A History of Formal Methods in Railways

MAURICE H. TER BEEK*, CNR–ISTI, Italy
ALESSANDRO FANTECHI, University of Florence, Italy
ALESSIO FERRARI, University College Dublin, Ireland
STEFANIA GNESI, CNR–ISTI, Italy
ANNE E. HAXTHAUSEN, Technical University of Denmark, Denmark
THIERRY LECOMTE, CLEARSY Systems Engineering, France

The engineering of industrial systems, particularly in safety-critical domains such as railways, demands rigorous verification and validation processes to ensure system dependability. Formal methods have emerged as powerful tools to complement traditional software engineering practices. In the railway sector, which increasingly relies on complex, distributed, and cyber-physical control systems, formal methods have demonstrated particular value for many decades now. In this paper, we provide a retrospective overview of the application of formal methods and tools in the railway domain, with emphasis on two prominent verification approaches and one frequently verified railway system: modeling and validation with the B method and tools and formal verification of interlocking systems by model checking. We explore their role in the design and development of key railway systems, highlighting both academic research and industrial success stories, as witnessed by international projects and initiatives. We conclude with an outlook on the potential of integrating AI and formal methods to enhance the efficiency of next-generation railway systems.

CCS Concepts: • **Software and its engineering → Formal methods**; • **Social and professional topics → History of computing**.

Additional Key Words and Phrases: Formal Methods, Railways

ACM Reference Format:
Maurice H. ter Beek, Alessandro Fantechi, Alessio Ferrari, Stefania Gnesi, Anne E. Haxthausen, and Thierry Lecomte. 2026. A History of Formal Methods in Railways. *Form. Asp. Comput.* 38, 3 (September 2026), 32 pages. https://doi.org/10.1145/3802545

Acronyms:

AI	Artificial Intelligence		Terrestre(French Land Transport Accident Investigation Bureau)	**CTCS**	Chinese Train Control System
APS	Autonomous Positioning System			**ERA**	European Railway Agency
ATC	Automatic Train Control	**CBTC**	Communication-Based Train Control	**ERTMS**	European Rail Traffic Management System
ATO	Automatic Train Operation				
ATP	Automatic Train Protection	**CENELEC**	Comité Européen de Normalisation Électrotechnique (European committee for electrotechnical standardization)	**ETCS**	European Train Control System
ATS	Automatic Train Supervision			**FM**	Formal Method
BEA-TT	Bureau d'Enquêtes sur les Accidents de Transport			**GE**	General Electric
				HL3	Hybrid Level 3

*First and corresponding author

Authors' Contact Information: Maurice H. ter Beek, maurice.terbeek@isti.cnr.it, CNR–ISTI, Pisa, Italy; Alessandro Fantechi, alessandro.fantechi@unifi.it, University of Florence, Florence, Italy; Alessio Ferrari, alessio.ferrari@ucd.ie, University College Dublin, Dublin, Ireland; Stefania Gnesi, stefania.gnesi@isti.cnr.it, CNR–ISTI, Pisa, Italy; Anne E. Haxthausen, aeha@dtu.dk, Technical University of Denmark, Kongens Lyngby, Denmark; Thierry Lecomte, thierry.lecomte@clearsy.com, CLEARSY Systems Engineering, Aix-en-Provence, France.

Manuscript submitted to ACM

IXL	Interlocking	**SACEM**	Système d'Aide à la Conduite, à l'Exploitation et à la Maintenance (driver assistance, operation, and maintenance system)	**SNCF**	Société nationale des chemins de fer français (national company of the French railways)
LLM	Large Language Model				
MA	Movement Authority				
NYCT	New York City Transit			**SSCS**	Subway Speed Control System
RATP	Régie Autonome des Transports Parisiens (Parisian Transport Autonomous Administration)	**SAT**	Boolean Satisfiability problem	**SysML**	Systems Modeling Language
		SIL	Safety Integrity Level	**S3**	Systerel Smart Solver
		SMC	Statistical Model Checking	**UIC**	Union Internationale des Chemins de fer (International Union of Railways)
RBC	Radio Block Center	**SMT**	Satisfiability Modulo Theories		
RER	Réseau Express Régional (regional express network)				
				UML	Unified Modeling Language

1 Introduction

The engineering of industrial systems typically follows a structured development methodology comprising successive phases, including requirements analysis, architectural and detailed design, implementation, verification, validation, and maintenance. Each phase yields formal or semi-formal artifacts (models) that must be internally consistent and aligned with preceding and subsequent phases. Ensuring this consistency—both within and across phases—is essential for building dependable systems, especially in safety- and mission-critical domains. Formal methods, which are mathematically rigorous techniques for the specification and (manual or automated) verification of software and hardware systems, offer strong guarantees of correctness and serve as a crucial complement to traditional software engineering practices.

Formal methods support the precise modeling of system requirements, behavior, and environments, enabling exhaustive analysis through mathematically defined notions of correctness. Unlike testing and simulation—which provide probabilistic or partial evidence—formal methods can rigorously prove the absence or presence of undesirable behavior by reasoning over the entire state space of a system. The two principal approaches are *model checking* [14, 63] and *theorem proving* [152, 164]. Theorem provers (e.g., the Rocq Prover (formerly Coq [30]), Isabelle [153], and Z3 [151]) facilitate expressive, user-guided reasoning about highly abstract or infinite-state systems, whereas model checkers (e.g., SPIN [125], NuSMV [59]/nuXmv [55], ProB [141], UPPAAL [26–28, 71, 72], FDR4 [109], CADP [104], and mCRL2 [12]) automate the verification of finite-state models against formal specifications, offering push-button analysis subject to state-space limitations.[1]

Extensions such as probabilistic model checking allow analysis of systems with stochastic or uncertain behavior, yielding quantitative guarantees (e.g., satisfaction probabilities of temporal properties). Statistical Model Checking (SMC) [6, 140], in particular, provides scalable, simulation-based verification with quantifiable confidence levels, although it remains challenged by occurrences of rare events. Arguably the best known SMC tool is UPPAAL SMC [72]. Complementary techniques, including static analysis, model-based testing, and symbolic/concolic execution, offer additional ways to explore or approximate system behavior, trading off between precision, automation, and scalability.

As software systems grow in complexity and criticality, the integration of formal methods into industrial development lifecycles becomes increasingly important. Despite challenges such as tool complexity, required expertise, and computational limits, formal methods deliver a level of assurance unattainable by empirical techniques alone. Their ability to rigorously validate models, detect subtle errors early, and provide exhaustive guarantees, makes them indispensable for the development of reliable, robust, and verifiably correct systems.

Modern railways are in most cases controlled by real-time computer-based systems. Those systems feature embedded, cyber-physical, distributed, and heterogeneous architectures, which are increasingly large and complex. To fulfill the safety requirements, railway control systems must undergo extensive verification and validation, which is typically rather time-consuming when conducted by intensive software testing. Model-based analyses and formal methods promise to make such verification activities less error-prone, therefore increasing the effectiveness and efficiency of the overall process.

The railway domain has traditionally been considered one of the most fruitful areas of application for formal methods. In this paper, we provide a retrospective on the history of formal methods in railways, highlighting the types of systems and subsystems in which

[1] At the end of the paper, we collect pointers to public resources for all the mentioned tools.

these approaches have been applied, including interlocking, automatic train control systems, and others, and illustrating the research and practical experiences conducted over the years. Our aim is not to present an exhaustive or definitive account, but rather to offer a historically informed narrative that brings into focus major technological strands, representative applications, and long-term trends.

In contrast to comprehensive and data-driven surveys such as the systematic mapping study by Ferrari and ter Beek [99], which categorizes and quantifies the literature, we adopt a selective and narrative viewpoint. Given the breadth of the field and the intricate relations among its many contributions, a fully systematic reconstruction would be beyond the scope of a single article. Instead, we focus on two particularly representative and influential strands, the B method and the use of model checking for interlocking systems, which have played a central role both in research and in industrial practice, as motivated next. Moreover, our historical narrative focuses primarily on European developments documented in English-language scientific publications, while we refer to existing surveys for broader geographical coverage and complementary perspectives.

The landscape of formal methods and tools used in the railway domain is broad [18, 99]. This makes it difficult for practitioners to decide which formal method to adopt. According to the aforementioned mapping study by Ferrari and ter Beek [99], the Unified Modeling Language (UML) is the dominant modeling language (cf., e.g., [39, 58, 61, 62, 113, 171, 183]), second to the B method during the most recent years [99, Fig. 9: Modeling language families considered in the studies]. UML is typically used in combination with different tools; a high-level UML model is translated into the input language of the specific formal verification tool used. Moreover, the B method and its evolution Event-B, together with their supporting tools such as ProB, Atelier B, and Rodin, are among the most widely used formal methods and tools, in particular among the most recent and the industrial studies [99, Figs. 9 and 10: Tools], with numerous success stories (cf. Section 4). Applications of the B method started in the early 90s [25, 70, 112] and the method evolved together with railway technologies and industrial priorities, until the latest development of tools, such as OVADO,[2] supporting the validation of configuration data (cf. Section 4.3).

Furthermore, the mapping study showed that model checking is by far the most commonly adopted formal verification technique, used in roughly half of the studies [99, Fig. 7: Techniques]. Frequently used tools that are not from the B family include NuSMV/nuXmv (cf., e.g., [58, 61, 62]) and UPPAAL (cf., e.g., [15, 19, 178]) [99, Fig. 10: Tools], which has also enabled SMC to address scalability issues in the verification of railway systems of increasing complexity (cf. Section 5.2.2).

Finally, the mapping study showed that a vast majority of the literature (two fifths of the studies [99, Fig. 13: Category of railway subsystem]) focuses on interlocking platforms, among the different railway systems. Interlocking systems have been analyzed and validated through different technologies, including model checking and SMT solving, and more recent work focuses on analyzing more advanced system architectures such as distributed interlocking systems. Other systems, such as level crossings, platform screen doors, as well as more advanced railway signaling platforms, such as the moving block distancing system, have also gone through experimentation with formal methods [99, Fig. 13]. Notably, many of the studies are associated with real-world applications of formal methods since the early days dating back to the final decade of the last century, which indicate the practical suitability of these approaches, despite well-known scalability problems, such as, e.g., state-space explosion.

For the above mentioned reasons, this retrospective provides a more extensive treatment of these three topics as the two main contributions (namely, a history of the B method applied in the railway domain in Section 4 and a history of formal verification of interlocking systems by model checking in Section 5) compared to other methods and systems, which are however surveyed (in Section 6) as they help to characterize the large variety of solutions and applications.

Overall, this indicates that formal methods have demonstrated a sustained relevance in the railway domain, not only in research but also in industrial deployment. The continued adoption of formal methods underlines their adaptability to evolving system requirements and increasing complexity. As systems become more distributed and data-intensive, especially with the rise of cyber-physical and autonomous railway technologies, formal methods are increasingly being integrated into larger toolchains that support verification, simulation, and validation at various stages of the development lifecycle. Furthermore, the emergence of domain-specific modeling languages and domain-aware verification strategies has enhanced the usability and effectiveness of formal methods in practice. These developments have facilitated better abstraction mechanisms, modular verification strategies, and the combination of multiple formal

[2] https://www.ovado.net/

techniques (e.g., model checking, model-based development and testing, and abstract interpretation) to tackle challenges such as integration testing and code generation. Industry consortia and standardization bodies have also played a significant role in shaping the application of formal methods in railway contexts, encouraging the alignment of formal models with certification standards like the CENELEC standards EN 50128 [80], EN 50129 [82], and the recently adopted EN 50716 [83] for railway applications. This alignment has not only improved the credibility of formal methods in safety assurance processes but also fostered their adoption in safety-critical subsystems across various global railway infrastructures.

Ultimately, this evolving landscape confirms that formal methods, though demanding in terms of expertise and computational resources, continue to offer robust solutions to ensure the dependability of railway systems. Their integration into industrial practice—through tool support, case studies, and compliance with standards—indicates a maturing ecosystem where formal methods serve both as analytical instruments and as enablers of innovation in safety-critical system design.

This paper is structured as follows. Section 2 surveys related reviews, focusing on those with specific attention to the railway domain, and further motivates the reason to focus on the B method and on model checking interlocking systems. Section 3 discusses the main context in which the application of formal methods to railway systems has been developed. Sections 4 and 5 provide a historical perspective of the two main areas of long-running experience of the application of formal methods in railways: on the one hand the success stories of the B method on train distancing control systems, and on the other hand verification by model checking of railway interlocking systems. Section 6 offers glimpses of other railway equipment that has captured the interest of the formal methods community. Section 7 serves as source of information on the vast body of work developed on the theme of this paper by summarizing the projects and initiatives that have accompanied and supported research on railways. Section 8 discusses future work, after which Section 9 concludes the paper.

2 Railways in Related Surveys on Formal Methods

Formal methods have been the subject of academic research as well as the focus of industrial applications for several decades. Their adoption in the development of safety-critical software systems can be traced back to the early 1990s, with foundational contributions from both academics [157, 165, 179] and industrial practitioners [174, 175].

Significant early accounts document the challenges and lessons learned from deploying formal methods in real-world settings [45, 107]. One of the earliest and most comprehensive evaluations concerns a systematic survey analyzing the role of formal methods in industrial projects [67, 68]. The authors drew upon twelve industrial case studies, including the development of Automatic Train Protection (ATP) systems for subway networks in Paris and Calcutta [70, 112].

The influential 1996 survey on formal methods by Clarke and Wing and colleagues [64] presents a range of case studies related to specification and verification, notably including the previously mentioned railway signaling application, as well as an additional study focused on formalizing the signaling rules for railway interlocking systems [132].

In the early 2000s, initial informal surveys specifically focused on the railway domain started to appear [31, 161], offering anecdotal yet insightful perspectives on formal techniques and tools, along with illustrative examples of their application within railway systems. Noteworthy contributions include an accessible tutorial on the B method [3], alongside a concise summary of two high-profile industrial applications of B [2]. These include the use of B in developing safety-critical components for Paris's Metro Line 14 and the Roissy airport shuttle [13, 25].

The landmark 2009 survey on the industrial application of formal methods by Woodcock et al. [182] examined 62 industrial projects employing formal methods. In all but six of them, data was collected directly from project participants, ensuring a high level of authenticity and detail. Among the eight highlighted projects selected by the authors to be presented in detail, one focuses specifically on railway signaling and train control. The others include projects on firmware, smart cards, microprocessors, Airbus, and a Dutch Storm Surge Barrier.

Over the past decade, a number of studies have explored evolving trends, experiences, and lessons related to the role of formal methods in developing safety-critical systems [44, 110, 119]. A significant portion of this literature focuses on the railway sector, particularly from academic contributors such as Fantechi et al. [86–88, 93]. From an industrial perspective, especially within the broader transportation domain, Boulanger has examined the use of formal methods and tools like the B method and SCADE [40–42].

Several additional studies have delved into the insights and challenges encountered through decades of integrating formal methods across research, education, and industrial applications—again with a strong emphasis on transportation systems [32, 52, 73, 139, 149]. It is also relevant to note a collection of broader railway-related surveys [85, 124, 143, 145, 150, 177, 185], which, while informative, do not focus on formal methods.

A comprehensive and widely cited report by Garavel and Graf [103] outlines the state-of-the-art in formal methods across academia and industry. Their report includes a curated list of 30 detailed success stories—one per year from 1982 to 2011—many of which relate directly to railway systems, in particular railway signaling systems with an emphasis on railway interlocking verification (cf., e.g., [35, 36, 78, 102, 111]).

Moreover, three recent questionnaire-based studies [18, 20, 100], conducted under the Shift2Rail initiative, collected input from both academic and industrial stakeholders. These surveys highlight the most commonly used formal methods and tools in the railway domain and identify key features considered valuable in practice.

The most recent comprehensive review, by ter Beek et al. [22], expands the discussion beyond safety-critical domains. While reaffirming the success of formal methods in areas like transportation and railways, the study also highlights their growing use in diverse sectors such as lithography, cloud security, and e-commerce. The paper presents testimonials from industry professionals around the globe—Europe, Asia, North and South America—demonstrating the widespread and varied use of formal methods in real-world industrial projects. It emphasizes that the skills foundational to formal methods, such as abstraction and rigorous reasoning, remain critical competencies in computer science. Industry voices included in the study reinforce the value of these capabilities, especially for graduates transitioning into professionals.

Only recently, the first systematic mapping study on formal methods in the railway domain was performed [99]. It focuses on railway signaling, given the long history of applications of formal methods in this field [87]. The authors retrieved and selected 328 high-quality research papers from the literature in the time span 1989–2020, and categorized them according to three different facets: *demographic and empirical*, identifying years, publication venues, and research methods used; *formal methods*, categorizing techniques, tools, and languages; and *railway*, concerning systems and development phases addressed by the research. Moreover, they performed further analyses to understand the characteristics of the studies on industrial applications, as well as the main trends of the most recent years. The results from [99] confirm that formal methods in railways is a prosperous field of research with a prominent industrial participation.

Compared to the above mentioned related work, in this paper we take a different viewpoint. In particular compared to the systematic mapping study by Ferrari and ter Beek [99], which we referred to in the previous section for facts and figures supporting our quantitative claims, we use a more narrative and historical perspective, with the aim of better helping the reader to navigate the history of formal methods in railways, railways, without ever claiming completeness, but highlighting key milestones, recurring challenges, influential projects, and the evolution of formal methods and tools over time. However, addressing the intricate temporal and contextual relations between so many studies is outside the scope of this paper. Instead, as mentioned in the Introduction, in this paper, we focus on the B method (cf. Section 4) and on model checking interlocking systems (cf. Section 5), which we consider among the most representative and which have been extensively treated in the literature. We believe this approach provides a contextualized understanding of how formal methods have been adopted, adapted, and validated in the railway domain, offering insights that complement more data-driven or taxonomical analyses. Furthermore, the B method is strongly based on theorem proving for consistency and invariant verification, hence addressing the B ecosystem allows us to touch upon this prominent formal verification technique as well. Moreover, the main reported applications of the B method regard the ATC/ATP/ATO railway subsystems mentioned in Section 3, which can thus be considered as representative also from this point of view. In a very recent survey [101], Fokkink provides a comprehensive overview of the employment of formal methods and tools for developing and analyzing interlocking systems and how this has progressed over the last decades, including recent developments regarding satellite-based moving block technologies, as well as directions for future research.

Finally, the historical narrative of this paper focuses on European developments, accessible in scientific publications in the English language, since the theory and practice of formal methods has mainly been developed in Europe. Moreover, Europe has pioneered issuing and enforcing standards regarding railway systems, and this has provided European companies with a competitive advantage

in procurements all over the world. We refer the reader to the surveys cited above for non-European examples and in particular to [188, Section 5: Industry Contributions and Applications] for a paragraph on industrial contributions and applications of formal methods and tools in railway in China.

3 Railway Systems Subject to Formal Methods

The challenges posed by the metamorphosis of railway transportation, especially in Europe, due to economic, political, and logistic drivers like the liberalization of the access to the European railway infrastructure, the consequent distinction between infrastructure and operation, high-speed trains, European interoperability, etc., have had a significant impact. This has been counterbalanced by a growing adoption of innovative signaling equipment (most notably the ERTMS/ETCS European standard for railway control and management, cf. Section 3.1) and monitoring systems (like on board and wayside diagnosis systems). The evolution of railway signaling systems has seen railways moving from a protected market based on national railway companies and national manufacturers to an open market based on international standards for interoperability. In this context, systems of systems are providing more and more complex automated operation, but maintaining—and even strengthening—highly demanding safety standards. Basically all of the devices involved include software, which in the end makes up the major part of their design costs; the malleability of software is paramount for the innovation of solutions. On the other hand, software is notoriously often plagued by bugs that may threaten its correct functioning. An important question is thus how the high safety standards assumed as normal practice in railway operation can be compatible with such threats?

The employment of very stable technology and the quest for the highest possible guarantees have been key aspects for the adoption of computer-controlled equipment in railway applications. Formal specification, verification, and proof have therefore been seen as a necessity.

The early attention towards formal methods by the railway sector is witnessed by the coming into force back in 2001 of the first edition of EN 50128, the European guidelines for the development of safe software in signaling equipment [80]. These were the first guidelines to mention formal methods among the mandatory techniques for the highest Safety Integrity Level (SIL) software. A recommendation that was confirmed in the second edition from 2011, where model checking was explicitly mentioned among the most relevant formal methods. EN 50716, the successor of EN 50128, extended to rolling stock, which has recently come into force [83], confirms these recommendations about formal methods, with a stronger focus on modeling (either formal or semi-formal) techniques according to the emerging industrial trend of model-based design.

The designation of formal methods as mandatory techniques in EN 50128 has further strengthened the existing industrial interest in formal methods within the railway sector, in particular to address the safety-critical signaling systems. To facilitate the discussion in the next sections, in this section we provide the following rough classification (in two large classes) of the main safety-critical railway signaling equipment, excluding only a few future innovations:

(1) train movement and distancing control systems, including three subsystems:[3]
   ATC – Automatic Train Control
   ATP – Automatic Train Protection
   ATO – Automatic Train Operation
(2) **IXL** – Interlocking (Input and eXit Locking)

Figure 1 provides a broad outline of the purpose of the two classes. On the left, it showcases a train control system based on the communication between trains and a central controller—the Radio Block Center (RBC), according to ETCS terminology. On the right, it showcases an IXL system controlling the routing of trains inside a station. The drawing also hints at currently researched innovations, such as the use of GNSS-based satellite positioning of trains [19, 145].

[3]Terminology and taxonomy may vary in the literature on railway signaling; we assume this distinction to simplify the presentation.

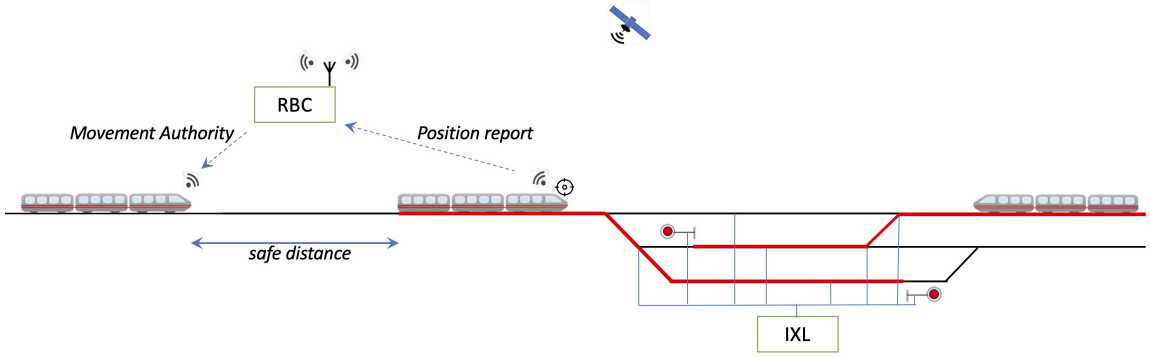

Fig. 1. Main classes of railway signaling systems, adapted from [23]

3.1 Automatic Train Control and Related Subsystems

ATC subsystems are complex *systems of systems*, made of distributed equipment located on the ground (a.k.a. wayside or trackside) and on board the trains. The main objective of the on-board ATC subsystem is to elaborate and apply the 'dynamic speed profile' (a.k.a. *braking curve*) to control the maximum train speed and automatically brake in case of need (i.e., in case of a risk of collision). To this aim, the on-board ATC subsystem receives the necessary information on the allowed maximum speed and on the status of the line from trackside subsystems. In current ATC subsystems, trains typically receive Movement Authority (MA) messages via radio from a monitoring centre that computes related information based on knowledge of the position of the trains along the line. The safety-critical enforcement of emergency braking is also called Automatic Train Protection (ATP).

Modern driverless ATC subsystems have Automatic Train Operation (ATO) functionalities, often used in metro railways, allowing the train to automatically accelerate and decelerate to respect the speed profile and even stop in stations for passenger service whenever required. ATC subsystems may also feature auxiliary control functionalities (e.g. train integrity check, or control of pantographs[4]).

ATC/ATP/ATO subsystems typically refer to international standards to ensure interoperability between the different subsystems described. Standard frameworks include ERTMS/ETCS (European Rail Traffic Management System/European Train Control System), its Chinese counterpart CTCS (Chinese Train Control System), both focusing on interoperability for passenger, high speed and freight lines, and CBTC (Communication-Based Train Control) systems, mainly aimed at the automatic operation of high-capacity metro lines.

The main characteristic of CBTC, shared with the ERTMS/ETCS Level 3 standard[5], is the concept of *moving block* signaling (cf. Section 6). In a nutshell, it consists of computing the safety distance between trains by considering the exact position of each train rather than considering the segment of the line occupied by the train as its position. The wayside ATP for CBTC systems is typically called Zone Controller.

3.2 Interlocking Systems

A railway *interlocking* (IXL) system is responsible for safely guiding trains through a network composed of track devices such as switches and crossings, ensuring exclusive access to the requested routes. Once a route is set for a train, all movable components involved are locked into position, and a proceed signal is given. After the train has passed, the section is released and made available for subsequent use by other trains.

[4]A pantograph is the device mounted on top of electric trains that collects power from the overhead catenary wires.
[5]The ERTMS/ETCS standards traditionally distinguished four levels (L0–L3) of operation of signaling systems, depending on the role of trackside equipment and on the way information is transmitted to and from trains (cf. https://www.era.europa.eu/domains/infrastructure/european-rail-traffic-management-system-ertms_en). L0 denotes no ETCS equipment, L1 and L2 are currently in operation. In 2023, key features of L3, such as moving block, which are currently in a trial phase, have been merged into L2 with CCS TSI 2023 (cf. https://www.ertms.net/faqs/ and https://eur-lex.europa.eu/eli/reg_impl/2023/1695/oj/eng).

An interlocking system is designed to ensure that a signal to proceed cannot be displayed unless the corresponding route has been proven safe. For example, it ensures that a signal indicating a diverging route cannot be activated unless the related switches have been properly aligned first.

3.2.1 Technical Development. Historically, the first interlocking systems were mechanical (from 1843) until they evolved over electromechanical systems to relay-based systems (since 1929), and finally to computer-based systems (since 1978) [127]. Today there are still some relay-based systems, but more and more of them are computer-based. This development reflects both advancements in technology and a growing emphasis on dependable, advanced safety solutions.

3.2.2 Major Design Paradigms. Interlocking systems have been designed according to different logical paradigms [155, 160]. Two of the most widely used are (1) *route-based/functional/tabular interlocking* and (2) *geographical interlocking*. The former is based on predefined routes through the rail network and uses an interlocking table specifying safety conflicts between different routes as well as point positions and signal states to be enforced before a route may be entered by a train. For the latter, routes through the railway network are not predefined, but can be allocated dynamically.

For the computer-based interlocking systems, the software is usually designed for re-use, so that novel interlocking software designs can be realized by instantiating a generic application with configuration data. For the route-based interlocking systems, interlocking tables are used for configuration data, while for geographical interlocking systems the track layout is used [160].

4 A History of the B Method Applied in the Railway Domain

The B method [1] is a formal method for the specification, design, and verification of software systems, particularly in safety-critical domains. It is rooted in mathematical logic and emphasizes refinement, proof obligations, and tool-supported development. Over 30 years, B has seen widespread industrial application [52], especially in the railway sector and for ATP functionalities. Its industrial adoption has been supported by continuous tool development, methodological refinement, and regulatory alignment, particularly with the EN50128 standard. In fact, as mentioned in Section 2, according to the systematic mapping study on formal methods in railways [99], the B method and its supporting tools are the most dominant formal methods and tools, in particular among the most recent and the industrial studies.

This section is structured as a synthesis of the major application areas of the B method, following its historical and technological progression, namely:

- Classical B for Software Development – focusing on the refinement-based generation of correct-by-construction safety-critical software.
- Event-B for System Modeling – extending B to early-phase system-level reasoning and specification across interacting components.
- Data Validation with B – the most recent and industrially successful application of B for verifying large-scale configuration datasets.

4.1 Classical B for Software Development

The B method, introduced by Jean-Raymond Abrial in the early 1990s as a successor to Z, was conceived to support software development through refinement and proof. Early adoption emerged within the Paris railway network, notably with RATP (the public company in charge of the ground transportation in and around Paris) and Alstom during the SACEM (embedded, automatic speed train protection system for rapid transit railways, first deployed on the RER Line A suburban railway in Paris in 1989) project [112]. The success of this pioneering effort led to the development of the driverless Line 14 of the Paris metro [25], a hallmark for B's industrial applicability.

The maturation of the method was paired with tool development. Initially, Alstom collaborated with Abrial to create an internal toolset (type checker, proof obligation generator, theorem prover), which evolved into Atelier B, industrialized by Digilog (now

CLEARSY) [2]. Atelier B is the industrial tool that allows for the operational use of the B method to develop defect-free proven software. This ecosystem—comprising RATP, CLEARSY, academia, and railway companies—enabled large-scale adoption and training [49].

Methodological questions emerged regarding integration into conventional development lifecycles, documentation, and safety assessment, particularly given the strict requirements of EN 50128. These were addressed by practice and codified through internal user guides and training programs.

Driverless metro systems, and specifically ATP subsystems, became prime showcases for the industrial application of the B method. The most emblematic example is Paris Métro Line 14 [25], which was the first fully automated line in the RATP network. The safety-critical software developed using B covered core functions of an ATP system, including train localization, speed supervision, movement authority management, automatic braking, and interface control with interlocking and platform screen doors. The formal development resulted in over 115,000 lines of B code and the generation of 27,800 proof obligations, demonstrating the method's scalability. Impressively, more than 90% of these were discharged automatically, and only a small fraction required manual proof (8.1%). A similar success was achieved with the Roissy Airport Shuttle (CDGVAL) [13], also using B for its ATP functions, where even larger models were built: 183,000 lines of code and 43,610 proof obligations, with a lower manual proof effort (3.3%). In both cases, the B models served as the single source of truth, from which Ada code was generated automatically using certified code generators. No unit testing was performed on the safety-critical components; instead, formal proofs substituted low-level testing, and only integration and system-level validation tests were carried out. These projects showed that rigorous formal proof could provide a sufficient basis for safety assurance in place of traditional test campaigns [135, 139]. The B method further expanded into international products like Alstom's URBALIS 400 (a CBTC signaling system designed for metro and urban rail networks), now deployed on over 100 lines worldwide.

The Canarsie Line [79] of the New York City Subway marked a major international deployment of B-based formal methods and the first full implementation of CBTC in the United States. The system was developed by Siemens Transportation and was derived from the formal development framework previously used for Paris Line 14. However, the Canarsie project presented greater system complexity, particularly due to mixed traffic conditions, where CBTC-equipped and non-equipped trains shared the same tracks.

The safety-critical software, developed entirely using the B method, included essential CBTC functions such as train localization using trackside beacons and odometry, dynamic movement authority calculation, speed monitoring and braking curve enforcement, safe management of interlocking interfaces, door control logic for platform screen doors, and fail-safe fallback modes and degraded operation handling. The application of B enabled the formal specification, refinement, and verification of these components. Semi-automated refinement tools, such as BART [48] (a rule-based refinement tool that generates B machine implementations and refinement components from specifications, now integrated into Atelier B), were heavily used to streamline the transition from abstract specifications to implementable models, significantly improving development productivity. These tools supported automatic generation of low-level B models (B0, the implementable subset of B that enforces deterministic, translatable constructs suitable for automatic code generation into programming languages), from which Ada code was generated. Despite the complexity of the system, the use of the B method helped Siemens manage the verification burden and obtain the necessary safety assurances. Proof obligations were automatically generated and discharged to demonstrate critical properties such as collision avoidance, speed limit compliance, and correct response to command and control signals. An important conclusion drawn by the Siemens team was that, although the formal method was initially perceived as an additional burden, the overall cost of development, from specification to validation and certification, was lower than that of conventional software development methods. The formalism not only reduced the number of late-stage errors, but also improved system maintainability and clarity.

The B method was also used for embedded systems and platform screen doors controllers using the CLEARSY Safety Platform, enabling development up to SIL 4 with redundant proof and execution layers [134]. Despite its success, recent industrial use of classical B outside the railway sector has declined. Key challenges include human expertise requirements and the cost of exhaustive proof obligations, though advances like automatic refinement tools mitigate some issues [136].

Beyond software, B has been used to generate safety-critical code for hardware platforms. CLEARSY's Platform Screen Door controllers and its SIL 4-certified CLEARSY Safety Platform [138] exemplify this transition. The platform integrates proof-based development, automatic code generation, and safety-assured execution, reinforcing B's applicability to embedded systems.

4.2 Event-B for System Modeling

While classical B focuses on the formal refinement of software specifications into executable code, Event-B extends the method to the system level, enabling the modeling of entire systems, their components, and their interactions [4].

This extension addresses a critical limitation of software-only approaches: software requirements often depend on assumptions about the operational environment, including interactions with hardware, other software modules, and human operators. If these environmental assumptions are incorrect or incomplete, the final system may malfunction despite being provably correct at the software level. Event-B emerged from the convergence of industrial needs and academic research, notably within several collaborative EU projects dedicated to its development and industrialization. The notation evolved from B, introducing constructs suitable for modeling complex, distributed systems while retaining the mathematical rigor of proof-based verification. The approach [166] supports early-phase system validation, ensuring that high-level architectural and behavioral properties are correct before detailed design and implementation.

A prominent application was conducted by CLEARSY for New York City Transit (NYCT) in two major CBTC projects [167]. In the first project, concerning the Flushing Line, CLEARSY developed formal Event-B models capturing train movements, switching logic, and speed enforcement. These models incorporated key safety properties—such as collision avoidance, prevention of traversing unlocked switches (to avoid derailment), and overspeed protection, which were formally specified and proved. The modeling process identified critical assumptions about subsystem behavior and interface contracts, ensuring these were made explicit and communicated to stakeholders.

The second NYCT project involved a different interlocking implementation, yet CLEARSY was able to reuse significant portions of the original system-level models thanks to their abstraction from implementation details. This reuse substantially reduced the safety analysis effort in the second deployment, illustrating Event-B's potential for model portability and cost efficiency across projects.

Event-B's benefits were also demonstrated in the validation by CLEARSY of the Octys CBTC system for RATP [65]. For each safety property, an informal reasoning step preceded formalization, framing the modeling approach. The transition from informal to formal reasoning often revealed gaps or implicit assumptions in the safety arguments, enabling their correction early in the process. Formal proofs further allowed the identification of minimal assumption sets needed to guarantee the properties, thus improving the precision and completeness of the safety requirements.

Another example concerns Alstom's URBALIS 400 Zone Controller [66], where—in a collaboration between CLEARSY, Alstom France, and Universität Düsseldorf—the classical B development ensured conformance of the software to its specification, but could not guarantee the algorithmic correctness with respect to system-level safety goals. Event-B models were constructed to bridge this gap, linking environmental variables (e.g., actual train positions) to software variables (e.g., calculated protection envelopes). Tools such as Atelier B and ProB were used to verify safety properties essential to tuning or extending the algorithms without compromising safety.

Event-B has also supported graphical modeling approaches to improve communication with domain experts. Thales Austria GmbH and researchers from the University of Southampton combined Event-B with UML-B and applied it to the RailGround interlocking system specification [53] (developed by Thales Austria GmbH). The graphical representation proved easier for signaling engineers to understand than textual models, enabling direct expert feedback and improved validation.

This approach was also applied to novel concepts such as ETCS Hybrid Level 3 (HL3)[6] [114], aiming to increase train throughput without additional track infrastructure. In this cooperation, Thales Deutschland GmbH and Universität Düsseldorf, with support by CLEARSY, used Event-B and ProB to develop a reference model of HL3 principles, including degraded modes and edge cases. The model was executed in real-time with actual trains in a field demonstrator, serving as a reference implementation during validation.

Finally, Event-B has been integrated into model-based systems engineering workflows [38, 43, 168] for standardizing railway signaling interfaces within the EULYNX and Shift2Rail initiatives. EULYNX is a European initiative in railway signaling that aims to standardize interfaces and elements of signaling systems in the railway industry, while Shift2Rail is the first European rail initiative to seek focused research and innovation and market-driven solutions (cf. Section 7). In these contexts, the Systems Modeling Language (SysML) is often used to capture system architecture; however, as a semi-formal language, SysML lacks formal proof capability. By

[6] After the merge of L3 into L2 (cf. Footnote 5), the concept known as ETCS Hybrid Level 3 (HL3) is referred to as ETCS Hybrid Train Detection (HTD).

translating SysML specifications into UML-B and Event-B, formal verification could be performed, enabling the systematic validation of interlocking logic and ensuring compliance with safety standards at an early design stage.

4.3 Data Validation with B

An important and increasingly widespread industrial application of the B method is data validation [137], a practice that has grown rapidly over the past decade. Many safety-critical railway systems—such as signaling, CBTC, and IXL, rely on configuration data that is tailored to the specific characteristics of each deployment. These parameters may encode track layouts, permissible speeds, braking curves, route definitions, or interlocking tables. Given their direct impact on operational safety, the correctness of this configuration data is essential: any inconsistency or omission can lead to unsafe system behavior, even if the underlying software is correct.

The B language has proven particularly well-suited for this task. Its ability to express precise properties, invariants, and constraints—combined with automated proof and model checking—allows for formal verification of configuration datasets against safety and operational requirements. This approach replaces rigid, custom-built validation programs, which were costly to maintain and update, with flexible, property-driven verification engines where a change in requirements can be accommodated simply by modifying the formal specification.

One of the earliest dedicated tools in this domain was OVADO, initially developed by CLEARSY for RATP. OVADO is a formal data validation tool for railway systems configuration which uses the B method to automatically check that configuration data meets safety requirements. Before OVADO, RATP relied on ad-hoc, hard-coded validators that required extensive redevelopment for every new line or system modification. OVADO provided a generic framework in which properties were written in B, enabling rapid adaptation and reuse across projects. In parallel, Siemens developed RDV, a similar tool built on top of the ProB model checker and constraint solver [84].

ProB played a transformative role in this field [115]. Originally designed for specification validation, it was extended to handle large datasets and complex constraints, providing automated detection of inconsistencies that often escaped manual verification. Siemens' deployment of ProB for data validation on Paris Metro Lines 1 and 14, the Paris Charles De Gaulle airport automated shuttle (CDGVAL), São Paulo Line 4, and Barcelona Line 9 revealed at least one real defect missed by manual review (during a first test with intentionally corrupted data, four errors were uncovered whereas three were expected), underlining the value of automated formal analysis. Before the adoption of ProB, Siemens relied on Atelier B with customized proof rules for large data sets, but this approach did not scale well for highly parameterized systems, making automation critical.

CLEARSY and its partners have developed a family of domain-specific data validation tools, tailored for different customers and contexts:

- DTVT, for Alstom's URBALIS 400 CBTC system, used in deployments such as Mexico City, Toronto, São Paulo, and Panama.
- Caval, also called CLEARSY Data Solver, is CLEARSY's most recent generic validation platform, incorporating lessons learned from earlier tools and achieving EN 50128 certification (Class T2) in 2019.
- Dave, for General Electric (GE) signaling products.
- Rubin, used by Thales to verify engineering rules for ETCS RBCs.

In several cases, these tools implement a dual-chain verification architecture, in which a primary tool performs the validation and a secondary, independent tool re-checks the results. This redundancy further reduces the risk of tool-induced errors.

The railway sector's safety standards, particularly EN 50128, classify data validation tools as Class T2, meaning that a tool "supports the test or verification of the design or executable code, where errors in the tool can fail to reveal but cannot directly create errors in the executable software." Compliance requires rigorous specification of the tool's purpose, a well-defined verification methodology, and extensive testing. OVADO, DTVT, and Caval all meet these requirements, with some undergoing certification by multiple independent bodies for use in international projects.

Data validation with B has now become a standard practice in railway projects worldwide, covering metro, mainline, and high-speed systems. Its industrial success can be attributed to the following three factors.

- Scalability: ProB's constraint solving enables the efficient handling of very large data sets.

- Flexibility: new or modified operational rules can be incorporated simply by editing the B property set, without altering the validation engine.
- Safety assurance: the combination of mathematical proof and exhaustive checking ensures a high degree of confidence in the correctness of deployed configurations.

As a result, data validation using B has evolved from a niche application into a core element of the safety assurance process for modern railway signaling systems. It complements both classical B and Event-B, ensuring that the software and system-level proofs are not undermined by incorrect operational parameters.

5 A History of Formal Verification of Interlocking Systems by Model Checking

Conventional development and verification of interlocking systems has been informal and mostly manual, as described in [121], and therefore inefficient and error-prone. That is quite problematic as interlocking systems are safety-critical systems for which the correctness is vital due to the potential for catastrophic failures. Therefore, with the introduction of formal methods, the research community saw an opportunity to overcome these problems and they started in the late 1980s and the 1990s to research the use of formal methods for the specification and verification of interlocking systems [99]. As mentioned in Section 1, the two core formal (verification) methods are model checking and theorem proving. The latter has been addressed in Section 4. In this section, we address model checking by providing a (non-exhaustive) historical account of the application of formal verification by model checking to interlocking systems. As mentioned in Section 2, according to the systematic mapping study on formal methods in railways [99], model checking is the most common formal verification technique, used in roughly half of the studies, while almost half of the studies focus on interlocking systems.

5.1 Early Applications of Model Checking for Interlocking Systems

Model checking [63] is a technique that, given a finite-state model of a system and a property in some appropriate logical formalism (typically a temporal logic), automatically checks the validity of the given property for the given model. As this technique, in contrast to theorem proving, is fully automated, it has been considered quite attractive in a variety of domains, especially for industrial use.

In the 1990s, formal verification of interlocking systems by model checking started. The first papers on the applications of model checking include [173], published in 1990, and [9, 111, 116], all published in 1995. Of these, [9] and [116] used the AMC model checker and the SMV model checker, respectively, whereas [111, 173] did not use a traditional model checker, but a prover based on Stålmarck's method (a SAT-style propositional logic solver) [172].

These early applications addressed only interlocking systems for small (fractions of) railways networks. The models expressed the dynamic behavior of the systems and the properties to be verified were typically at design level, expressing signaling principles in terms of the states of entities like signals, points, and routes (e.g., that two that opposing signals are not open (green) at the same time). Only later high-level (design-independent) safety requirements expressing the absence of train collisions and derailments were used as properties.

In the second half of the 1990s, several other applications followed, including an industrial application of the SPIN model checker to model and formally verify a rather complex part of a railway interlocking system [60].

5.2 Further Development

In the following years, from the late 1990s onward, a lot of research was dedicated to how to improve model checking of interlocking systems and industrial adoption started to increase. Some of the major research questions were how to ease the formal modeling and how to make the verification scalable such that large interlocking systems of industrial size could be verified. In Sections 5.2.1 and 5.2.2, we describe some efforts that were made in regard to these two questions, while Section 5.2.3 considers industrial adoption.

5.2.1 Dedicated Tools and Domain-Specific Languages. In the late 1990s and early 2000s, researchers and companies started to develop dedicated domain-specific languages, tools, and methodologies for model checking interlocking systems. We now describe these ideas that became trend setting, both in academia and in industry.

It was recognized that the creation of formal models and properties for a product line of interlocking systems could be automated and hidden for the user, such that formal models and properties should no longer be made by hand. The idea was as follows: for a given product line, software developers should provide (1) a domain-specific language for describing interlocking systems, using terms and concepts from the railway domain (e.g., track layouts and interlocking tables[7] as already used by railway engineers) and (2) tools for generating verifiable, formal models and properties from specifications in the domain-specific language. Hence, for each system to be verified, the railway specialists should (i) specify application-specific parameters in the domain-specific language, (ii) apply the generators to the specification to automatically generate a model and desired properties, and (iii) apply a model checker to the output of step (ii).

Following this idea, for relay-based interlocking systems several tools were produced to derive models from relay circuit diagrams used by railway engineers for specifying/documenting relay interlocking systems (cf., e.g., [8, 119]). Similarly, tools were produced to derive formal properties: high-level properties of no collision and no derailments from the track layouts, and design-level properties from the interlocking tables (cf., e.g., [117, 118]). The purpose of the subsequent model-checking step is to verify safety and correctness of the circuit design.

Also for computer-based interlocking systems the idea of using dedicated tools and domain-specific languages, such as OnTrack and SafeCap, was proposed (cf., e.g., [126, 128, 130, 158, 181]) and adopted by companies like Prover Technology and Systerel. In these cases the domain-specific languages typically support descriptions of track layouts and for route-based systems also interlocking tables, and the model and property generators instantiate a generic (configurable) model and generic properties with configuration data derived from the domain-specific descriptions. (As previously mentioned, interlocking systems are typically configurable and consequently the models can be so too.)

The above approach to model checking of computer-based interlocking systems checks the correctness of (the model of the dynamic behavior of) a concrete instance of a generic interlocking system and can be used to catch errors in configuration data as well as in the generic application.

5.2.2 Addressing Scalability. In the 1990s, 2000s, and 2010s, a significant challenge for the scalability of (global) model checking was encountered: the state-space explosion problem, due to the number of possible states in the system growing exponentially with its size and complexity. For instance, in [95], a systematic study of applicability bounds of the symbolic model checker NuSMV and the explicit model checker SPIN showed that these model checkers could only verify interlocking systems for small railway networks. Since then, many research groups have investigated techniques to overcome this scalability problem pushing the applicability bounds toward industrial size.

One of the most frequently used techniques has been to *abstract* certain aspects (data or behavior) of the system to reduce the size of the model, while also ensuring that the abstraction preserves the correctness of the model checking results. The choice of abstraction technique usually depends on the specific characteristics of the system. Many domain-specific abstraction techniques have been proposed [33, 95, 129, 180], usually exploring the fact that interlocking systems typically exhibit a high degree of *locality* such that a property to be verified only depends on a part (the so-called *cone of influence*) of the model.

A related technique is *compositional verification* as proposed for interlocking systems in [120, 142]. Here the idea is to decompose the network under control into several sub-networks and then apply model checking for the interlocking restricted to each of the sub-networks.

Yet another way to tackle the problem has been to improve the verification engines of model-checking tools. Here, a promising approach has been to use *bounded model checking* utilizing powerful SAT/SMT solvers. When bounded model checking is combined with induction reasoning (in *k-induction*), it is possible to verify safety properties without having to explore the whole state space (cf., e.g., [126, 159]). These ideas have also been adopted in commercial tools, for instance by Systerel [46] and by Prover Technology [37].

Finally, as anticipated in the Introduction, also SMC has been applied to tackle scalability. Compared to traditional model checking, which produces results that are exhaustive, i.e., if there exists any execution that violates the property this will lead to a counterexample, SMC uses a simulation- and sample-based approach to reason about precise properties specified in a stochastic temporal logic, offering

[7] An *interlocking table* is a table that specifies the train routes in the network under control and for each of these routes the conditions for allocating them.

a scalability advantage over exhaustive (or probabilistic) model checking due to the fact that there is no need to analyze the entire state space. Moreover, even though the outputs of sample-based methods are not always correct, statistical inference enables quantifying the confidence in the obtained result, thus trading the lack of exact results (100% confidence) for improved scalability [6, 140].

5.2.3 Industrialization and Standardization. During the 1990s, the first pilot studies and academic/industry collaborations on model checking interlockings took place (cf., e.g., [9, 60], as mentioned in Section 5.1) and industrial adoption started. During this period, several commercial tools developed by Industrilogik/Prover Technology were used for interlocking verification [36, 173]. These tools were not based on traditional model checking, but were based on Stålmarck's method (which can be seen as a natural deduction proof system with a novel proof-theoretic notion of proof depth) that inspired the later use of bounded model checking.

With the start of the new century, bounded model checking based on induction and SAT solving appeared, allowing larger stations to be verified, and Prover Technology readily applied this technology in their interlocking verification tools, allowing a significant number of real systems to be verified [37]. At the start of the new century, other companies such as CLEARSY and Systerel, specialized in formal method solutions (including model checking) for railways, were established and these have since then supported railway infrastructure managers and suppliers.

Since then, railway companies such as RATP, SNCF, Alstom, and Siemens have adopted the use of model checking for interlocking systems on a routine base, notably using ProB [141], not publicly available (S3)[8] [46, 156], and Prover iLock Verifier[9].

The ProB model checker has been used for validating configuration data for interlocking systems. For instance, dedicated tools built on top of ProB have been developed by CLEARSY and applied for data validation[10] of interlocking systems for customers like Atkins and SNCF.

S3 has been used for safety verification of interlocking systems (e.g., it has been applied by Alstom to verify several Smartlock 400 GP interlocking applications on multiple subway lines, like Amsterdam, Lusail, Guadalajara, etc.) [156]. Lately, S3 has been used for safety verification on the following industrial interlockings: the ESTER project for Trafikverket (Sweden) with Ansaldo (2018–2020), the PHPI family (a kind of relay-based IXL) at RATP (2024–2026), and on the new interlockings based on the AMS (Atelier Métier Signalisation, a signaling engineering framework used at SNCF to develop their newest generation of IXL) ARGOS project for SNCF with interlockings from both Hitachi and Thales (started in 2023, still ongoing in 2026).

Prover iLock Verifier has been used for safety verification of a significant number of interlocking systems around the world[11], e.g., for Bankverket (Sweden), Stockholm Metro (Sweden), Bane Nor (Norway), Network Rail (UK), New York City Transit (US), and Canadian Pacific. RATP and Prover Technology collaborated in the late 2000s on creating a formal verification solution called Prover Certifier[12] (using the Prover PSL model checker[13]) to meet the RATP demand for safety verification of interlocking software[14] [34]. The solution was first evaluated in parallel with a traditional testing effort on the interlocking systems for the Paris metro line 3bis in 2009. After that, the solution entered production use for interlocking systems on the Paris metro lines 1, 4, 8, and 12, as well as on other interlocking systems.

In the 1990s, work on a standard (CENELEC EN 50128) for the development of software for railway control and protection started and the first version was ready in 2001 and was later updated [80]. The 2023 standard EN 50716 supersedes EN 50128 (incorporating its content) as the current CENELEC software standard for railway applications in line with current industry practices and technologies, including both signaling systems and rolling stock applications. EN 50128 will only effectively be withdrawn once EN 50716 is fully adopted across Europe. According to these standards, interlocking systems have the highest safety integrity level (SIL 4) and the use of formal methods including model checking is strongly recommended (cf. Section 3).

[8] https://www.systerel.fr/en/innovation/products/systerel-smart-solver/
[9] https://www.prover.com/products/prover-ilock/
[10] Descriptions of some selected CLEARSY projects using data validation can be found here: https://www.clearsy.com/en/railway/formal-methods-for-validating-parameterization-data-clearsy-is-chosen/ and https://www.clearsy.com/en/railway/clearsy-carries-out-the-formal-validation-of-data-from-sncf-reseaus-mistral-ng-program/
[11] Descriptions of selected projects using the Prover tools can be found here: https://www.prover.com/references/.
[12] https://www.prover.com/products/prover-certifier/
[13] https://www.prover.com/products/prover-psl/
[14] https://www.prover.com/references/ratp-paris/

6 Other Equipment and Formal Methods

The previous two sections illustrate how the application of formal methods has developed when addressing the two main classes of railway systems introduced in Section 3. That classification excludes some other railway equipment that nevertheless captured the interest of formal methods practitioners due to the high degree of responsibility for software to ensure safety. These include safety-critical systems that are specific subsystems of those introduced in Section 3 (such as level crossing control or platform screen doors), as well as systems focused on traffic management and supervision, or advances in train control policies.

In this section, we briefly mention some of these systems, in particular those to which formal methods have been applied successfully. This illustrates the diversity of railway systems that have benefited from the application of formal methods. To begin with, we briefly mention some success stories involving formal methods different from B.

6.1 Formal Methods and Tools Used in the Railway Domain

Specific success stories with formal methods other than B worth mentioning include the aforementioned development of Rio de Janeiro's metro control system [94] supported by Simulink/Stateflow and by static analysis via Abstract Interpretation; the verification of ERTMS/ETCS [58] by means of NuSMV and that of ERTMS/ETCS HL3 with a variety of formal methods and tools [19, 51] (cf. Section 6.5); the modeling of the MA scenario of CTCS Level 3 and its verification, reported in [7], with an interactive theorem prover based on Isabelle/HOL; and—more recently—the analysis of an Autonomous Positioning System (APS) for Florence tramways [17], carried out with the support of UPPAAL SMC. This is just a glimpse, we refer to [96, 97, 99, 147] for comparisons of different formal methods and tools for railway system design. In particular, in [96, Fig. 1: Evaluation table] the authors present a systematic evaluation of a set of 13 formal methods tools, including SPIN, nuXmv, ProB, Atelier B, UPPAAL, FDR4, CADP, mCRL2, TLA+, and UMC, against a set of 33 features, grouped into functional features (e.g., type of formal verification), language expressiveness (e.g., non-functional aspects and modularity), and quality features (e.g., tool flexibility, maturity, and CENELEC certification).[15]

Moreover, not all industrial success stories are published, on the contrary. We mention here the SIL 4-certified SafeProver model checker (developed by SafeRiver, now part of SERMA).[16] It uses bounded model checking for reachability analysis as well as algorithms derived from the k-Induction principle for invariant satisfaction analysis. SafeProver offers the possibility to formally analyze discrete-time bounded systems in an exhaustive manner, but both system and properties to be verified must be written in ICL, a proprietary input language. Success stories include the formal verification of a CBTC system for the former GE Transportation and a SACEM speed control system for RATP. SafeProver has recently been extended to prove C language properties on C language models and tested on industrial-size source code generated from Simulink models. SafeProver for C is currently under industrialization and will be available for further industrial experiments during 2026.

6.2 Level Crossings

Level crossing control has been used as a case study in several studies on the use of formal methods in railways, due to the high safety concerns generated by the intersection of road and rail traffic. Two notable examples can be found in [148], where timed automata are used for the specification of a level crossing protection system and validation is conducted by model checking with UPPAAL, and in [108], where a novel design of a level crossing protection system is modeled with timed Petri nets in order to elaborate a control strategy suitable for the ERTMS Level 2/3 operation context. These two examples show how different modeling techniques can be applied to similar systems when aiming at different purposes (validation vs. control strategy definition).

6.3 Platform Screen Doors

In automated metros, typically operating in a closed environment, platform screen doors are adopted to prevent passengers from falling on to the tracks. A platform screen door controller has the responsibility of opening only when the train's doors are perfectly aligned with the platform doors. It is worth mentioning that the software of several such installations around the world was developed

[15] Subsequent to [96, Fig. 1], in 2024, Atelier B has been certified against CENELEC EN 50128 and EN 50129: https://www.clearsy.com/en/tools/atelier-b/.
[16] https://www.serma-safety-security.com/en/formal-methods/

with the help of the B method [134], which guarantees a correct-by-construction behavior of the system in absence of failure of its components.

6.4 Automatic Train Supervision (ATS)

ATS systems are meant for the supervision of all those high-level monitoring, track optimization and maintenance operations of the railway system that are not addressed by the other subsystems. While many tasks related to ATS systems (e.g., remote route lock/unlock command) cannot be considered safety critical, since other subsystems will provide the necessary protection against hazards, there are situations in which a certain level of criticality is assigned to those systems as well. An ATS system typically acts by issuing route requests to an IXL system; doing so, it can easily incur in deadlocks due to the constraints posed by the IXL. Deadlock avoidance can be tackled by model checking, as shown in [146].

6.5 Moving Block

The ERTMS/ETCS standard considers different levels of operation for compliant ATC systems. In the most advanced one, ERTMS/ETCS Level 3, there are no track occupancy sensors and it is the responsibility of an on-board odometry system to keep track of the train's position, as well as to compute the current train speed. The on-board computer of each train periodically sends to the RBC a position report and the results of a train integrity check. In turn, the RBC sends the MA back to each train. The MA is computed by considering the minimum safe rear end of the foregoing train (*moving block* signaling), further improving a line's throughput and reducing maintenance costs.

The absence of track circuits as safe train detection and localization mechanism, and the difficulty of computing the exact train position, have so far prevented the actual deployment of ETCS Level 3 systems, due to safety concerns. Nevertheless, ETCS Level 3 is currently the most promising level of operation in terms of safety increase, capacity gains and maintenance cost reduction. As such it provides a challenging case study; in particular, there is a rich literature on the application of a variety of formal methods and tools, including UML-B, mCRL2, SPIN and UPPAAL, to a downsized version named ERTMS/ETCS HL3 [5, 10, 16, 19, 51, 69, 74, 114, 144, 176].

6.6 Satellite Positioning

The localization of trains along a line is currently detected by specific trackside sensors (such as track circuits or axle counters) that are able to detect the occupancy of a track section. More precise computation of the current position of a train, required by moving block signaling systems, can be achieved by on-board odometry, accelerometers and other sensors. Satellite positioning promises to become an absolute positioning system, significantly reducing the need and cost of trackside sensing equipment (think also of its maintenance). The statistical nature of positioning information given by GNSS sensors[17] requires a paradigm shift from qualitative formal verification of safety towards quantitative evaluation aimed at the validation of probabilistic safety requirements. In this regard, UPPAAL's statistical model-checking features were used in [122, 123] for the evaluation of GNSS localization in the context of ETCS Level 3. The choice of UPPAAL SMC was also followed for the aforementioned safety verification of the satellite-based APS of the Florence tramways in [17].

7 International Projects and Initiatives on Formal Methods and Railways

The steady advancement of the use of formal methods in the railway sector over the past decades has been accompanied and supported by numerous initiatives aimed at fostering their adoption and refinement. With no claim to completeness, we briefly describe some international projects and initiatives spanning from the end of the previous century to the present days. The selection is based on the knowledge of the authors, who have been in the field for over 30 years, and on the survey of projects carried out by part of the authors within the ASTRail EU project [100]. While researchers' bias is unavoidable, given the absence of a specific search engine for projects, we believe that the current list provides a fairly representative picture and historical perspective.

[17] A GNSS receiver provides the coordinates of its position together with a *protection level* (PL), a statistical bound error that guarantees that the probability of the real position error exceeding PL is smaller than or equal to a target value called *integrity risk* (IR).

FMERail (1998–1999): This was an EU ESPRIT ESSI project (no. 26538) aimed at boosting the real-world adoption of formal methods in the railway industry. It facilitated the communication between formal-methods technology providers and the railway industry through a series of five workshops, case studies, and tutorials. The project partners were Technical University of Denmark (DTU), IFAD (Denmark), TERMA Elektronik A/S (Denmark), Formal Systems Europe Ltd. (United Kingdom), and Steria Méditerranée S.A. (France).

EuroInterlocking (1999–2006): This project of the Union Internationale des Chemins de fer (UIC) aimed at the harmonization, joint development, and standardization of IXL and signaling systems in Europe. In particular, it has contributed to the development of standardized file formats for IXL data exchange, and to the construction of a generic simulation tool (exploiting the project-defined location and IXL file formats) for the verification and validation of IXL rules.

EuRailCheck (2007–2010)[18] (European Railway Formalization and Validation): This was a project of the European Railway Agency (ERA), aiming at the development of a methodology and supporting tools for the formalization and validation of (a subset of) the ETCS specifications. Within the project, three main results were achieved: a methodology for the formalization and validation of the ETCS specifications that goes from the informal analysis of the requirements to their formalization and validation; a set of support tools, covering the various phases of the methodology; and a realistic subset of the formalized specifications.

INESS (2008–2012)[19] (INtegrated European signaling System): The main goal of this EU FP7 project was to extend and enhance the standardization process defining and developing specifications for a new generation of IXL systems. One of its tasks was to identify safety requirements of the IXL model and their representation in a formal format, as invariant state properties (using UML-B). A prototypical tool for the verification of these invariants was developed.

EULYNX (since 2014)[20] (European Initiative Linking Interlocking Subsystems): This is a long-running initiative of European Infrastructure Managers. The project aspires to a mutually shared vision toward harmonization of railway signaling systems, their technical architecture, functions and interfaces. The project includes items like system architecture, modeling and testing, data preparation, interfaces between IXL systems, interfaces to track vacancy detection and adjacent IXL or signaling subsystems: requirement management tools, UML, and SysML modeling techniques are used to formalize unambiguous requirements. EULYNX's has published standardized interfaces and architectures that will allow interoperable signaling and control components across Europe and help to reduce life-cycle costs, promote supplier competition, and accelerate the adoption of digital technologies. Academic efforts have led, e.g., to a model-based testing setup to automatically test an implementation for compliance to the EULYNX standard and a systematic method to automatically formalize EULYNX SysML models in the mCRL2 model checker [43].

ASTRail (2017–2019)[21] (SAtellite-based signaling and Automation SysTems on Railways along with Formal Method and Moving Block validation): This EU H2020 Shift2Rail project included (i) an analysis phase, dedicated to the comparison and evaluation of the main formal methods and tools that were being used at that time in the railway industry to guarantee that software design and implementation criticalities do not jeopardize the safety, as well as (ii) an application phase, in which selected formal methods were used to model and analyze two main goals addressed by the project, namely moving block distancing and automatic driving. The aim was to validate that formal methods are not only able to guarantee safety issues, but also—more in general—the long term reliability and availability of the software. The main output of the project, for what

[18] https://es.fbk.eu/index.php/projects/eurailcheck/
[19] https://cordis.europa.eu/project/id/218575/reporting
[20] https://eulynx.eu
[21] https://projects.shift2rail.org/s2r_ip2_n.aspx?p=S2R_ASTRAIL

concerns formal methods in railways, was an in-depth analysis of the applicability of several formal tools to the moving block problem [97], as well as a comprehensive guide of how to select the most appropriate tool for the case at hand [96]. The project also included a set of surveys about formal methods in railways, which have shown the relevance of usability and certification support as primary characteristics that companies expect from formal methods tools [18, 20].

X2RAIL-2 (2017–2021)[22] (Enhancing Railway Signaling Systems): This EU H2020 Shift2Rail project carried out a survey to identify the railway signaling industry's expectations of formal methods and tools, in terms of their most important characteristics, benefits and challenges. This survey [184] showed that formal methods can provide significant benefits to railway signaling system development in terms of improved safety, better requirement quality and reliability and, finally, reduced time-to-market and cost. However, the survey indicated that there are also significant obstacles increasing to the widespread use of formal methods to gain such benefits. The main obstacle is the high learning curve, and indeed formal methods have the image of being too difficult to apply for "ordinary engineers". This survey moreover showed that the use of formal methods would be helped by more standardized interfaces.

4SECURail (2019–2021)[23] (FORmal Methods and CSIRT for the RAILway sector): This EU H2020 Shift2Rail project addressed the development of a demonstrator for the use of state-of-the-art formal methods in the railway environment. The aims of the demonstrator was answering the questions raised by X2RAIL-2, that is to evaluate the learning curve and the use of standard interfaces, and to perform a cost-benefit analysis of the adoption of formal methods in the railway industry. The demonstrator has been applied to a railway signaling subsystem described using standard interfaces aimed at illustrating some usable state-of-the-art techniques for rigorous standard interface specification, as well as at supporting a cost-benefit analysis to back this strategy with sound economic arguments.

PERFORMINGRAIL (2020–2023)[24] (PERformance-based Formal modeling and Optimal tRaffic Management for movING-block RAILway signaling): This EU H2020 Shift2Rail project aimed to deliver formal modeling and optimal traffic management of a moving block railway signaling system using advanced train positioning approaches that mitigate potential hazards in the diverse market segments. It implemented a holistic system approach to address the open challenges for the moving block and *virtual coupling* concepts in terms of safe operational principles and specifications, reliable TIM technologies, high-accuracy train localization and optimized moving block traffic management algorithms. The main objectives were to enhance and verify existing specifications for moving block signaling, while developing formal models, algorithms and proof-of-concepts to test and validate an integrated future moving block system architecture that will provide safe and effective operational performance.

X2RAIL-5 (2020–2023)[25] (Advanced Traffic Management & Control Systems): This EU H2020 Shift2Rail project had the objective to improve the standardization and integration of formal methods application in the development of Europe's rail control systems while reducing time to market and improving effectiveness in the introduction of new signaling and supervision systems. A particular project output was to propose and apply a methodology and toolchain to automate the transformation of semi-formal (specification) models into models suitable for formal verification. The objective was to create a tool that can automatically translate the semi-formal SysML models into a formal model, to obtain a more precise and rigorous representation of the system, and to apply formal verification to prove properties against the formal model. According to this, two toolchains were proposed for the automated transformation of EULYNX SysML models into formal models.

[22] https://projects.shift2rail.org/s2r_ip2_n.aspx?p=X2RAIL-2
[23] https://projects.shift2rail.org/s2r_ip2_n.aspx?p=S2R_4SECURAIL
[24] https://cordis.europa.eu/project/id/101015416
[25] https://projects.shift2rail.org/s2r_ip2_n.aspx?p=X2RAIL-5

DisCoRail (since 2019) ("Formal methods for DIStributed COmputing in future RAILway system"): This workshop series aims to provide a forum to discuss how distributed computing affects the railway signaling domain, and in particular the intertwining of formal methods and distributed computing in the design and development of innovative train control systems. Its proceedings showcase recent experiences and advances in the application of formal methods in railways [89–91].

R2DATO (2022–2026)[26] (Rail to Digital automated up to Autonomous Train Operation): This project appears to be the only one funded by Europe's rail that mentions formal methods as "enabling techniques": modeling techniques to support the development, engineering and planning, verification and validation solutions, while incorporating formal methods to address the basic needs of rail stakeholders. This indicates that formal methods have gradually evolved from being the subject of research projects to becoming established components that are now part of the toolkit available for the development and innovation of railway transport.

8 Future Work: Synergies between (Generative) AI and Formal Methods in Railways

The recent advances in Large Language Models (LLMs) and generative Artificial Intelligence (AI) make it impossible to ignore the potential role that these technologies can play also in the railway process in general, and in a formal methods-enhanced process in particular. Companies such as Prover Technology[27] are already working on the integration of AI-based solutions with formal methods-based solutions in specific railway subsystems. According to Prover Technology, "AI can enhance the efficiency of formal verification tools, such as using AI-tuned tactics to guide the model checker during proof searches" [162]. Furthermore, "LLMs can make formal methods more accessible and easier to use" and "simplify the use of complex tools and the creation of necessary artifacts like formalized requirements and models" [163]. In this sense, LLMs can be regarded as productivity enablers, rather than safety-critical components, while preserving formal verification as the primary source of assurance. Indeed, a major bottleneck in the application of formal methods in railway projects lies in the early lifecycle phases, especially requirements engineering. Signaling and interlocking systems are typically specified using large collections of natural-language requirements, interface specifications, and safety constraints distributed across standards, national rules, and project-specific documents. LLMs are promising means to assist in transforming these heterogeneous textual artifacts into structured and analyzable representations. LLMs can support the derivation of candidate formal requirements from natural-language specifications, as done, e.g., by Durante et al. [77], for instance by translating operational rules for route setting, point locking, or MA management into temporal logic properties or contract-based specifications. By accelerating the creation of such formal artifacts, LLMs reduce the manual effort required from formal methods experts and domain experts, and lower the entry barrier for applying formal verification techniques.

Beyond requirements formalization, LLMs can also be used as interactive assistants for formal verification tools. Model checking and theorem proving for railway systems often require specialized expertise to interpret counterexamples, refine models, or adjust properties. LLMs can help explain verification results in domain-specific terms, guide engineers in refining interlocking logic or signaling models, and suggest corrective actions when safety properties are violated. Importantly, these interactions do not alter the underlying semantics of the formal tools; instead, they enhance usability and productivity, which is essential for industrial-scale railway projects. LLM explanations of formal artifacts can also improve communication between railway practitioners and formal methods experts in a process according to which the former provide the requirements and the latter produce formal artifacts, possibly with the support of LLMs. Better communication between these roles reduces the risk of misinterpretation of the railway requirements, ensuring not only that the system is built right, by means of formal verification, but also that the right system is built. To achieve this goal, however, it is necessary to equip LLMs with railway domain-specific knowledge. Recent work [92] has shown that LLMs struggle when generating models from specialized domains. The use of retrieval-augmented generation architectures, in which domain and project documents can be used to support LLM reasoning, can potentially overcome these hurdles [11].

LLMs may also provide heuristic guidance during verification activities by suggesting proof strategies, invariants, or abstractions. For example, when verifying an interlocking model, an LLM might propose candidate invariants related to mutual exclusion of conflicting

[26] https://rail-research.europa.eu/rail-projects/fp2-r2dato/
[27] https://www.prover.com/

routes or safe point positions. Any such suggestions remain subject to formal validation, ensuring that the trusted computing base remains unchanged. To achieve this goal, however, LLMs need to be equipped with formal logic knowledge and strong logical reasoning capabilities, so that hallucinations are prevented. There is work addressing this problem (cf., e.g., Cheng et al. [57] for a survey), and other contributions are also exploring the use of LLMs directly as theorem provers [187], thus these goals are realistically achievable.

While LLMs can significantly improve efficiency and accessibility, their use in safety-critical domains raises legitimate concerns regarding correctness, trustworthiness, and explainability. First and foremost, hallucinations cannot be fully avoided, also due to the way LLMs are trained, which prevents them from saying "I don't know" and leads them to produce output also in presence of uncertainty [131, 186]. This means that incorrect formal artifacts could be generated as well as misleading artifact explanations. The problem is further exacerbated by the surface-level quality and potentially deceiving nature of the typical output produced, which can look convincing and well-formed, despite being incorrect. Prior work shows that users may over-trust LLM output and adopt the recommendations even when these contain inconsistencies and errors, indicating a risk of distorted human decision making through persuasive but wrong output [170]. This implies that process-level guardrails need to be put into place to avoid over-reliance, and formal methods experts as well as railway experts and developers still need to play a primary role. They remain essential both to independently validate the generated artifacts and to ensure that any AI-assisted step is embedded within a rigorous assurance workflow, including traceability, review, verification, and accountability. Indeed, including LLMs in the formal development cycle necessarily requires humans-in-the-loop and on-the-loop. Novel workflows and processes are needed, in railways and in other domains alike, to guarantee that human-AI collaboration results in better products and decisions, rather than faster, yet insufficiently scrutinized, development outcomes. LLM-based assistance should be integrated only within workflows that preserve independent judgment and clearly allocate responsibility for the resulting artifacts and decisions.

Besides humans and workflows, formal methods can also play a role in increasing LLM trustworthiness. Ferrari and Spoletini provide roadmaps to, on the one hand, increase the usability of formal methods with the support of LLMs, along the lines of the ideas suggested above, and, on the other hand, use formal methods to make LLMs more reliable [98]. According to the authors, formal methods should be applied directly to the artifacts generated by LLMs, rather than relying on the AI's internal reasoning. In a railway context, this means that LLM-generated requirements, models, or properties must be formally checked for consistency, completeness, and correctness. For instance, an LLM-generated formalization of an ETCS braking requirement can be model checked against a reference train control model, or an interlocking rule proposed by an LLM can be verified to preserve collision freedom and derailment prevention. In such workflows, the LLM acts as a specification accelerator, while formal methods serve as a gatekeeper that filters out incorrect or unsafe artifacts.

Formal methods can also provide explainable and certifiable evidence for LLM-assisted development. Formal verification results—such as proofs, counterexamples, or verification certificates—offer objective and auditable evidence that can be integrated into railway safety cases, while LLM explanations can guide interpretation. This is particularly relevant for compliance with CENELEC standards, which require traceability, justification, and independent evidence of correctness. By grounding assurance in formal artifacts rather than AI behavior, the opacity of LLMs does not undermine certification.

Another important aspect is to constrain LLM outputs using formal specifications and domain-specific rules. In railway signaling and interlocking, formally defined grammars, specification patterns, or safety invariants can restrict the space of admissible LLM-generated artifacts. This reduces the risk of hallucinations and ensures that generated models or requirements remain aligned with established signaling principles and ERTMS/ETCS rules.

A two-way integration of formal methods and railways aligns well with existing railway engineering and certification practices. LLMs can be positioned as supportive engineering tools that improve productivity and reduce human error, while formal methods remain the authoritative mechanism for demonstrating safety and correctness. This separation *allows LLMs to remain outside the safety-critical scope*, avoiding the need to certify the AI itself. In signaling, interlocking, and ERTMS/ETCS projects, this approach enables a pragmatic adoption of AI technologies: LLMs assist engineers in handling complexity and scale, while formal verification provides the rigorous guarantees required for safety assurance. While LLMs can be used to support and enhance a formal methods-based development process, it should be remarked that: (1) LLMs are not expected to be embedded in the railway safety-critical components themselves, given their limited predictability; and (2) the overall process still includes all the standard and CENELEC-recommended

development practices, including static analysis, all the different levels of testing and associated coverage metrics, safety analyses (e.g., hazard/risk analysis, failure modes, and effect analysis), and external auditing. Overall, the ultimate product responsibility falls upon the railway system supplier who signs the safety case, and all measures need to be taken to ensure safety—including the use of formal methods—while leveraging LLMs to support process efficiency.

According to Seisenberger et al. [169], there is a skills gap concerning knowledge of AI principles, techniques, and practices among railway practitioners as well as formal methods researchers. At the same time, based on four recent white papers [22, 47, 56, 76]—all together authored by more than 50 computer scientists and practitioners worldwide—, ter Beek et al. [21] argue that Formal Methods need to be introduced as a core knowledge area in the ACM/IEEE/AAAI CS curricula guidelines [133]. Currently, Formal Methods are covered only as elective topics in distinct knowledge units of two of the 17 core knowledge areas deemed essential for Computer Science, namely Foundations of Programming Languages and Software Engineering. The evidence put forward suggests that the lack of indicating Formal Methods as a core knowledge area in the CS2023 guidelines is unjustified, arguably similar to the fact that just 10 years ago AI was not included in the CS2013 guidelines. Computer Science graduates educated in AI and formal methods will be able to reduce this skills gap and drive the integration of AI-based solutions with formal methods-based solutions in the railway sector.

9 Conclusion

Railway signaling has evolved from manual systems, which were prone to human error and limited in capacity, to automated solutions that ensure higher efficiency and safety in modern railway control. However, the transition to automatic railway control systems has posed significant challenges, particularly in terms of safety. A small failure in signaling can lead to catastrophic events such as train collisions. The shift towards automation has also entailed substantial financial investment, especially when considering the high safety standards required. The CENELEC EN 50128 European standard outlines the development requirements for safety-critical software in railway systems, recommending formal methods for the design and verification of systems requiring high safety integrity levels (SIL 3 or SIL 4). These levels correspond to maximum acceptable probabilities of dangerous failure ranging between 10^{-7} and 10^{-9} per hour. Despite the growing recognition of formal methods' advantages, particularly in terms of safety, there is still a lack of comprehensive, quantitative cost-benefit analyses. While some qualitative assessments have been conducted, the field remains under-explored in terms of the economic impact of the application of formal methods in railway control systems.

We are aware of only one cost-benefit analysis, reported in [29]. In the context of EU project 4SECURail (cf. Section 7), the authors evaluated the use of formal methods for developing the communication layer supporting the execution of the RBC/RBC handover protocol. UML/SysML models were translated into UMC [24], from which ProB and CADP/LNT [104] models were generated automatically. Based on data collected within the 4SECURail demonstrator test case, the economic analysis demonstrated a net convenience of the adoption of formal methods, generating benefits 5 times higher than cost borne by the infrastructure managers. Furthermore, benefits from time saved for passengers turned out to be the most relevant benefit category, justifying the adoption of formal methods and the necessary investment. In [154], the authors evaluated (without any monetary measurements) the effect of applying a formal method to an industrial project, and compared the positive results concerning code quality (good) and productivity (high) with those of 13 similar projects that used other formal methods such as B. The authors observed a strong reduction of the number of defects and an increase in productivity. The 2020 survey among 130 formal methods experts [105], which covered the history, current state and future outlook of formal methods in academia, industry and education, also contained a question that asked the experts to make an informal cost-benefit analysis of the return on investment over time. A small majority of 58.5% of the respondents answered that the application of formal methods is *profitable in medium and long terms*; 15% answered that they are *immediately profitable* and 12.3% that they are *profitable in the long term only*, while 2.3% answered that there is *no return on investment* and 11.5% had *no opinion*. We agree with the authors of [29], who call for "greater attention of the formal methods community to the quantification of costs and benefits parameters [...] since the evidence of the beneficial effects of formal methods is mostly given instead in the literature in a qualitative way."

Formal methods offer distinct advantages over traditional testing and simulation. Unlike probabilistic or partial approaches, formal methods provide exhaustive analysis of system behavior, ensuring that safety properties are upheld and errors are detected before deployment. Two primary techniques for formal verification are theorem proving and model checking. Theorem provers, such as

the Rocq Prover and Isabelle, allow for high-level reasoning over abstract or infinite-state systems, while model checkers like SPIN and UPPAAL automate the process of verifying finite-state models against formal specifications. These methods are invaluable for ensuring correctness in complex, safety-critical systems, where even the smallest failure can have catastrophic consequences. In addition, probabilistic model checking and statistical model checking (SMC) have enhanced the capability of formal methods to handle systems with inherent uncertainty or stochastic behavior. These methods provide quantitative analysis of system properties, such as the likelihood of a system satisfying certain temporal properties. While SMC allows for scalable verification through simulation-based methods, it still faces challenges related to the detection of rare events.

The numerous success stories across the railway industry mentioned in this paper highlight the effective use of formal methods in developing and verifying railway systems. These include the verification of the ATP system on Paris' RER Line A, the Subway Speed Control System (SSCS) in Calcutta, and Line 14 of the Paris metro [75], and derivatives thereof, like line 1 or the New York Canarsie line [79], and the driverless Paris–Roissy Airport shuttle [25], all developed with the B method. The B method has played a significant role in these successful applications, providing a formal framework for specifying and verifying railway control systems. Tools like Simulink and Stateflow, which integrate model-based design and formal verification, have also been employed in the development of systems such as Alstom's U400 system [22]. Another success story concerns the metro control system of Rio de Janeiro, developed with the support of Simulink/Stateflow [94]. These applications underline the relevance and growing impact of formal methods in the railway sector. Further success stories concern the verification of the ERTMS/ETCS European standard for railway control and management with NuSMV [58] and that of ERTMS/ETCS HL3 with a variety of formal methods and tools [19, 51, 114].

It would be interesting to understand the historical perspective of how formal methods became so well used in the railway sector, also because this might—in principle—suggest how to successfully integrate formal methods into other sectors as well. It is our firm belief that the adoption of formal methods in the railway sector was driven less by theoretical appeal than by concrete evidence that rigorous verification could catch what conventional testing could not.

The story begins in the late 1980s, when RATP and the consortium developing SACEM, the automatic speed protection system deployed on Paris RER Line A (cf. Section 4.1), faced the challenge of validating the first safety-critical control/command software ever operated in French railways. They turned to Jean-Raymond Abrial, who proposed formalization and verification of the software specification using what would become a sketch of the B method (cf. Section 4); critically, this process uncovered major safety issues that had gone undetected by other means. SACEM entered into operation in 1988 [52], and the experience had direct institutional consequences: RATP subsequently requested the use of the B method for the driverless Paris Métro Line 14 [52]. SACEM's deployment also eliminated the need for an additional railway line, a saving estimated at hundreds of millions of dollars [106], providing a compelling economic argument alongside the safety case.

A more recent episode illustrates how incidents continue to reinforce this dynamic. In December 2019, a TGV from Paris to Rennes crossed a diversion switch at La Milesse at 165 km/h, well above the 100 km/h speed limit imposed by the switch geometry, because the ETCS L2 system was displaying 170 km/h as the maximum authorized speed. The French Land Transport Accident Investigation Bureau (BEA-TT) traced the cause to an error in ETCS data parameterization, undetected during verification, and the evaluator's response to the resulting recommendations explicitly called for the use of formal proof tools and compliance with applicable standards [50].

We identify a recurring pattern: operational failures expose verification gaps, and formal methods emerge as the preferred solution, given that they constitute the most rigorous techniques available, recognized as such by standards like EN 50128 [80] (cf. Section 3). The industrial uptake of this approach has been documented across multiple operators and suppliers well beyond the French context [18]. The normative evolution has continued with EN 50716 [83], which replaces and unifies EN 50128 and EN 50657 [81] into a single standard (cf. Section 5.2.3). Significantly, EN 50716 extends its strong recommendation of formal proof verification to all safety integrity levels from SIL 1 to SIL 4 [54], whereas previously formal methods were strongly recommended primarily at SIL 3 and SIL 4. This marks a decisive shift: formal verification is now a baseline expectation across the entire railway software domain.

The integration of formal methods into the development lifecycle of railway signaling systems holds the promise of not only improving safety but also optimizing costs in the long run. However, the adoption of formal methods in this context faces barriers, including the complexity of tools, the need for specialized expertise, and computational limitations. The clear recommendation of

formal methods in the railway industry standards has caused the railway sector to invest in the adoption of formal methods and tools. Unfortunately, this does not provide any concrete suggestions on how to improve the integration of formal methods into other sectors.

We are not aware of any literature reporting evidence of projects that substantially failed to employ formal methods in the railway sector. Failure received little publicity both in industry and in the academic community. But the challenges often referred to in documents and the not fully pervasive industrial adoption do suggest that failures have been experienced. Indeed, the discontinuing of pilot projects, with the apparently missed adoption of their results in industry could be taken as an indication of cases of low satisfaction of the application of formal methods, but it is difficult to understand from the outside whether a company has completely ignored the results of a pilot project, and to trace the reasons for doing so, or whether the company has rather integrated (some of) the project results in their development process without disclosing this.

The success stories on the application of formal methods in the railway industry have also been affected by the recent trend of railway systems manufacturing companies to merge and incorporate each other, to form a few large global actors, in contrast with the many smaller national manufacturers of the past. This process has often pushed uniformity in software development processes, dismissing previous experiences that may have included formal methods adoption. Indeed, all companies were successfully building safe products, each with their own method, and the merits and advantages of formal methods may have gone unrecognized, especially in the case of a large company incorporating a small one. However, this phenomenon can only be referred to as personal experience of the authors in their collaborations with industry, since it is not documented in the literature.

Despite the challenges, the ability of formal methods to provide exhaustive, mathematically guaranteed correctness makes them an indispensable part of the modern railway system engineering process. An important goal for the future is to integrate the use of AI-based solutions in the current formal methods-based solutions in railways.

Pointers to Public Resources for Tools

AMC	integrated in UMC	**Prover PSL**	not publicly available
Atelier B	https://www.atelierb.eu/	**RDV**	not publicly available
BART	integrated in Atelier B	**Rocq Prover**	https://rocq-prover.org/
CADP	https://cadp.inria.fr/	**Rodin**	https://www.event-b.org/
Caval	not publicly available	**Rubin**	not publicly available
Dave	not publicly available	**SafeProver**	not publicly available
DTVT	not publicly available	**SCADE**	not publicly available
FDR4	https://cocotec.io/fdr/	**Simulink**	not publicly available
Isabelle	https://isabelle.in.tum.de/	**SMV**	http://www-2.cs.cmu.edu/~modelcheck/smv.html
mCRL2	https://mcrl2.org/	**SPIN**	https://spinroot.com/
NuSMV	https://nusmv.fbk.eu/	**Stateflow**	not publicly available
nuXmv	https://nuxmv.fbk.eu/	**S3**	not publicly available
OVADO	https://www.ovado.net/	**TLA+**	https://foundation.tlapl.us/
ProB	https://prob.hhu.de/	**UMC**	https://fmt.isti.cnr.it/umc/
Prover Certifier	not publicly available	**UPPAAL (SMC)**	https://uppaal.org/
Prover iLock	not publicly available	**Z3**	https://github.com/Z3Prover/

Acknowledgments

The authors would like to thank Arne Borälv, Nicolas Breton, and Véronique Delebarre for information on model-checking tools at Prover Technology, Systerel, and SafeRiver, respectively. The authors would also like to thank the four anonymous reviewers for their useful comments and suggestions. Part of this work was carried out within the MUR PRIN 2022 PNRR P2022A492B project ADVENTURE (ADVancEd iNtegraTed evalUation of Railway systEms) and the MOST – Sustainable Mobility National Research Center and received funding from the European Union NextGenerationEU (PIANO NAZIONALE DI RIPRESA E RESILIENZA (PNRR) – MISSIONE 4, COMPONENTE 2, INVESTIMENTO 1.4 – D.D. 1033 17/06/2022, CN00000023. This manuscript reflects only the authors' views and opinions, neither the European Union nor the European Commission can be considered responsible for them.

References

[1] Jean-Raymond Abrial. 1996. *The B-Book: Assigning Programs to Meanings.* Cambridge University Press, UK. doi:10.1017/CBO9780511624162

[2] Jean-Raymond Abrial. 2006. Formal methods in industry: achievements, problems, future. In *Proceedings of the 28th International Conference on Software Engineering (ICSE'06).* ACM, USA, 761–768. doi:10.1145/1134285.1134406

[3] Jean-Raymond Abrial. 2007. Formal Methods: Theory Becoming Practice. *J. Univers. Comput. Sci.* 13, 5 (2007), 619–628. doi:10.3217/jucs-013-05-0619

[4] Jean-Raymond Abrial. 2010. *Modeling in Event-B: System and Software Engineering.* Cambridge University Press, UK. doi:10.1017/CBO9781139195881

[5] Jean-Raymond Abrial. 2020. The ABZ-2018 case study with Event-B. *Int. J. Softw. Tools Technol. Transf.* 22, 3 (2020), 257–264. doi:10.1007/s10009-019-00525-3

[6] Gul Agha and Karl Palmskog. 2018. A Survey of Statistical Model Checking. *ACM Trans. Model. Comput. Simul.* 28, 1 (2018), 6:1–6:39. doi:10.1145/3158668

[7] Ehsan Ahmad, Yunwei Dong, Brian R. Larson, Jidong Lü, Tao Tang, and Naijun Zhan. 2015. Behavior modeling and verification of movement authority scenario of Chinese Train Control System using AADL. *Sci. China Inf. Sci.* 58, 11 (2015), 1–20. doi:10.1007/s11432-015-5346-2

[8] Arturo Amendola, Anna Becchi, Roberto Cavada, Alessandro Cimatti, Andrea Ferrando, Lorenzo Pilati, Giuseppe Scaglione, Alberto Tacchella, and Marco Zamboni. 2022. NORMA: a tool for the analysis of Relay-based Railway Interlocking Systems. In *Proceedings of the 28th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'22) (LNCS, Vol. 13243)*, Dana Fisman and Grigore Rosu (Eds.). Springer, Germany, 125–142. doi:10.1007/978-3-030-99524-9_7

[9] A. Anselmi, Cinzia Bernardeschi, Alessandro Fantechi, Stefania Gnesi, Salvatore Larosa, Giorgio Mongardi, and Fernando Torielli. 1995. An Experience in Formal Verification of Safety Properties of a Railway Signalling Control System. In *Proceedings of the 14th International Conference on Computer Safety, Reliability and Security (SAFECOMP'95)*, Gerhard Rabe (Ed.). Springer, Germany, 474–488. doi:10.1007/978-1-4471-3054-3_33

[10] Paolo Arcaini, Jan Kofroň, and Pavel Ježek. 2020. Validation of the Hybrid ERTMS/ETCS Level 3 using SPIN. *Int. J. Softw. Tools Technol. Transf.* 22, 3 (2020), 265–279. doi:10.1007/s10009-019-00539-x

[11] Chetan Arora, Tomas Herda, and Verena Homm. 2024. Generating Test Scenarios from NL Requirements Using Retrieval-Augmented LLMs: An Industrial Study. In *Proceedings of the 32nd International Requirements Engineering Conference (RE'24).* IEEE, USA, 240–251. doi:10.1109/RE59067.2024.00031

[12] Muhammad Atif and Jan Friso Groote. 2023. *Understanding Behaviour of Distributed Systems Using mCRL2.* Studies in Systems, Decision and Control, Vol. 458. Springer, Germany. doi:10.1007/978-3-031-23008-0

[13] 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, Vol. 3455)*, Helen Treharne, Steve King, Martin C. Henson, and Steve A. Schneider (Eds.). Springer, Germany, 334–354. doi:10.1007/11415787_20

[14] Christel Baier and Joost-Pieter Katoen. 2008. *Principles of Model Checking.* MIT Press, USA. https://mitpress.mit.edu/9780262026499/principles-of-model-checking/

[15] Yongxiang Bao, Mingsong Chen, Qi Zhu, Tongquan Wei, Frédéric Mallet, and Tingliang Zhou. 2017. Quantitative Performance Evaluation of Uncertainty-Aware Hybrid AADL Designs Using Statistical Model Checking. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* 36, 12 (2017), 1989–2002. doi:10.1109/TCAD.2017.2681076

[16] Maarten Bartholomeus, Bas Luttik, and Tim Willemse. 2018. Modeling and Analysing ERTMS Hybrid Level 3 with the mCRL2 Toolset. In *FMICS 2018 (LNCS, Vol. 11119)*, Falk Howar and Jiří Barnat (Eds.). Springer, Germany, 98–114. doi:10.1007/978-3-030-00244-2_7

[17] Davide Basile, Alessandro Fantechi, Luigi Rucher, and Gianluca Mandò. 2021. Analysing an autonomous tramway positioning system with the Uppaal Statistical Model Checker. *Form. Asp. Comput.* 33, 6 (2021), 957–987. doi:10.1007/s00165-021-00556-1

[18] 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. In *Proceedings of the 14th International Conference on Integrated Formal Methods (iFM'18) (LNCS, Vol. 11023)*, Carlo A. Furia and Kirsten Winter (Eds.). Springer, Germany, 20–29. doi:10.1007/978-3-319-98938-9_2

[19] Davide Basile, Maurice H. ter Beek, Alessio Ferrari, and Axel Legay. 2022. Exploring the ERTMS/ETCS full moving block specification: An experience with formal methods. *Int. J. Softw. Tools Technol. Transf.* 24, 3 (2022), 351–370. doi:10.1007/s10009-022-00653-3

[20] ter Beek, Maurice H., 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, Vol. 11800)*, Maurice H. ter Beek, Annabelle McIver, and José N. Oliveira (Eds.). Springer, Germany, 762–772. doi:10.1007/978-3-030-30942-8_46

[21] ter Beek, Maurice H., Manfred Broy, and Brijesh Dongol. 2024. The Role of Formal Methods in Computer Science Education. *ACM Inroads* 15, 4 (2024), 58–66. doi:10.1145/3702231

[22] ter Beek, Maurice H., Rod Chapman, Rance Cleaveland, Hubert Garavel, Rong Gu, Ivo ter Horst, Jeroen J. A. Keiren, Thierry Lecomte, Michael Leuschel, Kristin Yvonne Rozier, Augusto Sampaio, Cristina Seceleanu, Martyn Thomas, Tim A. C. Willemse, and Lijun Zhang. 2025. Formal Methods in Industry. *Form. Asp. Comput.* 37, 1 (2025), 7:1–7:38. doi:10.1145/3689374

[23] ter Beek, Maurice H., Alessandro Fantechi, and Stefania Gnesi. 2025. Formal Methods for Industrial Critical Systems: 30 Years of Railway Applications. In *The Combined Power of Research, Education, and Dissemination (LNCS, Vol. 15240)*, Mike Hinchey and Bernhard Steffen (Eds.). Springer, Germany, 327–344. doi:10.1007/978-3-031-73887-6_21

[24] ter Beek, Maurice H., Alessandro Fantechi, Stefania Gnesi, and Franco Mazzanti. 2011. A state/event-based model-checking approach for the analysis of abstract system properties. *Sci. Comput. Program.* 76, 2 (2011), 119–135. doi:10.1016/j.scico.2010.07.002

[25] 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, Vol. 1708)*, Jeannette M. Wing, Jim Woodcock, and Jim Davies (Eds.). Springer, Germany, 369–387. doi:10.1007/3-540-48119-2_22

[26] Gerd Behrmann, Agnès Cougnard, Alexandre David, Emmanuel Fleury, Kim G. Larsen, and Didier Lime. 2007. UPPAAL-Tiga: Time for Playing Games!. In *Proceedings of the 19th International Conference on Computer Aided Verification (CAV'07) (LNCS, Vol. 4590)*, Werner Damm and Holger Hermanns (Eds.). Springer, Germany, 121–125. doi:10.1007/978-3-540-73368-3_14

[27] Gerd Behrmann, Alexandre David, and Kim Guldstrand Larsen. 2004. A Tutorial on Uppaal. In *Revised Lectures of the 4th International School on Formal Methods for the Design of Computer, Communication and Software Systems: Formal Methods for the Design of Real-Time Systems (SFM-RT'04) (LNCS, Vol. 3185)*, Marco Bernardo and Flavio Corradini (Eds.). Springer, Germany, 200–236. doi:10.1007/978-3-540-30080-9_7

[28] Gerd Behrmann, Alexandre David, Kim G. Larsen, John Håkansson, Paul Pettersson, Wang Yi, and Martijn Hendriks. 2006. UPPAAL 4.0. In *Proceedings of the 3rd International Conference on the Quantitative Evaluation of Systems (QEST'06)*. IEEE, USA, 125–126. doi:10.1109/QEST.2006.59

[29] Dimitri Belli, Alessandro Fantechi, Stefania Gnesi, Laura Masullo, Frando Mazzanti, Lisa Quadrini, Daniele Trentini, and Carlo Vaghi. 2023. The 4SECURail Case Study on Rigorous Standard Interface Specifications. In *Proceedings of the 28th International Conference on Formal Methods for Industrial Critical Systems (FMICS'23) (LNCS, Vol. 14290)*, Alessandro Cimatti and Laura Titolo (Eds.). Springer, Germany, 22–39. doi:10.1007/978-3-031-43681-9_2

[30] Yves Bertot and Pierre Castéran. 2004. *Interactive Theorem Proving and Program Development — Coq'Art: The Calculus of Inductive Constructions*. Springer, Germany. doi:10.1007/978-3-662-07964-5

[31] 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, Hungary, 20 pages. http://www.imm.dtu.dk/~dibj/dines-amore.pdf

[32] 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, Vol. 8442)*, Cliff Jones, Pekka Pihlajasaari, and Jun Sun (Eds.). Springer, Germany, 42–61. doi:10.1007/978-3-319-06410-9_4

[33] Andrea Bonacchi, Alessandro Fantechi, Stefano Bacherini, and Matteo Tempestini. 2016. Validation process for railway interlocking systems. *Sci. Comput. Program.* 128 (2016), 2–21. doi:10.1016/j.scico.2016.04.004

[34] David Bonvoisin. 2016. 25 Years of Formal Methods at RATP: From Manual Approach for Proof of Programs to Instrumented Demonstration of Railway Systems Safety. International Railway Safety Council Meeting (IRSC'16), 8 pages. https://international-railway-safety-council.com/wp-content/uploads/2017/09/bonvoisin-25-years-of-formal-methods-at-ratp.pdf

[35] Arne Borälv. 1997. The Industrial Success of Verification Tools Based on Stålmarck's Method. In *Proceedings of the 9th International Conference on Computer Aided Verification (CAV'97) (LNCS, Vol. 1254)*, Orna Grumberg (Ed.). Springer, Germany, 7–10. doi:10.1007/3-540-63166-6_3

[36] Arne Borälv. 1998. Case Study: Formal Verification of a Computerized Railway Interlocking. *Form. Asp. Comput.* 10, 4 (1998), 338–360. doi:10.1007/S001650050021

[37] Arne Borälv. 2018. Interlocking Design Automation Using Prover Trident. In *Proceedings of the 22nd International Symposium on Formal Methods (FM'18) (LNCS, Vol. 10951)*, Dieter Hutter, Werner Stephan, Paolo Traverso, and Markus Ullmann (Eds.). Springer, Germany, 653–656. doi:10.1007/978-3-319-95582-7_39

[38] Arne Borälv, Randolf Berglehner, Ibtihel Cherif, Daniel Fredholm, Dominik Hansen, Javier Magro, Luis-Fernando Mejia, David Mentré, Abdul Rasheeq, Daniel Schwencke, and Tom Werner. 2022. Holistic Study of Formal Methods and Standardization in Specification, Development, Verification and Validation of Railway Signalling System Software. 13th World Congress on Railway Research (WCRR'22), 6 pages. https://www.researchgate.net/profile/Arne-Boralv/publication/363535915_Holistic_study_of_Formal_Methods_and_Standardization_in_development_verification_and_validation_of_railway_signalling_system_software/links/63217f59873eca0c00872c32/Holistic-study-of-Formal-Methods-and-Standardization-in-development-verification-and-validation-of-railway-signalling-system-software.pdf

[39] Mark Bosschaart, Egidio Quaglietta, Bob Janssen, and Rob M. P. Goverde. 2015. Efficient formalization of railway interlocking data in RailML. *Inf. Syst.* 49 (2015), 126–141. doi:10.1016/j.is.2014.11.007

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

[41] Jean-Louis Boulanger (Ed.). 2012. *Industrial Use of Formal Methods: Formal Verification*. Wiley, UK. doi:10.1002/9781118561829

[42] Jean-Louis Boulanger (Ed.). 2014. *Formal Methods Applied to Industrial Complex Systems: Implementation of the B Method*. Wiley, UK. doi:10.1002/9781119002727

[43] Mark Bouwman, Djurre van der Wal, Bas Luttik, Mariëlle Stoelinga, and Arend Rensink. 2023. A Case in Point: Verification and Testing of a EULYNX Interface. *Form. Asp. Comput.* 35, 1 (2023), 2:1–2:38. doi:10.1145/3528207

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

[45] 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, Vol. 670)*, Jim Woodcock and Peter Gorm Larsen (Eds.). Springer, Germany, 183–195. doi:10.1007/BFb0024646

[46] Nicolas Breton and Yoann Fonteneau. 2016. S3: Proving the Safety of Critical Systems. In *Proceedings of the 1st International Conference on Reliability, Safety, and Security of Railway Systems: Modelling, Analysis, Verification, and Certification (RSSRail'16) (LNCS, Vol. 9707)*. Springer, Germany, 231–242. doi:10.1007/978-3-319-33951-1_17

[47] Manfred Broy, Achim D. Brucker, Alessandro Fantechi, Mario Gleirscher, Klaus Havelund, Markus Alexander Kuppe, Alexandra Mendes, André Platzer, Jan Oliver Ringert, and Allison Sullivan. 2025. Does Every Computer Scientist Need to Know Formal Methods? *Form. Asp. Comput.* 37, 1 (2025), 6:1–6:17. doi:10.1145/3670795

[48] Lilian Burdy and Jean-Marc Meynadier. 1999. Automatic Refinement. In *Proceedings of the FM'99 B Users Group Meeting on Applying B in an industrial context: Tools, Lessons and Techniques (BUGM'99)*. 3–15. https://www-sop.inria.fr/everest/personnel/Lilian.Burdy/ug020003.pdf

[49] Lilian Burdy and Jean-Marc Meynadier. 2000. Experience on the Use of a Formal Method in a Railway Company. *IFAC Proc. Vol.* 33 (2000), 193–197. doi:10.1016/S1474-6670(17)38145-4 Proceedings of the 9th IFAC Symposium on Control in Transportation Systems (CTS'00).

[50] Bureau d'Enquêtes sur les Accidents de Transport Terrestre (BEA-TT). 2021. *Rapport d'enquête technique sur la survitesse d'un TGV à La Milesse (72) le 22 décembre 2019.* Technical Report EQ-BEAT–21-6–FR. BEA-TT. https://www.bea-tt.developpement-durable.gouv.fr/IMG/pdf/rapport_beatt_2020_01.pdf In French.

[51] Michael Butler, Thai Son Hoang, Alexander Raschke, and Klaus Reichl. 2020. Introduction to special section on the ABZ 2018 case study: Hybrid ERTMS/ETCS Level 3. *Int. J. Softw. Tools Technol. Transf.* 22, 3 (2020), 249–255. doi:10.1007/s10009-020-00562-3

[52] 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, Vol. 12327)*, Maurice H. ter Beek and Dejan Ničković (Eds.). Springer, Germany, 189–209. doi:10.1007/978-3-030-58298-2_8

[53] Michael J. Butler, Dana Dghaym, Tomas Fischer, Thai Son Hoang, Klaus Reichl, Colin F. Snook, and Peter Tummeltshammer. 2017. Formal Modelling Techniques for Efficient Development of Railway Control Products. In *Proceedings of the 2nd International Conference on Reliability, Safety, and Security of Railway Systems: Modelling, Analysis, Verification, and Certification (RSSRail'17) (LNCS, Vol. 10598)*, Alessandro Fantechi, Thierry Lecomte, and Alexander B. Romanovsky (Eds.). Springer, Germany, 71–86. doi:10.1007/978-3-319-68499-4_5

[54] Ricardo Camacho. 2025. The Impact of EN 50716 on Rail Digitalization and Advanced Technologies. Parasoft blog. https://www.parasoft.com/blog/en-50716-impact-on-rail-digitalization/

[55] Roberto Cavada, Alessandro Cimatti, Michele Dorigatti, Alberto Griggio, Alessandro Mariotti, Andrea Micheli, Sergio Mover, Marco Roveri, and Stefano Tonetta. 2014. The nuXmv Symbolic Model Checker. In *Proceedings of the 26th International Conference on Computer Aided Verification (CAV'14) (LNCS, Vol. 8559)*, Armin Biere and Roderick Bloem (Eds.). Springer, Germany, 334–342. doi:10.1007/978-3-319-08867-9_22

[56] Antonio Cerone, Markus Roggenbach, James Davenport, Casey Denner, Marie Farrell, Magne Haveraaen, Faron Moller, Philipp Körner, Sebastian Krings, Peter Csaba Ölveczky, Bernd-Holger Schlingloff, Nikolay Shilov, and Rustam Zhumagambetov. 2021. Rooting Formal Methods Within Higher Education Curricula for Computer Science and Software Engineering – A White Paper. In *Revised Selected Papers of the 1st International Workshop on Formal Methods – Fun for Everybody (FMFun'19) (CCIS, Vol. 1301)*, Antonio Cerone and Markus Roggenbach (Eds.). Springer, Germany, 1–26. doi:10.1007/978-3-030-71374-4_1

[57] Fengxiang Cheng, Haoxuan Li, Fenrong Liu, Robert van Rooij, Kun Zhang, and Zhouchen Lin. 2025. Empowering LLMs with Logical Reasoning: A Comprehensive Survey. In *Proceedings of the 34th International Joint Conference on Artificial Intelligence (IJCAI'25)*. IJCAI, USA, 10400–10408. doi:10.24963/ijcai.2025/1155

[58] 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, USA, 109–118. doi:10.1145/1810295.1810312

[59] Alessandro Cimatti, Edmund M. Clarke, Enrico Giunchiglia, Fausto Giunchiglia, Marco Pistore, Marco Roveri, Roberto Sebastiani, and Armando Tacchella. 2002. NuSMV 2: An OpenSource Tool for Symbolic Model Checking. In *Proceedings of the 14th International Conference on Computer Aided Verification (CAV'02) (LNCS, Vol. 2404)*, Ed Brinksma and Kim G. Larsen (Eds.). Springer, Germany, 359–364. doi:10.1007/3-540-45657-0_29

[60] Alessandro Cimatti, Fausto Giunchiglia, Giorgio Mongardi, Dario Romano, Fernando Torielli, and Paolo Traverso. 1998. Formal Verification of a Railway Interlocking System using Model Checking. *Form. Asp. Comput.* 10, 4 (1998), 361–380. doi:10.1007/S001650050022

[61] Alessandro Cimatti, Marco Roveri, Angelo Susi, and Stefano Tonetta. 2011. Formalizing requirements with object models and temporal constraints. *Softw. Syst. Model.* 10, 2 (2011), 147–160. doi:10.1007/s10270-009-0130-7

[62] Alessandro Cimatti, Marco Roveri, Angelo Susi, and Stefano Tonetta. 2012. Validation of requirements for hybrid systems: A formal approach. *ACM Trans. Softw. Eng. Methodol.* 21, 4 (2012), 22:1–22:34. doi:10.1145/2377656.2377659

[63] Edmund M. Clarke, Thomas A. Henzinger, Helmut Veith, and Roderick Bloem (Eds.). 2018. *Handbook of Model Checking*. Springer, Germany. doi:10.1007/978-3-319-10575-8

[64] 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. doi:10.1145/242223.242257

[65] Mathieu Comptier, David Déharbe, Julien Molinero Perez, Louis Mussat, Pierre Thibaut, and Denis Sabatier. 2017. Safety Analysis of a CBTC System: A Rigorous Approach with Event-B. In *Proceedings of the 2nd International Conference on Reliability, Safety, and Security of Railway Systems: Modelling, Analysis, Verification, and Certification (RSSRail'17) (LNCS, Vol. 10598)*, Alessandro Fantechi, Thierry Lecomte, and Alexander B. Romanovsky (Eds.). Springer, Germany, 148–159. doi:10.1007/978-3-319-68499-4_10

[66] Mathieu Comptier, Michael Leuschel, Luis-Fernando Mejia, Julien Perez, and Mareike Mutz. 2019. Property-Based Modelling and Validation of a CBTC Zone Controller in Event-B. 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)*, Simon Collart Dutilleul, Thierry Lecomte, and Alexander B. Romanovsky (Eds.). Springer, Germany, 202–212. doi:10.1007/978-3-030-18744-6_13

[67] Dan Craigen, Susan Gerhart, and Ted Ralston. 1995. Formal Methods Reality Check: Industrial Usage. *IEEE Trans. Softw. Eng.* 21, 2 (1995), 90–98. doi:10.1109/32.345825

[68] 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, UK. doi:10.1016/C2009-0-20452-1

[69] Alcino Cunha and Nuno Macedo. 2020. Validating the Hybrid ERTMS/ETCS Level 3 concept with Electrum. *Int. J. Softw. Tools Technol. Transf.* 22, 3 (2020), 281–296. doi:10.1007/s10009-019-00540-4

[70] 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, Vol. C-10)*, Michel Diaz and Roland Groz (Eds.). North-Holland, The Netherlands, 199–213.

[71] Alexandre David, Peter G. Jensen, Kim G. Larsen, Marius Mikucionis, and Jakob H. Taankvist. 2015. Uppaal Stratego. In *Proceedings of the 21st International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'15) (LNCS, Vol. 9035)*, Christel Baier and Cesare Tinelli (Eds.). Springer, Germany, 206–211. doi:10.1007/978-3-662-46681-0_16

[72] Alexandre David, Kim G. Larsen, Axel Legay, Marius Mikucionis, and Danny B. Poulsen. 2015. Uppaal SMC tutorial. *Int. J. Softw. Tools Technol. Transf.* 17, 4 (2015), 397–415. doi:10.1007/S10009-014-0361-Y

[73] 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, Vol. 8187)*, Charles Pecheur and Michael Dierkes (Eds.). Springer, Germany, 63–77. doi:10.1007/978-3-642-41010-9_5

[74] Dana Dghaym, Mohammadsadegh Dalvandi, Michael Poppleton, and Colin Snook. 2020. Formalising the Hybrid ERTMS Level 3 specification in iUML-B and Event-B. *Int. J. Softw. Tools Technol. Transf.* 22, 3 (2020), 297–313. doi:10.1007/s10009-019-00548-w

[75] Daniel Dollé, Didier Essamé, and Jérôme Falampin. 2003. B dans le transport ferroviaire: L'expérience de Siemens Transportation Systems. *Tech. Sci. Inform.* 22, 1 (2003), 11–32. doi:10.3166/tsi.22.11-32

[76] Brijesh Dongol, Catherine Dubois, Stefan Hallerstede, Eric Hehner, Carroll Morgan, Peter Müller, Leila Ribeiro, Alexandra Silva, Graeme Smith, and Erik de Vink. 2025. On Formal Methods Thinking in Computer Science Education. *Form. Asp. Comput.* 37, 1 (2025), 8:1–8:23. doi:10.1145/3670419

[77] Damiano Duranti, Paolo Giorgini, Andrea Mazzullo, Marco Robol, and Marco Roveri. 2024. LLM-Driven Knowledge Extraction in Temporal and Description Logics. In *Proceedings of the 24th International Conference on Knowledge Engineering and Knowledge Management (EKAW'24) (LNCS, Vol. 15370)*, Mehwish Alam, Marco Rospocher, Marieke van Erp, Laura Hollink, and Genet Asefa Gesese (Eds.). Springer, Germany, 190–208. doi:10.1007/978-3-031-77792-9_12

[78] Cindy Eisner. 2002. Using symbolic CTL model checking to verify the railway stations of Hoorn-Kersenboogerd and Heerhugowaard. *Int. J. Softw. Tools Technol. Transf.* 4, 1 (2002), 107–124. doi:10.1007/s100090100063

[79] Didier Essamé and Daniel Dollé. 2007. B in Large Scale Projects: The Canarsie Line CBTC Experience. In *Proceedings of the 7th International Conference of B Users (B'07) (LNCS, Vol. 4355)*, Jacques Julliand and Olga Kouchnarenko (Eds.). Springer, Germany, 252–254. doi:10.1007/11955757_21

[80] 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/14317747/en-50128

[81] European Committee for Electrotechnical Standardization. 2017. CENELEC EN 50657: Railway Applications – Rolling stock applications – Software on Board Rolling Stock. https://standards.globalspec.com/std/10182637/en-50657

[82] European Committee for Electrotechnical Standardization. 2018. CENELEC EN 50129: Railway applications – Communication, signalling and processing systems – Safety related electronic systems for signalling. https://standards.globalspec.com/std/13113133/en-50129

[83] European Committee for Electrotechnical Standardization. 2023. CENELEC EN 50716: Railway Applications – Requirements for software development. https://standards.globalspec.com/std/14648351/en-50716

[84] Jérôme Falampin, Hung Le-Dang, Michael Leuschel, Mikael Mokrani, and Daniel Plagge. 2013. Improving Railway Data Validation with ProB. In *Industrial Deployment of System Engineering Methods*, Alexander B. Romanovsky and Martyn Thomas (Eds.). Springer, Germany, 27–43. doi:10.1007/978-3-642-33170-1_4

[85] 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. doi:10.1109/TITS.2015.2446985

[86] 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 Global, USA, Chapter 12, 282–297. doi:10.4018/978-1-4666-1643-1.ch012

[87] 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, Vol. 8368)*, Steve Counsell and Manuel Núñez (Eds.). Springer, Germany, 167–183. doi:10.1007/978-3-319-05032-4_13

[88] 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, UK, Chapter 4, 61–84.

doi:10.1002/9781118459898.ch4

[89] Alessandro Fantechi, Stefania Gnesi, and Anne E. Haxthausen. 2020. Formal Methods for Distributed Computing in Future Railway Systems. In *Proceedings of the 9th International Symposium on Leveraging Applications of Formal Methods, Verification and Validation: Applications (ISoLA'20) (LNCS, Vol. 12478)*, Tiziana Margaria and Bernhard Steffen (Eds.). Springer, Germany, 389–392. doi:10.1007/978-3-030-61467-6_24

[90] Alessandro Fantechi, Stefania Gnesi, and Anne E. Haxthausen. 2022. Formal Methods for Distributed Control Systems of Future Railways. In *Proceedings of the 11th International Symposium on Leveraging Applications of Formal Methods, Verification and Validation: Practice (ISoLA'22) (LNCS, Vol. 13704)*, Tiziana Margaria and Bernhard Steffen (Eds.). Springer, Germany, 243–245. doi:10.1007/978-3-031-19762-8_19

[91] Alessandro Fantechi, Stefania Gnesi, and Anne E. Haxthausen. 2024. Formal methods for Distributed Computing in future Railway systems. In *Proceedings of the 12th International Symposium on Leveraging Applications of Formal Methods, Verification and Validation: Application Areas (ISoLA'24) (LNCS, Vol. 15223)*, Tiziana Margaria and Bernhard Steffen (Eds.). Springer, Germany, 109–111. doi:10.1007/978-3-031-75390-9_7

[92] Alessio Ferrari, Sallam Abualhaija, and Chetan Arora. 2024. Model Generation with LLMs: From Requirements to UML Sequence Diagrams. In *Proceedings of the 32nd International Requirements Engineering Conference Workshops (REW'24)*. IEEE, USA, 291–300. doi:10.1109/REW61692.2024.00044

[93] 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. doi:10.1109/MS.2013.44

[94] 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. doi:10.1016/j.scico.2012.04.003

[95] Alessio Ferrari, Gianluca Magnani, Daniele Grasso, and Alessandro Fantechi. 2010. Model Checking Interlocking Control Tables. In *Proceedings of the 8th Symposium on Formal Methods for Automation and Safety in Railway and Automotive Systems (FORMS/FORMAT'10)*, Eckehard Schnieder and Géza Tarnai (Eds.). Springer, Germany, 107–115. doi:10.1007/978-3-642-14261-1_11

[96] Alessio Ferrari, Franco Mazzanti, Davide Basile, and Maurice H. ter Beek. 2022. Systematic Evaluation and Usability Analysis of Formal Methods Tools for Railway Signaling System Design. *IEEE Trans. Softw. Eng.* 48, 11 (2022), 4675–4691. doi:10.1109/TSE.2021.3124677

[97] 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, USA, 62–74. doi:10.1145/3377811.3380373

[98] Alessio Ferrari and Paola Spoletini. 2025. Formal requirements engineering and large language models: A two-way roadmap. *Inf. Softw. Technol.* 181 (2025), 107697. doi:10.1016/J.INFSOF.2025.107697

[99] Alessio Ferrari and Maurice H. ter Beek. 2022. Formal Methods in Railways: a Systematic Mapping Study. *ACM Comput. Surv.* 55, 4 (2022), 69:1–69:37. doi:10.1145/3520480

[100] 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)*, Simon Collart-Dutilleul, Thierry Lecomte, and Alexander Romanovsky (Eds.). Springer, Germany, 226–241. doi:10.1007/978-3-030-18744-6_15

[101] Wan Fokkink. 2026. A Survey on Formal Methods for Railway Interlockings: From Relays to Satellite-Based Moving Block Signaling. *IEEE Trans. Intell. Transp. Syst.* (2026). doi:10.1109/TITS.2026.3661866

[102] Wan F. Fokkink. 1996. Safety Criteria for the Vital Processor Interlocking at Hoorn-Kersenboogerd. In *Proceedings of the 5th Conference on Computers in Railways (COMPRAIL'96)*, Vol. I: Railway Systems and Management. Computational Mechanics Publications, Germany, 101–110.

[103] 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/SharedDocs/Downloads/DE/BSI/Publikationen/Studien/formal_methods_study_875/formal_methods_study_875.html

[104] Hubert Garavel, Frédéric Lang, Radu Mateescu, and Wendelin Serwe. 2013. CADP 2011: a toolbox for the construction and analysis of distributed processes. *Int. J. Softw. Tools Technol. Transf.* 15, 2 (2013), 89–107. doi:10.1007/s10009-012-0244-z

[105] 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, Vol. 12327)*, Maurice H. ter Beek and Dejan Ničković (Eds.). Springer, Germany, 3–69. doi:10.1007/978-3-030-58298-2_1

[106] Susan Gerhart, Dan Craigen, and Ted Ralston. 1994. Case study: Paris Metro Signaling System. *IEEE Softw.* 11, 1 (1994), 32–35. doi:10.1109/MS.1994.1279941

[107] Susan Gerhart, Dan Craigen, and Ted Ralston. 1994. Experience with Formal Methods in Critical Systems. *IEEE Softw.* 11, 1 (1994), 21–28. doi:10.1109/52.251198

[108] Mohamed Ghazel. 2017. A Control Scheme for Automatic Level Crossings Under the ERTMS/ETCS Level 2/3 Operation. *IEEE Trans. Intell. Transp. Syst.* 18, 10 (2017), 2667–2680. doi:10.1109/TITS.2017.2657695

[109] Thomas Gibson-Robinson, Philip J. Armstrong, Alexandre Boulgakov, and A. W. Roscoe. 2014. FDR3 – A Modern Refinement Checker for CSP. In *Proceedings of the 20th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'14) (LNCS, Vol. 8413)*, Erika Ábrahám and Klaus Havelund (Eds.). Springer, Germany, 187–201. doi:10.1007/978-3-642-54862-8_13

[110] Stefania Gnesi and Tiziana Margaria (Eds.). 2013. *Formal Methods for Industrial Critical Systems: A Survey of Applications.* Wiley, UK. doi:10.1002/9781118459898

[111] J. F. Groote, S. F. M. van Vlijmen, and J. W. C. Koorn. 1995. The Safety Guaranteeing System at Station Hoorn-Kersenboogerd. In *Proceedings of the 10th Annual Conference on Computer Assurance Systems Integrity, Software Safety and Process Security (COMPASS'95)*. IEEE, USA, 57–68. doi:10.1109/CMPASS.1995.521887

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

[113] Brahim Hamid and Jon Pérez. 2016. Supporting pattern-based dependability engineering via model-driven development: Approach, tool-support and empirical validation. *J. Syst. Softw.* 122 (2016), 239–273. doi:10.1016/j.jss.2016.09.027

[114] Dominik Hansen, Michael Leuschel, Philipp Körner, Sebastian Krings, Thomas Naulin, Nader Nayeri, David Schneider, and Frank Skowron. 2020. Validation and real-life demonstration of ETCS hybrid level 3 principles using a formal B model. *Int. J. Softw. Tools Technol. Transf.* 22, 3 (2020), 315–332. doi:10.1007/s10009-020-00551-6

[115] Dominik Hansen, David Schneider, and Michael Leuschel. 2016. Using B and ProB for Data Validation Projects. In *Proceedings of the 5th International Conference on Abstract State Machines, Alloy, B, TLA, VDM, and Z (ABZ'16) (LNCS, Vol. 9675)*, Michael J. Butler, Klaus-Dieter Schewe, Atif Mashkoor, and Miklós Biró (Eds.). Springer, Germany, 167–182. doi:10.1007/978-3-319-33600-8_10

[116] Vicky Hartonas-Garmhausen, Thomas R. Kurfess, Edmund M. Clarke, and David E. Long. 1995. Automatic verification of industrial designs. In *Proceedings of the Workshop on Industrial-Strength Formal Specification Techniques (WIFT'95)*. IEEE, USA, 88–96. doi:10.1109/WIFT.1995.515481

[117] Anne E. Haxthausen. 2010. Towards a Framework for Modelling and Verification of Relay Interlocking Systems. In *Revised Selected Papers of the 16th Monterey Workshop on Foundations of Computer Software: Modeling, Development, and Verification of Adaptive Systems (LNCS, Vol. 6662)*, Radu Calinescu and Ethan K. Jackson (Eds.). Springer, Germany, 176–192. doi:10.1007/978-3-642-21292-5_10

[118] Anne E. Haxthausen. 2012. Automated Generation of Safety Requirements from Railway Interlocking Tables. In *Proceedings of the 5th International Symposium on Leveraging Applications of Formal Methods, Verification and Validation: Applications and Case Studies (ISoLA'12) (LNCS, Vol. 7610)*, Tiziana Margaria and Bernhard Steffen (Eds.). Springer, Germany, 261–275. doi:10.1007/978-3-642-34032-1_25

[119] Anne E. Haxthausen, Marie Le Bliguet, and Andreas A. Kjær. 2008. Modelling and Verification of Relay Interlocking Systems. In *Revised Selected Papers of the 15th Monterey Workshop on Foundations of Computer Software: Future Trends and Techniques for Development (LNCS, Vol. 6028)*, Christine Choppy and Oleg Sokolsky (Eds.). Springer, Germany, 141–153. doi:10.1007/978-3-642-12566-9_8

[120] Anne E. Haxthausen and Alessandro Fantechi. 2023. Compositional Verification of Railway Interlocking Systems. *Form. Asp. Comput.* 35, 1 (2023), 4:1–4:46. doi:10.1145/3549736

[121] Anne E. Haxthausen and Jan Peleska. 2013. Efficient Development and Verification of Safe Railway Control Software. In *Railways: Types, Design and Safety Issues*, Cacilie Reinhardt and Klaus Shroeder (Eds.). Nova Science, USA, Chapter 5, 127–148.

[122] Ouail Himrane, Julie Beugin, and Mohamed Ghazel. 2021. Toward Formal Safety and Performance Evaluation of GNSS-based Railway Localisation Function. *IFAC-Pap.* 54, 2 (2021), 159–166. doi:10.1016/j.ifacol.2021.06.049 Proceedings of the 16th IFAC Symposium on Control in Transportation Systems (CTS'21).

[123] Ouail Himrane, Julie Beugin, and Mohamed Ghazel. 2023. Implementation of a Model-Oriented Approach for Supporting Safe Integration of GNSS-Based Virtual Balises in ERTMS/ETCS Level 3. *IEEE Open J. Intell. Transp. Syst.* 4 (2023), 294–310. doi:10.1109/OJITS.2023.3267142

[124] 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. doi:10.1109/TITS.2014.2366512

[125] Gerard J. Holzmann. 2003. *The SPIN Model Checker: Primer and Reference Manual*. Addison-Wesley, USA.

[126] Linh Vu Hong, Anne E. Haxthausen, and Jan Peleska. 2017. Formal modelling and verification of interlocking systems featuring sequential release. *Sci. Comput. Program.* 133 (2017), 91–115. doi:10.1016/j.scico.2016.05.010

[127] Lujiang Huang. 2020. The Past, Present and Future of Railway Interlocking System. In *Proceedings of the 5th IEEE International Conference on Intelligent Transportation Engineering (ICITE'20)*. IEEE, USA, 170–174. doi:10.1109/ICITE50838.2020.9231438

[128] Alexei Iliasov, Dominic Taylor, Linas Laibinis, and Alexander Romanovsky. 2023. Practical Verification of Railway Signalling Programs. *IEEE Trans. Dependable Secur. Comput.* 20, 1 (2023), 695–707. doi:10.1109/TDSC.2022.3141555

[129] Phillip James, Faron Moller, Nguyen Hoang Nga, Markus Roggenbach, Steve Schneider, and Helen Treharne. 2014. Techniques for modelling and verifying railway interlockings. *Int. J. Softw. Tools Technol. Transf.* 16, 6 (2014), 685–711. doi:10.1007/S10009-014-0304-7

[130] Phillip James, Faron Moller, Hoang Nga Nguyen, Markus Roggenbach, Helen Treharne, and Xu Wang. 2016. OnTrack: The Railway Verification Toolset. In *Proceedings of the 7th International Symposium on Leveraging Applications of Formal Methods, Verification and Validation: Discussion, Dissemination, Applications (ISoLA'16) (LNCS, Vol. 9953)*, Tiziana Margaria and Bernhard Steffen (Eds.). Springer, Germany, 294–296. doi:10.1007/978-3-319-47169-3_21

[131] Adam Tauman Kalai, Ofir Nachum, Santosh S. Vempala, and Edwin Zhang. 2025. Why Language Models Hallucinate. doi:10.48550/arXiv.2509.04664 arXiv:2509.04664

[132] 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, Vol. 873)*, Maurice Naftalin, B. Tim Denvir, and Miquel Bertran (Eds.). Springer, Germany, 45–54. doi:10.1007/3-540-58555-9_86

[133] Amruth N. Kumar, Rajendra K. Raj, Sherif G. Aly, Monica D. Anderson, Brett A. Becker, Richard L. Blumenthal, Eric Eaton, Susan L. Epstein, Michael Goldweber, Pankaj Jalote, Douglas Lea, Michael Oudshoorn, Marcelo Pias, Susan Reiser, Christian Servin, Rahul Simha, Titus Winters, and Qiao Xiang. 2024. *Computer Science Curricula 2023*. ACM, IEEE, and AAAI, USA. doi:10.1145/3664191

[134] Thierry Lecomte. 2008. Safe and Reliable Metro Platform Screen Doors Control/Command Systems. In *FM 2008 (LNCS, Vol. 5014)*, Jorge Cuéllar, Tom Maibaum, and Kaisa Sere (Eds.). Springer, Germany, 430–434. doi:10.1007/978-3-540-68237-0_32

[135] Thierry Lecomte. 2009. Applying a Formal Method in Industry: A 15-Year Trajectory. In *Proceedings of the 14th International Workshop on Formal Methods for Industrial Critical Systems (FMICS'09) (LNCS, Vol. 5825)*, María Alpuente, Byron Cook, and Christophe Joubert (Eds.). Springer, Germany, 26–34. doi:10.1007/978-3-642-04570-7_3

[136] Thierry Lecomte. 2014. Return of Experience on Automating Refinement in B. In *Proceedings of the 1st International Workshop about Sets and Tools (SETS'14)*, David Delahaye and Catherine Dubois (Eds.). CNAM, France, 57–68. http://sets2014.cnam.fr/papers/00010057.pdf

[137] Thierry Lecomte, Lilian Burdy, and Michael Leuschel. 2020. Formally Checking Large Data Sets in the Railways. doi:10.48550/arXiv.1210.6815 arXiv:1210.6815 Proceedings of the ICFEM'12 Workshop on the experience of and advances in developing dependable systems in Event-B (DS-Event-B'12).

[138] Thierry Lecomte, David Deharbe, Paulin Fournier, and Marcel Oliveira. 2020. The CLEARSY safety platform: 5 years of research, development and deployment. *Sci. Comput. Program.* 199 (2020), 102524. doi:10.1016/j.scico.2020.102524

[139] 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, Vol. 10623)*, Simone Cavalheiro and José Fiadeiro (Eds.). Springer, Germany, 70–87. doi:10.1007/978-3-319-70848-5_6

[140] Axel Legay, Anna Lukina, Louis-Marie Traonouez, Junxing Yang, Scott A. Smolka, and Radu Grosu. 2019. Statistical Model Checking. In *Computing and Software Science: State of the Art and Perspectives*, Bernhard Steffen and Gerhard J. Woeginger (Eds.). LNCS, Vol. 10000. Springer, Germany, 478–504. doi:10.1007/978-3-319-91908-9_23

[141] Michael Leuschel and Michael J. Butler. 2008. ProB: an automated analysis toolset for the B method. *Int. J. Softw. Tools Technol. Transf.* 10, 2 (2008), 185–203. doi:10.1007/s10009-007-0063-9

[142] Christophe Limbrée and Charles Pecheur. 2018. A Framework for the Formal Verification of Networks of Railway Interlockings - Application to the Belgian Railway. *Electron. Commun. EASST* 76 (2018), 17 pages. doi:10.14279/tuj.eceasst.76.1077 Proceedings of the 18th International Workshop on Automated Verification of Critical Systems (AVoCS'18).

[143] 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. doi:10.1016/j.ejor.2017.07.044

[144] Amel Mammar, Marc Frappier, Steve J. Tueno Fotso, and Régine Laleau. 2020. A formal refinement-based analysis of the hybrid ERTMS/ETCS level 3 standard. *Int. J. Softw. Tools Technol. Transf.* 22, 3 (2020), 333–347. doi:10.1007/s10009-019-00543-1

[145] 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. doi:10.1109/TITS.2017.2658179

[146] Franco Mazzanti, Alessio Ferrari, and Giorgio Oronzo Spagnolo. 2016. Experiments in Formal Modelling of a Deadlock Avoidance Algorithm for a CBTC System. In *ISoLA 2016 (LNCS, Vol. 9953)*, Tiziana Margaria and Bernhard Steffen (Eds.). Springer, Germany, 297–314. doi:10.1007/978-3-319-47169-3_22

[147] 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. doi:10.1007/s10009-018-0488-3

[148] Ahmed Mekki, Mohamed Ghazel, and Armand Toguyéni. 2012. Validation of a New Functional Design of Automatic Protection Systems at Level Crossings with Model-Checking Techniques. *IEEE Trans. Intell. Transp. Syst.* 13, 2 (2012), 714–723. doi:10.1109/TITS.2011.2178238

[149] 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)*. Cyber-Physical Systems Virtual Organization, USA, 25 pages. http://cps-vo.org/node/3434

[150] 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. doi:10.1109/MCOM.2015.7295465

[151] Leonardo de Moura and Nikolaj Bjørner. 2008. Z3: An Efficient SMT Solver. In *Proceedings of the 14th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'08) (LNCS, Vol. 4963)*, C. R. Ramakrishnan and Jakob Rehof (Eds.). Springer, Germany, 337–340. doi:10.1007/978-3-540-78800-3_24

[152] Monty Newborn. 2001. *Automated Theorem Proving*. Springer, Germany. doi:10.1007/978-1-4613-0089-2

[153] Tobias Nipkow, Lawrence C. Paulson, and Markus Wenzel (Eds.). 2002. *Isabelle/HOL: A Proof Assistant for Higher-Order Logic*. LNCS, Vol. 2283. Springer, Germany. doi:10.1007/3-540-45949-9

[154] Ammar Osaiweran, Mathijs Schuts, Jozef Hooman, Jan Friso Groote, and Bart J. van Rijnsoever. 2016. Evaluating the effect of a lightweight formal technique in industry. *Int. J. Softw. Tools Technol. Transf.* 18, 1 (2016), 93–108. doi:10.1007/S10009-015-0374-1

[155] Jörn Pachl. 2018. *Railway Operation and Control* (4 ed.). VTD Rail Publishing, USA.

[156] Camille Parillaud, Yoann Fonteneau, and Fabien Belmonte. 2019. Interlocking Formal Verification at Alstom Signalling. 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)*, Simon Collart-Dutilleul, Thierry Lecomte, and Alexander Romanovsky (Eds.). Springer, Germany, 215–225. doi:10.1007/978-3-030-18744-6_14

[157] 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, Vol. C-22)*, Richard L. Tenney, Paul D. Amer, and M. Ümit Uyar (Eds.). North-Holland, The Netherlands, 189–203.

[158] Jan Peleska, Alexander Baer, and Anne E. Haxthausen. 2000. Towards Domain-Specific Formal Specification Languages for Railway Control Systems. *IFAC Proc. Vol.* 33, 9 (2000), 119–124. doi:10.1016/S1474-6670(17)38134-X Proceedings of the 9th IFAC Symposium on Control in Transportation Systems (CTS'00).

[159] Jan Peleska, Daniel Große, Anne E. Haxthausen, and Rolf Drechsler. 2004. Automated Verification for Train Control Systems. In *Proceedings of Formal Methods for Automation and Safety in Railway and Automotive Systems (FORMS/FORMAT'04)*, E. Schnieder and G. Tarnai (Eds.). Technical University of Braunschweig, Germany, 252–265.

[160] Jan Peleska, Niklas Krafczyk, Anne E. Haxthausen, and Ralf Pinger. 2021. Efficient Data Validation for Geographical Interlocking Systems. *Form. Asp. Comput.* 33, 6 (2021), 925–955. doi:10.1007/s00165-021-00551-6

[161] 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, Vol. 156)*, René Jacquart (Ed.). Springer, Germany, 629–636. doi:10.1007/978-1-4020-8157-6_61

[162] Prover. 2020. Our model checker PSL up to 60 times faster using AI-tuned proof tactics. Blog. https://www.prover.com/quality/psl-up-to-sixty-times-faster-using-ai-optimized-proof-strategies/

[163] Prover. 2024. Prover takes railway Signaling Design Automation to the next level with AI and formal methods. News. https://www.prover.com/formal-methods/railway-signaling-design-automation-with-ai-and-formal-methods/

[164] J. Alan Robinson and Andrei Voronkov (Eds.). 2001. *Handbook of Automated Reasoning*. Elsevier, The Netherlands. https://www.sciencedirect.com/book/9780444508133/handbook-of-automated-reasoning

[165] 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/

[166] Denis Sabatier. 2016. Using Formal Proof and B Method at System Level for Industrial Projects. In *Proceedings of the 1st International Conference on Reliability, Safety, and Security of Railway Systems: Modelling, Analysis, Verification, and Certification (RSSRail'16) (LNCS, Vol. 9707)*, Thierry Lecomte, Ralf Pinger, and Alexander Romanovsky (Eds.). Springer, Germany, 20–31. doi:10.1007/978-3-319-33951-1_2

[167] Denis Sabatier, Lilian Burdy, Antoine Requet, and Jérôme Guéry. 2012. Formal Proofs for the NYCT Line 7 (Flushing) Modernization Project. In *Proceedings of the 3rd International Conference on Abstract State Machines, Alloy, B, VDM, and Z (ABZ'12) (LNCS, Vol. 7316)*, John Derrick, John S. Fitzgerald, Stefania Gnesi, Sarfraz Khurshid, Michael Leuschel, Steve Reeves, and Elvinia Riccobene (Eds.). Springer, Germany, 369–372. doi:10.1007/978-3-642-30885-7_34

[168] Shubhangi Salunkhe, Randolf Berglehner, and Abdul Rasheeq. 2021. Automatic Transformation of SysML Model to Event-B Model for Railway CCS Application. In *Proceedings of the 8th International Conference on Rigorous State-Based Methods (ABZ'21) (LNCS, Vol. 12709)*, Alexander Raschke and Dominique Méry (Eds.). Springer, Germany, 143–149. doi:10.1007/978-3-030-77543-8_14

[169] Monika Seisenberger, Maurice H. ter Beek, Xiuyi Fan, Alessio Ferrari, Anne E. Haxthausen, Phillip James, Andrew Lawrence, Bas Luttik, Jaco van de Pol, and Simon Wimmer. 2022. Safe and Secure Future AI-Driven Railway Technologies: Challenges for Formal Methods in Railway. In *Proceedings of the 11th International Symposium on Leveraging Applications of Formal Methods, Verification and Validation: Practice (ISoLA'22) (LNCS, Vol. 13704)*, Tiziana Margaria and Bernhard Steffen (Eds.). Springer, Germany, 246–268. doi:10.1007/978-3-031-19762-8_20

[170] Chenglei Si, Navita Goyal, Tongshuang Wu, Chen Zhao, Shi Feng, Hal Daumé III, and Jordan L. Boyd-Graber. 2024. Large Language Models Help Humans Verify Truthfulness— Except When They Are Convincingly Wrong. In *Proceedings of the 17th Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL'24)*, Kevin Duh, Helena Gómez-Adorno, and Steven Bethard (Eds.), Vol. 1. ACL, USA, 1459–1474. doi:10.18653/V1/2024.NAACL-LONG.81

[171] Colin F. Snook, Thai Son Hoang, Dana Dghaym, Asieh Salehi Fathabadi, and Michael J. Butler. 2021. Domain-specific scenarios for refinement-based methods. *J. Syst. Archit.* 112 (2021), 101833. doi:10.1016/j.sysarc.2020.101833

[172] Gunnar Stålmarck. 1992. A system for determining propositional logic theorems by applying values and rules to triplets that are generated from a formula. Swedish Patent No. 467 076. International Equivalents: U.S. Patent No. 5 276 897 (1994), European Patent No. 0403 454 (approved 1995).

[173] Gunnar Stålmarck and Mårten Säflund. 1990. Modeling and Verifying Systems and Software in Propositional Logic. *IFAC Proc. Vol.* 23, 6 (1990), 31–36. doi:10.1016/S1474-6670(17)52173-4 Proceedings 9th IFAC/EWICS/SARS Symposium on Safety of Computer Control Systems (SAFECOMP'90).

[174] 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, UK, 9/1–9/3. doi:10.1016/0141-9331(90)90127-H

[175] Martyn Thomas. 1993. The industrial use of formal methods. *Microprocess. Microsystems* 17, 1 (1993), 31–36. doi:10.1016/0141-9331(93)90091-K

[176] Steve J. Tueno Fotso, Marc Frappier, Régine Laleau, and Amel Mammar. 2020. Modeling the hybrid ERTMS/ETCS level 3 standard using a formal requirements engineering approach. *Int. J. Softw. Tools Technol. Transf.* 22, 3 (2020), 349–363. doi:10.1007/s10009-019-00542-2

[177] 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. doi:10.1155/2016/7308604

[178] Yuemiao Wang, Lei Chen, David Kirkwood, Peng Fu, Jidong Lv, and Clive Roberts. 2018. Hybrid Online Model-Based Testing for Communication-Based Train Control Systems. *IEEE Intell. Transp. Syst. Mag.* 10, 3 (2018), 35–47. doi:10.1109/MITS.2018.2842230

[179] Jeannette M. Wing. 1990. A Specifier's Introduction to Formal Methods. *IEEE Comput.* 23, 9 (1990), 8–24. doi:10.1109/2.58215

[180] Kirsten Winter. 2012. Optimising Ordering Strategies for Symbolic Model Checking of Railway Interlockings. In *ISoLA 2012 (LNCS, Vol. 7610)*, Tiziana Margaria and Bernhard Steffen (Eds.). Springer, Germany, 246–260. doi:10.1007/978-3-642-34032-1_24

[181] Kirsten Winter and Neil J. Robinson. 2003. Modelling Large Railway Interlockings and Model Checking Small Ones. In *Proceedings of the 26th Australasian Computer Science Conference (ACSC'03) (CRPIT, Vol. 16)*, Michael J. Oudshoorn (Ed.). Australian Computer Society, Australia, 309–316. http://crpit.com/confpapers/CRPITV16Winter.pdf

[182] 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. doi:10.1145/1592434.1592436

[183] Daohua Wu and Eckehard Schnieder. 2018. Scenario-based system design with colored Petri nets: an application to train control systems. *Softw. Syst. Model.* 17, 1 (2018), 295–317. doi:10.1007/s10270-016-0517-1

[184] X2Rail-2. 2018. Deliverable D5.1, Formal Methods (Taxonomy and Survey), Proposed Methods and Applications. https://projects.shift2rail.org/download.aspx?id=b4cf6a3d-f1f2-4dd3-ae01-2bada34596b8

[185] 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. doi:10.1016/j.comcom.2016.04.003

[186] Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli. 2024. Hallucination is Inevitable: An Innate Limitation of Large Language Models. doi:10.48550/arXiv.2401.11817 arXiv:2401.11817

[187] Kaiyu Yang, Aidan M. Swope, Alex Gu, Rahul Chalamala, Peiyang Song, Shixing Yu, Saad Godil, Ryan J. Prenger, and Animashree Anandkumar. 2023. LeanDojo: Theorem Proving with Retrieval-Augmented Language Models. In *Proceedings of the 37th Conference on Neural Information Processing Systems (NeurIPS'23)*. Curran Associates, USA, 21573–21612. https://papers.neurips.cc/paper_files/paper/2023/file/4441469427094f8873d0fecb0c4e1cee-Paper-Datasets_and_Benchmarks.pdf

[188] Naijun Zhan, Jim Woodcock, Ji Wang, and Mingshuai Chen. 2026. A Brief History of Formal Methods in China. *Form. Asp. Comput.* (2026). doi:10.1145/3783996

Received 10 August 2025; revised 10 March 2026; accepted 10 March 2026