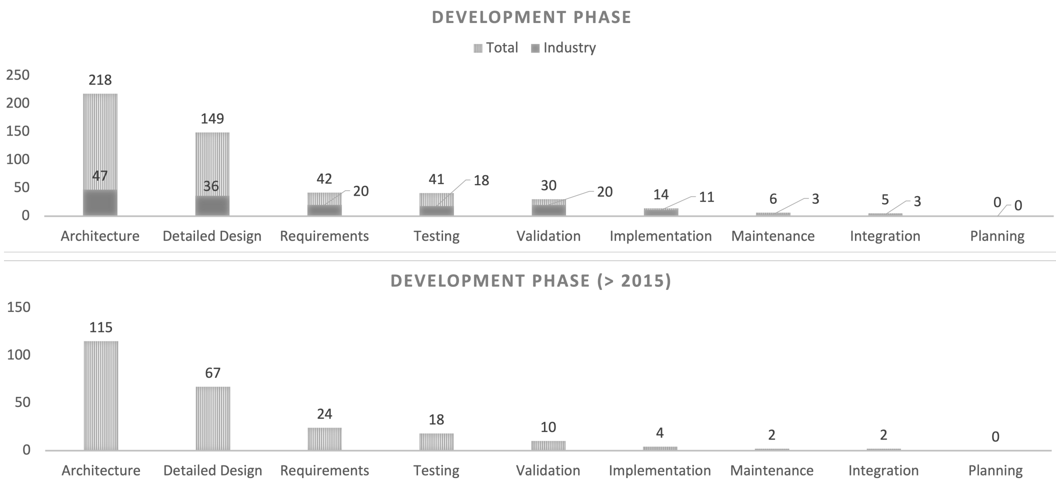

Fig. 15. Railway development phase.

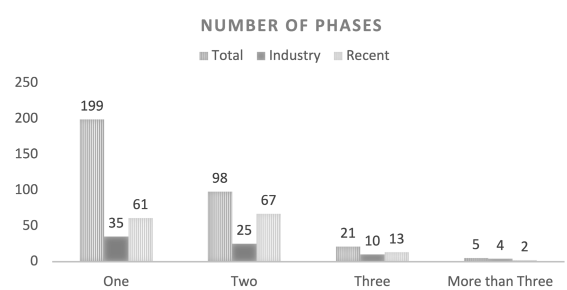

Fig. 16. Number of railway development phases.

these appear to be more focused on later phases of system development and on lower-level system representation. In particular, Validation and Implementation are considered in a relevant number of industrial studies (20 out of 30 for Validation, 11 out of 14 for Implementation). Furthermore, the percentage of studies focused on Architecture is lower for industrial studies with respect to the general trend (47, 59% vs 66%). No notable differences can be observed for recent studies.

Fig. 16 shows the number of phases considered by each study. Most of the studies focus on one phase only (199, 61%), followed by two‘(98, 30%) and three (21, 6%) phases. A limited number of studies considers more than three phases. Industrial works tend to consider a higher number of phases, with 35 (44% vs 61%) focusing on one phase only, and 4 works covering more than three phases (5% vs 2% of the total studies). The majority of recent works consider two phases (67, 47% vs 30%) instead of one, marking a relevant difference with respect to the historical trends.

Fig. 17 reports the most frequent combinations of phases. Although many works focus solely on Architecture (113, 34%) and Detailed Design (69, 21%), several studies consider a combination of Architecture with other phases, namely Detailed Design (40, 12%), Testing (18, 5%) and Requirements (16, 5%). Industrial works that do not strictly focus on Architecture appear to be distributed over different combinations of phases, without a clear dominance. For recent works, Architecture + Detailed Design (33, 23%) directly follows Architecture alone (43, 30%) as typical combination.

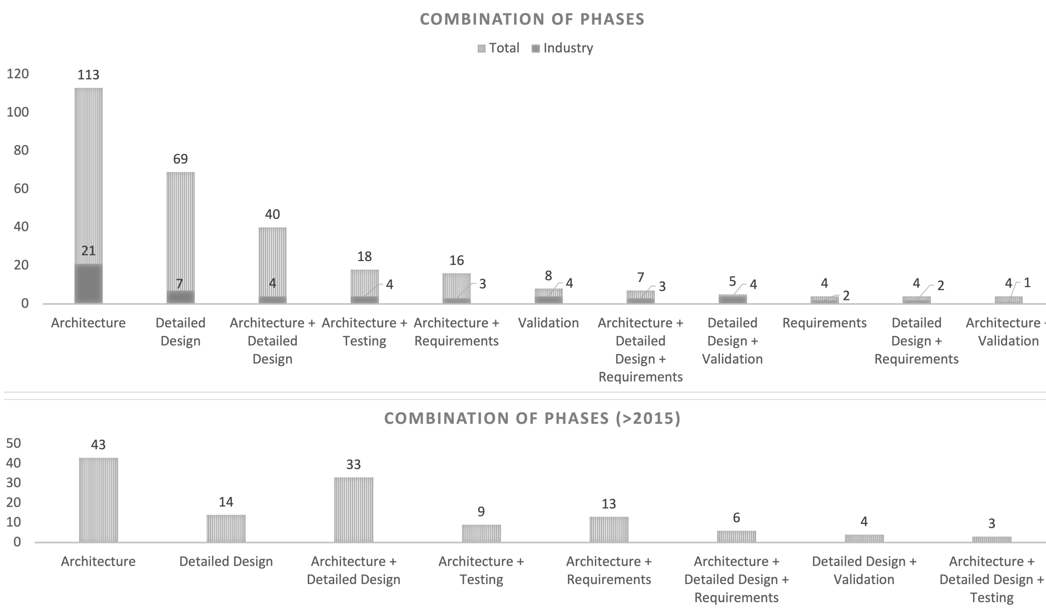

Fig. 17. Combination of railway development phases.

To summarize, the statistics show that formal methods have seen applications in almost all phases of railway system development with more focus on the design phases, namely Architecture and Detailed Design, also in combination. Industrial work tends to give more relevance to later phases, such as Implementation and Validation, and tends to consider a combination of a higher number of phases with respect to academic works.

5 SUMMARY AND DISCUSSION

In the following, we summarize the empirical findings of the study in relation to the main RQs, and, for each question, we discuss implications for research in the field of formal methods for railways.

RQ1. How is research demographically and empirically characterized in the field of applications of formal methods in the railway domain?

Timeline. Studies in formal methods for railways start in the late '80s, with a radical increase since 2016, thanks to the creation of dedicated venues (e.g., RSSRail) and the Shift2Rail program.

Publication venues. 70% of the works is published in conferences and 30% in journals. Conferences are application-centered (RSSRail, ICIRT) as well as formal methods-centered (FM, FMICS, SAFECOMP, SEFM). Dominant journals are STTT and SCP.

Evaluation. The majority of studies are evaluated through Examples (41%) and Experience Reports (38%), while Case Studies are limited (1.5%).

Industrial Involvement. 68% of the studies have academic authors only, 8% have authors coming exclusively from industry and 24% have mixed affiliations. The majority (68%) considers industrial problems in laboratory settings, 16% validate the results with industrial partners and 5% document the development of real railway products with formal methods.

Research in formal methods for railways has a solid tradition and several studies were published in collaboration with industrial partners. This indicates that formal techniques have a strong appeal for industries, and practitioners have interest in applying them to address problems that cannot be solved with other means. The presence of EU funding and dedicated venues clearly supports the development of research in the field. It is therefore advisable for researchers to take advantage of the current positive conjuncture, and make a step forward to better answer industrial demands by increasing the empirical rigor of their research. Despite the potential for sound industrial works, the empirical maturity of the field is still limited. Many works do not follow empirical standards, but simply report Examples or, in the best cases, retrospective Experience Reports. Based on this evidence, we argue that the community should attempt at answering existing questions with empirical software engineering lenses [96]. This way, pressing questions, for which industry demands answers, can be addressed and the field can grow on the basis of scientific evidence. Research questions to address include the ones already discussed in previous work, also from other domains (e.g., aerospace and cybersecurity [77, 78]), and revolve around the applicability of formal methods in real contexts, the maturity of tools [51, 52], their learning curve [55, 90], their connection with the software engineering practice and processes [48, 77, 78] and how independent a company can realistically become from academic formal methods experts, e.g., through the usage of covert, hidden or lightweight formal methods [69]. These issues have been widely discussed in the literature, and appear to have not substantially changed over the years [56].

Given the possibility offered by the strong industrial presence in the field, it is advisable to carry out research in the form of Case Studies, following established guidelines, like those by Runeson *et al.* [86]. Furthermore, not only Case Studies should be pursued, but also Laboratory Experiments, for example to compare software tools and evaluate user-related aspects. Quite surprisingly, our mapping study did not identify any form of controlled experiment. These are particularly common in software engineering [72], especially using students as subjects [37, 43]. A primary role here can be played by the community of formal methods education and training [40]. Specifically, by performing controlled experiments with students, instructors of formal methods can contribute not only to improving teaching practices, but also to the empirical assessment of formal methods. Overall, to advance the empirical maturity of the field, Experience Reports should become Case studies in the future, while Examples—which dominate the current literature, with novel formal approaches evaluated on limited cases—should become more sound Laboratory Experiments.

We believe that carrying out more empirically sound studies may lead to publications outside the formal methods arena in leading software engineering venues, such as the Empirical Software Engineering journal, IEEE Transactions on Software Engineering and the ACM/IEEE Int. Conf. on Software Engineering (ICSE), where publications in formal methods and railways have already been published [28, 52]. This would give broader visibility to the formal methods community itself.

RQ2: What formal methods are used in the railway domain?

Formal vs Semi-formal. Most of the studies are strictly formal (65%), while others use semi-formal methods (9%) or, more frequently, a combination of both (26%).

Techniques. Formal modeling is applied in 95% of the studies and formal verification in 67%. Model checking is the most commonly adopted technique (47%), followed by simulation (27%), theorem proving (19.5%) and refinement (18%). Less commonly used techniques are those strictly related to code, like test generation (6%), code generation (6%) and static analysis (1%). 38% of the papers use only one technique, while the rest uses combinations of two or more. Theorem proving in conjunction with refinement is the most frequent combination.

Languages and Tools. A large variety of modeling language families and tools is used. The dominant languages are UML (18%) and B (15%), while frequently used tools are ProB (9%), NuSMV (8%) and UPPAAL (7%). UML is normally used in combination with different formal tools. A typical paper considers a single formal tool, but multiple modeling languages.

The landscape of techniques, languages and tools is extensive. This confirms the findings of a previous questionnaire with railway stakeholders [9], which highlighted the long tail of over 40 tools, even with only 44 respondents. On the one hand, this indicates that railways can be regarded as an appropriate field for research to experiment with a large variety of techniques, and confirms that this is a domain in which novel approaches can be tested. On the other hand, the fragmentation of techniques, languages and tools does not facilitate the work of practitioners, who face a paradox of choice when deciding what formal methods to adopt, as also observed by Steffen [90]. This is also not facilitated by the need to use combinations of techniques or tools, as done in part of the papers. There is therefore a need for a clearer classification of what techniques, languages and tools can and cannot do to facilitate the choice of practitioners.

Despite this fragmentation, however, some latent patterns emerge, which deserve to be highlighted. UML is normally used for high-level representation, and models are normally translated into the input language of some formal verification engine. Typical choices are ProB, UPPAAL and NuSMV, which cover quite diverse needs [52], e.g., UPPAAL is appropriate when quantitative aspects come into play and when simulation is the best option; NuSMV can be chosen when complete state-space exploration is needed and the problem at hand can easily be represented as a state machine; ProB is recommended when prototyping, when an open platform is needed and also when one aims at top-down development of a monolithic system.

Areas that need more exploration are also present, even in the wide landscape currently mapped. Specifically, research appears to neglect techniques that are closer to code, such as test generation, code generation and static analysis. Though it is widely believed since decades that formal methods and in particular formal verification techniques are at their best in the early design phases [56, 80, 87], it is the testing and debugging of the railway software that is the most resource consuming activity (about 50% of the overall cost [82]) in safety-critical systems like railway systems. We thus encourage more research on applications of code-related formal methods, including software model checking and static analysis by means of abstract interpretation.

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

Systems. 70% of the studies do not refer to any product standard, thus being either proprietary products or examples inspired by real cases. Product standards are considered in some cases, with ERTMS-ETCS (19%) and CBTC (8%) products. Most frequently considered systems are interlocking ones, and models of the high-level control logic describing the interaction of multiple subsystems. These are particularly common for standardized products.

Phases. The studies cover most of the railway development phases, with dominance of Architecture (66%) and Detailed Design (45%), followed by Requirements (13%), Testing (13%) and Validation (9%). Most of the studies focus on only one phase (61%), followed by two (30%) and three (6%) phases. Architecture is frequently combined with other phases, namely Detailed Design (12%), Testing (5%) and Requirements (5%).

The limited consideration of product standards is correlated with the higher attention to interlocking products, which are typically not standardized. When standard systems are considered, works focus on the verification of their high-level control logic. This is in line with the needs of the railway infrastructure managers (e.g., RFI for Italy, SNCF for France), who need to ensure that the high-level specifications are satisfied by the products developed by different vendors [8], so that they do not have to rely on a single provider. Nevertheless, formal methods are also needed for the providers themselves, as the CENELEC norms highly recommend their usage for the development of specific products [42]. Furthermore, since the platforms that need to receive the certification are the individual subsystems (e.g., ATP, Axle Counters, etc.), more research should be dedicated to the application of formal methods to the verification and validation of single, standardized, subsystems.

The statistics also show that almost all the core railway development phases can be addressed with the support of formal methods, and this is in line with the recommendations of the norms [42]. Nevertheless, additional effort should be dedicated to the later phases of the development process, and especially testing, implementation and validation, which are currently not sufficiently addressed.

RQ-I: What are the characteristics of the studies reporting industrial applications?

Demographics. The total of 79 industrial represent 24% of the whole corpus. Industrial studies are more frequently published in conferences with respect to academic ones, and they are more frequently evaluated through Experience Reports (58% of industrial studies).

Formal Methods and Techniques. Industrial studies follow the general trends for what concerns the usage of formal methods, with some differences. Specifically, the usage of semi-formal methods is more frequent in industrial studies with respect to academic ones. In addition, studies that account for code-related aspects (i.e., using code generation or static analysis) often have some industrial involvement. The most frequent combination of techniques is *model checking & simulation* (vs *theorem proving & refinement* for academic studies).

Languages and Tools. B is the most frequent modeling language family used (22%), followed by UML (18%) and Statecharts (10%). As for tools, those in the B family dominate, although NuSMV is the most frequently used individual tool (11%). Some closed-source tools have a clear industrial vocation (e.g., Prover Engines, S3 and IBM Rational). Others are applied almost exclusively in academic studies (e.g., CPN Tools and UPPAAL).

Railway Systems. Industrial applications follow the same trends as academic ones, with slightly more applications using ERTMS-ETCS (22% vs 19%) and CBTC (13% vs 8%). None of the industrial studies consider CTCS systems. Industrial works tend to give more relevance to later phases, such as Implementation and Validation, and tend to consider a combination of a higher number of phases with respect to academic works.

Industrial works are a relevant part of the identified body of literature, which confirms the vocation of the field for industrial collaborations. Some aspects also indicate that industrial studies address issues that are less considered by academic ones, such as code-related techniques, later development phases and the reference to product standards. The main characteristics observed for the whole corpus also hold for industrial studies, and discrepancies are not substantial. One distinctive element, however, is the difference between tools with academic vocation and industrial ones. This implies that some tools, even widely used and industry-ready such as UPPAAL, are rarely used in railway-specific industrial works. We thus encourage researchers in formal methods to demonstrate the effectiveness of these tools in collaboration with railway partners. Furthermore, researchers should also consider experimenting with closed-source industrial tools like Prover Engines, S3 and the IBM Rational suite. While novel formal techniques can typically not be developed by researchers on top of these platforms, the evaluation of their usage in an industrial environment can highlight other process-related issues associated to the adoption of formal methods, and can open to further research opportunities for developers of academic tools.

RQ-T: What are the emerging trends of the last years?

Demographics. There is a radical increase of studies post-2015, with a peak of 34 in 2019. The total of recent studies is 143 (44% of all studies). After an increase also in industrial studies, recent years show a decline in favor of academic ones. Papers are mainly published in specialized application-oriented venues, like RSSRail and ICIRT. The historical trend of using Examples and Experience Reports as main evaluation methods did not change over the years.

Formal Methods and Techniques. The landscape of techniques is stable, but some hot techniques exist: for simulation (31% of recent studies), SMT solving (10%), model-based testing (18%) and test generation (8%), half of the studies were published during the last 5 years. Recent works more frequently use complex combinations of techniques.

Languages and Tools. In recent years, the B language has taken the place of UML as the most common modeling language (20%) and some languages fall in the long tail, including Statecharts (1% of recent works). The *hot* tools are UPPAAL (20, 14%) and ProB (13%). Many recently used tools are specialized for railways, e.g., RobustRailS, SafeCap and OnTrack.

Railway Systems. The majority of the works is still classified as Non-standard (58%), but a slight increment is observed on works considering the ERTMS-ETCS, CBTC and CTCS standards. Interlocking is still the subject of the majority of the studies, but other subsystems (e.g., ATO, ATP and ATS) tend to be considered more frequently in recent years with respect to the past. Considered phases are in line with the historical trend, although recent works tend to address two phases instead of one only (47% vs 30%).

The last 5 years see a rich amount of works, almost half of the total number of publications starting from 1989. These papers are characterized by a higher railway specialization, in terms of venues and tools. This is in line with recommendations for the use of domain-specific formal methods already highlighted in the past [77]. Interestingly, recent works address some general shortcomings of preceding literature, like code-related aspects receiving more attention. The clear emergence of the use of tools like UPPAAL, together with the verification of non-safety critical railway systems like ATS and ATO, suggest a shift from the verification of the traditionally addressed safety problems to the verification of *availability* problems, as previously recommended by Fantechi [46].

What is worrying, however, is the decline of industrial studies in recent years. This may be due to the lower interest of industrial partners in the solutions offered by formal methods researchers, or to the stronger focus of academics on experimentation of more advanced techniques that are not industry-ready. In any case, we believe that the gap needs to be addressed to prevent disjuncture of the formal methods community from its traditional industrial connection.

6 THREATS TO VALIDITY

We discuss the threats to validity of the current study and mitigation actions according to the categories identified by Ampatzoglou *et al.* [4] for systematic reviews.

Study Selection Validity. The main threats to validity in this category are related to: (a) the construction of the search string and its possibility to fail in identifying all relevant papers; (b) the weaknesses of the search engines of the libraries used; (c) the application of inclusion/exclusion criteria and quality criteria, which could be subjective. To address (a), we piloted the string, and included a secondary search strategy through snowballing, which allowed us to identify additional papers not covered by the search string. To mitigate (b), we performed the search multiple times, in three different rounds, and considering different engines. To mitigate (c), we defined objective criteria based on previous work and piloted them, and when issues were identified, these were resolved through discussion among the authors. Furthermore, quality scores for each paper where systematically cross-checked and discussed among the authors.

Data Validity. Major threats to data validity are: (a) publication bias, as some applications of formal methods may have not appeared in research venues; (b) data extraction bias, due to possible subjectivity in data extraction; (c) bias of the classification schemes, which are oriented to identify only specific data. Threats entailed by (a) could not be mitigated entirely, although we argue that this issue is inherently reduced by the strong participation of industrial partners in the studies, and the presence of practitioner-oriented venues, such as RSSRail. To mitigate (b), the data extractors, who have complementary competences, systematically cross-checked their results, and disagreement were addressed by involving a third expert subject. To address (c), classification schemes were largely adapted from previous literature. Novel classification schemes introduced were piloted to ensure that they were correctly covering the content of the papers, using appropriate terminology.

Research Validity. To ensure research validity, we clearly reported the whole search and extraction process, and we shared the raw results our analysis, such that replication and independent analysis is possible. Research validity was further improved by the repetition of the process across three iterations, which confirmed that the adopted protocol can be replicated.

7 CONCLUSION

This paper presents a systematic mapping study of applications of formal methods in the railway domain. We retrieve 328 high-quality studies published during the last 30 years, and we classify them considering their empirical maturity, the types of formal methods applied and railway specific aspects. Furthermore, we analyze recent trends and the characteristics of those studies that involve practitioners. Our results show that the field has a strong connection with the railway industry and research is currently thriving, with dedicated venues (RSSRail, ICIRT) and specialized tools (RobustRailS, SafeCap, OnTrack). We also identify a large and diverse set of languages, techniques and tools applied to different types of railway systems, highlighting the applicability of formal methods and tools and the suitability of the domain for the application of formal methods. On the other hand, we observe that the field needs to progress in terms of empirical maturity, as most of the published works are concerned with Examples or Experience Reports rather than more rigorous research efforts. Furthermore, we also notice that most of the research has so far focused on high levels of system abstraction and early development phases, while less work has been done in later railway phases, such as code and testing. Our work complements other empirical studies performed by the authors, which previously considered the perspective of stakeholders [9, 10] and surveyed different tools for railway system design [51, 52, 76]. This paper represents the cornerstone of this research endeavor oriented to present evidence concerning the state-of-the-art of formal methods in railways. As such, it provides a literature-based framework that can be used to understand and steer the research in the field, while facilitating further synergies with the railway industry.

ACKNOWLEDGMENTS

This work has been partially funded by the ASTRail and 4SECURail projects. These projects received funding from the Shift2Rail Joint Undertaking under the European Union's Horizon 2020 research and innovation programme under grant agreements No. 777561 (ASTRail) and 881775 (4SECURail). The content of this paper reflects only the authors' view and the Shift2Rail Joint Undertaking is not responsible for any use that may be made of the included information.

REFERENCES

[1] Jean-Raymond Abrial. 2006. Formal methods in industry: achievements, problems, future. In *Proceedings of the 28th International Conference on Software Engineering (ICSE'06)*. ACM, 761–768.

[2] Jean-Raymond Abrial. 2007. Formal Methods: Theory Becoming Practice. *J. Univers. Comput. Sci.* 13, 5 (2007), 619–628.

[3] Gul Agha and Karl Palmskog. 2018. A Survey of Statistical Model Checking. *ACM Trans. Model. Comput. Simul.* 28, 1 (2018), 6:1–6:39.

[4] Apostolos Ampatzoglou, Stamatia Bibi, Paris Avgeriou, Marijn Verbeek, and Alexander Chatzigeorgiou. 2019. Identifying, categorizing and mitigating threats to validity in software engineering secondary studies. *Inf. Softw. Technol.* 106 (2019), 201–230.

[5] Marc Antoni. 2010. Complementarity between Axle Counters and Tracks Circuits. In *Proceedings of the 8th Symposium on Formal Methods for Automation and Safety in Railway and Automotive Systems (FORMS/FORMAT'10)*, Eckehard Schnieder and Géza Tarnai (Eds.). Springer, 65–76.

[6] Frédéric Badeau and Arnaud Amelot. 2005. Using B as a High Level Programming Language in an Industrial Project: Roissy VAL. In *Proceedings of the 4th International Conference of B and Z Users (ZB'05) (LNCS)*, Helen Treharne, Steve King, Martin C. Henson, and Steve A. Schneider (Eds.), Vol. 3455. Springer, 334–354.

[7] Christel Baier and Joost-Pieter Katoen. 2008. *Principles of Model Checking.* MIT Press.

[8] Davide Basile, Maurice H. ter Beek, Alessandro Fantechi, Alessio Ferrari, Stefania Gnesi, Laura Masullo, Franco Mazzanti, Andrea Piattino, and Daniele Trentini. 2020. Designing a Demonstrator of Formal Methods for Railways Infrastructure Managers. In *Proceedings of the 9th International Symposium on Leveraging Applications of Formal Methods, Verification and Validation: Applications (ISoLA'20) (LNCS)*, Tiziana Margaria and Bernhard Steffen (Eds.), Vol. 12478. Springer, 467–485.

[9] Davide Basile, Maurice H. ter Beek, Alessandro Fantechi, Stefania Gnesi, Franco Mazzanti, Andrea Piattino, Daniele Trentini, and Alessio Ferrari. 2018. 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) (LNCS)*, Carlo A. Furia and Kirsten Winter (Eds.), Vol. 11023. Springer, 20–29.

[10] Maurice H. ter Beek, Arne Borälv, Alessandro Fantechi, Alessio Ferrari, Stefania Gnesi, Christer Löfving, and Franco Mazzanti. 2019. Adopting Formal Methods in an Industrial Setting: The Railways Case. In *Proceedings of the 3rd World Congress on Formal Methods: The Next 30 Years (FM'19) (LNCS)*, Maurice H. ter Beek, Annabelle McIver, and José N. Oliveira (Eds.), Vol. 11800. Springer, 762–772.

[11] Patrick Behm, Paul Benoit, Alain Faivre, and Jean-Marc Meynadier. 1999. Météor: A Successful Application of B in a Large Project. In *Proceedings of the 1st World Congress on Formal Methods in the Development of Computing Systems (FM'99) (LNCS)*, Jeannette M. Wing, Jim Woodcock, and Jim Davies (Eds.), Vol. 1708. Springer, 369–387.

[12] Armin Biere, Marijn Heule, Hans van Maaren, and Toby Walsh (Eds.). 2009. *Handbook of Satisfiability.* Frontiers in Artificial Intelligence and Applications, Vol. 185. IOS Press.

[13] Dines Bjørner. 2003. New Results and Trends in Formal Techniques and 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)*, Géza Tarnai and Eckehard Schnieder (Eds.). L'Harmattan.

[14] Dines Bjørner and Klaus Havelund. 2014. 40 Years of Formal Methods: Some Obstacles and Some Possibilities?. In *Proceedings of the 19th International Symposium on Formal Methods (FM'14) (LNCS)*, Cliff Jones, Pekka Pihlajasaari, and Jun Sun (Eds.), Vol. 8442. Springer, 42–61.

[15] Robin E. Bloomfield, Dan Craigen, Frank Koob, Markus Ullmann, and Stefan Wittmann. 2000. Formal Methods Diffusion: Past Lessons and Future Prospects. In *Proceedings of the 19th International Conference on Computer Safety, Reliability and Security (SAFECOMP'00) (LNCS)*, Floor Koornneef and Meine van der Meulen (Eds.), Vol. 1943. Springer, 211–226.

[16] Jean-Louis Boulanger (Ed.). 2012. *Formal Methods: Industrial Use from Model to the Code.* Wiley-ISTE.

[17] Jean-Louis Boulanger (Ed.). 2012. *Industrial Use of Formal Methods: Formal Verification.* Wiley-ISTE.

[18] Jean-Louis Boulanger (Ed.). 2014. *Formal Methods Applied to Industrial Complex Systems: Implementation of the B Method.* Wiley.

[19] Jonathan P. Bowen, Ricky W. Butler, David L. Dill, Robert L. Glass, David Gries, Anthony Hall, Michael G. Hinchey, C. Michael Holloway, Daniel Jackson, Cliff B. Jones, Michael J. Lutz, David Lorge Parnas, John M. Rushby, Jeannette M. Wing, and Pamela Zave. 1996. An Invitation to Formal Methods. *Comput.* 29, 4 (1996), 16–30.

[20] Jonathan P. Bowen and Michael G. Hinchey. 1995. Seven More Myths of Formal Methods. *IEEE Softw.* 12, 4 (1995), 34–41.

[21] Jonathan P. Bowen and Michael G. Hinchey. 1995. Ten Commandments of Formal Methods. *Comput.* 28, 4 (1995), 56–63.

[22] Jonathan P. Bowen and Michael G. Hinchey. 2006. Ten Commandments of Formal Methods ...Ten Years Later. *Comput.* 39, 1 (2006), 40–48.

[23] Jonathan P. Bowen and Michael G. Hinchey. 2014. Formal Methods. In *Computing Handbook*, Teofilo F. Gonzalez, Jorge Diaz-Herrera, and Allen Tucker (Eds.). CRC, Chapter 71, 71–25.

[24] Jonathan P. Bowen and Victoria Stavridou. 1993. The Industrial Take-up of Formal Methods in Safety-Critical and Other Areas: A Perspective. In *Proceedings of the 1st International Symposium of Formal Methods Europe (FME'93) (LNCS)*, Jim Woodcock and Peter Gorm Larsen (Eds.), Vol. 670. Springer, 183–195.

[25] Michael Butler, Philipp Körner, Sebastian Krings, Thierry Lecomte, Michael Leuschel, Luis-Fernando Mejia, and Laurent Voisin. 2020. The First Twenty-Five Years of Industrial Use of the B-Method. In *Proceedings of the 25th International Conference on Formal Methods for Industrial Critical Systems (FMICS'20) (LNCS)*, Maurice H. ter Beek and Dejan Ničković (Eds.), Vol. 12327. Springer, 189–209.

[26] Benoît Caillaud, Philippe Darondeau, Luciano Lavagno, and Xiaolan Xie (Eds.). 2002. *Synthesis and Control of Discrete Event Systems.* Springer.

[27] Lianping Chen and Muhammad Ali Babar. 2011. A systematic review of evaluation of variability management approaches in software product lines. *Inf. Softw. Technol.* 53, 4 (2011), 344–362.

[28] Angelo Chiappini, Alessandro Cimatti, Luca Macchi, Oscar Rebollo, Marco Roveri, Angelo Susi, Stefano Tonetta, and Berardino Vittorini. 2010. Formalization and validation of a subset of the European Train Control System. In *Proceedings of the 32nd International Conference on Software Engineering (ICSE'10).* ACM, 109–118.

[29] Edmund M. Clarke, Thomas A. Henzinger, Helmut Veith, and Roderick Bloem (Eds.). 2018. *Handbook of Model Checking.*

[30] Edmund M. Clarke, Jeannette M. Wing, et al. 1996. Formal Methods: State of the Art and Future Directions. *ACM Comput. Surv.* 28, 4 (1996), 626–643.

[31] Dan Craigen. 1995. Formal Methods Technology Transfer: Impediments and Innovation. In *Proceedings of the 6th International Conference on Concurrency Theory (CONCUR'95) (LNCS)*, Insup Lee and Scott A. Smolka (Eds.), Vol. 962. Springer, 328–332.

[32] Dan Craigen. 1999. Formal Methods Adoption: What's Working, What's Not!. In *Proceedings of the 5th and 6th International International Workshops on Theoretical and Practical Aspects of SPIN Model Checking (SPIN'99) (LNCS)*, Dennis Dams, Rob Gerth, Stefan Leue, and Mieke Massink (Eds.), Vol. 1680. Springer, 77–91.

[33] Dan Craigen, Susan Gerhart, and Ted Ralston. 1995. *Industrial Applications of Formal Methods to Model, Design and Analyze Computer Systems: An International Survey.* William Andrew.

[34] Dan Craigen, Susan L. Gerhart, and Ted Ralston. 1992. An International Survey of Industrial Applications of Formal Methods. In *Proceedings of the 7th Z User Workshop*, Jonathan P. Bowen and John E. Nicholls (Eds.). Springer, 1–5.

[35] Dan Craigen, Susan L. Gerhart, and Ted Ralston. 1995. Formal Methods Reality Check: Industrial Usage. *IEEE Trans. Softw. Eng.* 21, 2 (1995), 90–98.

[36] Clara DaSilva, Babak Dehbonei, and Fernando Mejia. 1992. Formal specification in the development of industrial applications: Subway speed control system. In *Proceedings of the IFIP TC6/WG6.1 5th International Conference on Formal Description Techniques for Distributed Systems and Communication Protocols (FORTE'92) (IFIP Transactions)*, Michel Diaz and Roland Groz (Eds.), Vol. C-10. North-Holland, 199–213.

[37] Marian Daun, Carolin Hübscher, and Thorsten Weyer. 2017. Controlled Experiments with Student Participants in Software Engineering: Preliminary Results from a Systematic Mapping Study. arXiv:1708.04662

[38] Jennifer A. Davis, Matthew A. Clark, Darren D. Cofer, Aaron Fifarek, Jacob Hinchman, Jonathan A. Hoffman, Brian W. Hulbert, Steven P. Miller, and Lucas G. Wagner. 2013. 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) (LNCS)*, Charles Pecheur and Michael Dierkes (Eds.), Vol. 8187. Springer, 63–77.

[39] Falk Dietrich and Jean-Pierre Hubaux. 2002. Formal methods for communication services: meeting the industry expectations. *Comput. Networks* 38, 1 (2002), 99–120.

[40] Brijesh Dongol, Luigia Petre, and Graeme Smith (Eds.). 2019. *Proceedings of the 3rd International Workshop and Tutorial on Formal Methods Teaching (FMTea'19).* LNCS, Vol. 11758. Springer.

[41] Tore Dybå and Torgeir Dingsøyr. 2008. Empirical studies of agile software development: A systematic review. *Inf. Softw. Technol.* 50, 9 (2008), 833–859.

[42] European Committee for Electrotechnical Standardization. 2011. CENELEC EN 50128 — Railway applications – Communication, signalling and processing systems – Software for railway control and protection systems. https://standards.globalspec.com/std/1678027/cenelec-en-50128.

[43] Davide Falessi, Natalia Juristo, Claes Wohlin, Burak Turhan, Jürgen Münch, Andreas Jedlitschka, and Markku Oivo. 2018. Empirical software engineering experts on the use of students and professionals in experiments. *Empir. Softw. Eng.* 23, 1 (2018), 452–489.

[44] Wei Fang, Shengxiang Yang, and Xin Yao. 2015. A Survey on Problem Models and Solution Approaches to Rescheduling in Railway Networks. *IEEE Trans. Intell. Transp. Syst.* 16, 6 (2015), 2997–3016.

[45] Alessandro Fantechi. 2012. The Role of Formal Methods in Software Development for Railway Applications. In *Railway Safety, Reliability, and Security:Technologies and Systems Engineering*, Francesco Flammini (Ed.). IGI, Chapter12, 282–297.

[46] Alessandro Fantechi. 2013. 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 (LNCS)*,

Steve Counsell and Manuel Núñez (Eds.), Vol. 8368. Springer, 167–183.

[47] Alessandro Fantechi, Wan Fokkink, and Angelo Morzenti. 2013. Some Trends in Formal Methods Applications to Railway Signaling. In *Formal Methods for Industrial Critical Systems: A Survey of Applications*, Stefania Gnesi and Tiziana Margaria (Eds.). Wiley, Chapter 4, 61–84.

[48] Alessio Ferrari, Alessandro Fantechi, and Stefania Gnesi. 2012. Lessons Learnt from the Adoption of Formal Model-Based Development. In *Proceedings of the 4th International Symposium on NASA Formal Methods (NFM'12) (LNCS)*, Alwyn Goodloe and Suzette Person (Eds.), Vol. 7226. Springer, 24–38.

[49] Alessio Ferrari, Alessandro Fantechi, Stefania Gnesi, and Gianluca Magnani. 2013. Model-based development and formal methods in the railway industry. *IEEE Softw.* 30, 3 (2013), 28–34.

[50] Alessio Ferrari, Alessandro Fantechi, Gianluca Magnani, Daniele Grasso, and Matteo Tempestini. 2013. The Metrô Rio case study. *Sci. Comput. Program.* 78, 7 (2013), 828–842.

[51] Alessio Ferrari, Franco Mazzanti, Davide Basile, and Maurice H. ter Beek. 2021. Systematic Evaluation and Usability Analysis of Formal Tools for Railway System Design. arXiv:2101.11303v2

[52] Alessio Ferrari, Franco Mazzanti, Davide Basile, Maurice H. ter Beek, and Alessandro Fantechi. 2020. Comparing Formal Tools for System Design: a Judgment Study. In *Proceedings of the 42nd International Conference on Software Engineering (ICSE'20)*. ACM, 62–74.

[53] Alessio Ferrari, Maurice H. ter Beek, Franco Mazzanti, Davide Basile, Alessandro Fantechi, Stefania Gnesi, Andrea Piattino, and Daniele Trentini. 2019. 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) (LNCS)*, Vol. 11495. Springer, 226–241.

[54] Hubert Garavel and Susanne Graf. 2013. *Formal Methods for Safe and Secure Computer Systems*. BSI Study 875. Bundesamt für Sicherheit in der Informationstechnik. https://www.bsi.bund.de/DE/Publikationen/Studien/Formal_Methods_Study_875/study_875.html

[55] Hubert Garavel and Radu Mateescu. 2019. Reflections on Bernhard Steffen's Physics of Software Tools. In *Models, Mindsets, Meta: The What, the How, and the Why Not?*, Tiziana Margaria, Susanne Graf, and Kim Larsen (Eds.). LNCS, Vol. 11200. Springer, 186–207.

[56] Hubert Garavel, Maurice H. ter Beek, and Jaco van de Pol. 2020. The 2020 Expert Survey on Formal Methods. In *Proceedings of the 25th International Conference on Formal Methods for Industrial Critical Systems (FMICS'20) (LNCS)*, Maurice H. ter Beek and Dejan Ničković (Eds.), Vol. 12327. Springer, 3–69.

[57] Susan L. Gerhart, Dan Craigen, and Ted Ralston. 1994. Experience with Formal Methods in Critical Systems. *IEEE Softw.* 11, 1 (1994), 21–28.

[58] Mario Gleirscher and Diego Marmsoler. 2020. Formal Methods in Dependable Systems Engineering: A Survey of Professionals from Europe and North America. *Empir. Softw. Eng.* 25, 6 (2020), 4473–4546.

[59] Stefania Gnesi and Tiziana Margaria (Eds.). 2013. *Formal methods for industrial critical systems: A survey of applications.*

[60] Gérard Guiho and Claude Hennebert. 1990. SACEM Software Validation. In *Proceedings of the 12th International Conference on Software Engineering (ICSE'90)*. IEEE, 186–191.

[61] Anthony Hall. 1990. Seven Myths of Formal Methods. *IEEE Softw.* 7, 5 (1990), 11–19.

[62] Anthony Hall. 2007. Realising the Benefits of Formal Methods. *J. Univers. Comput. Sci.* 13, 5 (2007), 669–678.

[63] Anthony Hall, David Lorge Parnas, Nico Plat, John M. Rushby, and Chris T. Sennett. 1995. The Future of Formal Methods in Industry. In *Proceedings of the 9th International Conference of Z Users (ZUM'95) (LNCS)*, Jonathan P. Bowen and Michael G. Hinchey (Eds.), Vol. 967. Springer, 237–242.

[64] Anne E. Haxthausen. 2010. *An Introduction to Formal Methods for the Development of Safety-critical Applications.* Technical Report. Technical University of Denmark. https://orbit.dtu.dk/files/137536957/FormalMethodsNoteTS.pdf

[65] Michael G. Hinchey and Jonathan P. Bowen (Eds.). 1999. *Industrial-Strength Formal Methods In Practice.* Springer.

[66] Victoria J. Hodge, Simon O'Keefe, Michael Weeks, and Anthony Moulds. 2015. Wireless Sensor Networks for Condition Monitoring in the Railway Industry: A Survey. *IEEE Trans. Intell. Transp. Syst.* 16, 3 (2015), 1088–1106.

[67] IEEE. 2004. 1474.1-2004 – IEEE Standard for Communications-Based Train Control (CBTC) Performance and Functional Requirements. https://doi.org/10.1109/IEEESTD.2004.95746

[68] International Electrotechnical Commission. 2019. IEC 62290-1–3: Railway applications – Urban guided transport management and command/control systems – Parts 1–3. http://webstore.iec.ch/publication/28078.

[69] Daniel Jackson. 2001. Lightweight Formal Methods. In *Proceedings of the 10th International Symposium of Formal Methods Europe: Formal Methods for Increasing Software Productivity (FME'01) (LNCS)*, José N. Oliveira and Pamela Zave (Eds.), Vol. 2021. Springer, 1–1.

[70] Trevor King. 1994. Formalising British Rail's Signalling Rules. In *Proceedings of the 2nd International Symposium of Formal Methods Europe: Industrial Benefit of Formal Methods (FME'94) (LNCS)*, Maurice Naftalin, B. Tim Denvir, and Miquel Bertran (Eds.), Vol. 873. Springer, 45–54.

[71] Barbara Kitchenham. 2004. *Procedures for Performing Systematic Reviews.* Technical Report TR/SE-0401. Keele University.

[72] Andrew J. Ko, Thomas D. LaToza, and Margaret M. Burnett. 2015. A practical guide to controlled experiments of software engineering tools with human participants. *Empir. Softw. Eng.* 20, 1 (2015), 110–141.

[73] Thierry Lecomte, David Déharbe, Étienne Prun, and Erwan Mottin. 2017. 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) (LNCS)*, Simone Cavalheiro and José Fiadeiro (Eds.), Vol. 10623. Springer, 70–87.

[74] Richard Martin Lusby, Jesper Larsen, and Simon Bull. 2018. A survey on robustness in railway planning. *Eur. J. Oper. Res.* 266, 1 (2018), 1–15.

[75] Juliette Marais, Julie Beugin, and Marion Berbineau. 2017. A Survey of GNSS-Based Research and Developments for the European Railway Signaling. *IEEE Trans. Intell. Transp. Syst.* 18, 10 (2017), 2602–2618.

[76] Franco Mazzanti, Alessio Ferrari, and Giorgio O. Spagnolo. 2018. Towards formal methods diversity in railways: an experience report with seven frameworks. *Int. J. Softw. Tools Technol. Transf.* 20, 3 (2018), 263–288.

[77] John McDermid, Andy Galloway, Simon Burton, John Clark, Ian Toyn, Nigel Tracey, and Samuel Valentine. 1998. Towards Industrially Applicable Formal Methods: Three Small Steps, and One Giant Leap. In *Proceedings of the 2nd International Conference on Formal Engineering Methods (ICFEM'98)*. IEEE, 76–88.

[78] James Bret Michael, George W. Dinolt, and Doron Drusinsky. 2020. Open Questions in Formal Methods. *Comput.* 53, 5 (2020), 81–84.

[79] Steven P. Miller. 2012. Lessons from Twenty Years of Industrial Formal Methods. In *Proceedings of the 20th High Confidence Software and Systems Conference (HCSS'12)*. http://cps-vo.org/node/3434

[80] Steven P. Miller, Michael W. Whalen, and Darren D. Cofer. 2010. Software Model Checking Takes Off. *Commun. ACM* 53, 2 (2010), 58–64.

[81] Juan Moreno, José Manuel Riera, Leandro de Haro, and Carlos Rodríguez. 2015. A survey on future railway radio communications services: challenges and opportunities. *IEEE Commun. Mag.* 53, 10 (2015), 62–68.

[82] Glenford J. Myers, Corey Sandler, and Tom Badgett. 2011. *The Art of Software Testing*. Wiley.

[83] Graeme I. P. Parkin and Stephen Austin. 1993. Overview: Survey of Formal Methods in Industry. In *Proceedings of the 6th IFIP TC6/WG6.1 International Conference on Formal Description Techniques (FORTE'93) (IFIP Transactions)*, Richard L. Tenney, Paul D. Amer, and M. Ümit Uyar (Eds.), Vol. C-22. North-Holland, 189–203.

[84] Martin Penicka and Dines Bjørner. 2004. From railway resource planning to train operation: a brief survey of complementary formalisations. In *Proceedings of the IFIP 18th World Computer Congress: Topical Sessions (WCC'04) (IFIP Advances in Information and Communication Technology)*, René Jacquart (Ed.), Vol. 156. Springer, 629–636.

[85] John Alan Robinson and Andrei Voronkov (Eds.). 2001. *Handbook of Automated Reasoning*. Elsevier.

[86] Per Runeson, Martin Höst, Austen Rainer, and Björn Regnell. 2012. *Case Study Research in Software Engineering: Guidelines and Examples*. Wiley.

[87] John Rushby. 1993. *Formal Methods and the Certification of Critical Systems*. Technical Report SRI-CSL-93-7. Computer Science Laboratory, SRI International. http://www.csl.sri.com/papers/csl-93-7/

[88] Johnny Saldaña. 2021. *The Coding Manual for Qualitative Researchers*. SAGE.

[89] Muhammad Saqib Nawaz, Moin Malik, Yi Li, Meng Sun, and Muhammad Ikram Ullah Lali. 2019. A Survey on Theorem Provers in Formal Methods. arXiv:1912.03028

[90] Bernhard Steffen. 2017. The Physics of Software Tools: SWOT Analysis and Vision. *Int. J. Softw. Tools Technol. Transfer* 19, 1 (2017), 1–7.

[91] Martyn Thomas. 1990. The role of formal methods in developing safety-critical software. In *Proceedings of the IEE Colloquium on Safety Critical Software in Vehicle and Traffic Control*. IET, 9/1–9/3.

[92] Martyn Thomas. 1993. The industrial use of formal methods. *Microprocess. Microsystems* 17, 1 (1993), 31–36.

[93] Edward Tsang. 1993. *Foundations of Constraint Satisfaction*. Academic Press.

[94] Paul Unterhuber, Stephan Pfletschinger, Stephan Sand, Mohammad Soliman, Thomas Jost, Aitor Arriola, Iñaki Val, Cristina Cruces, Juan Moreno, Juan Pablo García-Nieto, Carlos Rodríguez, Marion Berbineau, Eneko Echeverría, and Imanol Baz. 2016. A Survey of Channel Measurements and Models for Current and Future Railway Communication Systems. *Mob. Inf. Syst.* 2016 (2016), 7308604:1–7308604:14.

[95] Jeannette M. Wing. 1990. A Specifier's Introduction to Formal Methods. *Comput.* 23, 9 (1990), 8–24.

[96] Claes Wohlin, Per Runeson, Martin Höst, Magnus C. Ohlsson, Björn Regnell, and Anders Wesslén. 2012. *Experimentation in Software Engineering*. Springer.

[97] Jim Woodcock, Peter Gorm Larsen, Juan Bicarregui, and John Fitzgerald. 2009. Formal methods: Practice and experience. *ACM Comput. Surv.* 41, 4 (2009), 19:1–19:36.

[98] Shengfeng Xu, Gang Zhu, Bo Ai, and Zhangdui Zhong. 2016. A survey on high-speed railway communications: A radio resource management perspective. *Comput. Commun.* 86 (2016), 12–28.

Received July 2021