Interlocking Formal Verification at Alstom Signalling

Camille Parillaud[1(✉)], Yoann Fonteneau[2], and Fabien Belmonte[1]

[1] Alstom Transport SA., 48 rue Albert Dhalenne, Saint-Ouen, France
camille.parillaud@alstomgroup.com
[2] Systerel, Toulouse, France

Abstract. Over the past decade, the growing number of safety-critical software in the railway signalling industry has led customers and industrials to look for efficient, cost-effective, verification and validation techniques. Formal methods, which have proven to be applicable and beneficial in terms of accuracy and completeness, are good candidates. However, they are still far from being used systematically for the verification of all safety-critical railway signalling systems. In order to evaluate their applicability, Alstom successfully experimented on its interlocking systems the model checking methods and tools developed by Systerel. This article describes the methodology used to industrialize this experimental model checking application process.

Keywords: Railway signalling · Formal verification · Industrial usage · Interlocking · Safety-critical systems

1 Introduction

This article presents the interlocking formal verification performed at Alstom Signalling using Systerel Smart Solver (S3) model checking solution. Applying formal verification to interlocking systems is not new, several use cases are well known in the railway signalling domain (as shown in previous communication made by Systerel [2]). However, formal methods are neither used for all interlocking systems nor for all signalling applications. This article argues that formal verification of interlocking system is a step forward to introduce the recent development of formal methods (such as optimization of model checking tools) in railway signalling applications. This is a first step towards building an industry-specific methodology. It starts with a presentation of the industrial issues of applying model checking to interlocking systems (Sect. 2), followed by a brief state-of-the-art (Sect. 3). Then the technical issue is described in Sect. 4. Section 5 explains how model checking has been introduced in Alstom's interlocking verification process before presenting the results in Sect. 6 and concluding in Sect. 7.

S. Collart-Dutilleul et al. (Eds.): RSSRail 2019, LNCS 11495, pp. 215–225, 2019.
https://doi.org/10.1007/978-3-030-18744-6_14

2 Industrial Issues

Alstom's signalling systems span from wayside equipment such as track circuits to complete signalling solutions such as Communication Based Train Control (CBTC) systems. The number of installed equipment since the 1970s grows each year and in particular the number of integrated solutions. For instance, the Urbalis 400 CBTC solution equips 56 lines around the world today and will soon be deployed on another 54 lines. This extensive number of systems in operation increases the risk exposure to safety-related hazards and potential accidents; hence the need for efficient verification and validation techniques.

Software development of signalling subsystems has intensively benefited from formal methods such as the B-Method which was used by Alstom to develop the safety critical software of its mainline and urban Automatic Train Protection Systems. However, formal methods have not been used at Alstom to develop legacy subsystems such as interlocking since they rely on old principles inherited from relay logic. Verification and validation of these old principles are performed by highly skilled individuals and knowledge management of these skills is hard to maintain. Moreover, several projects require the installation of CBTC systems interfaced with pre-existing relay-based interlocking systems which must be adapted to the CBTC features. Alstom looked for a way to capture and formalize critical knowledge of such systems, as well as an efficient method to validate the new and the pre-existing interlocking systems.

2.1 Limits and Difficulties of Classical Verification and Validation Process

The classical verification and validation process of interlocking subsystems relies on wide testing campaigns performed on virtual stations. These stations are conceived so that as many functional scenarios as possible are included. Their design and implementation are difficult and time-consuming activities. It is especially hard to demonstrate that all safety-related scenarios have been correctly tested. Indeed, some ripple effects due to modifications can be hard to foresee and test on a virtual station. When dealing with existing relay-based interlocking systems which have been improved and optimized over the years, extracting the principles and the associated safety concepts is a challenging task. The assessment of the potential impact on the global system functions and safety requires important effort.

2.2 Expected Results from Formal Verification

For these reasons, Alstom is introducing formal methods in the verification and validation process of its interlocking systems. Indeed, these methods are based on mathematical logic and they ensure an exhaustive and sound verification of the system. The objective is to formally verify conventionally developed interlocking systems whether they are computer-based or relay-based, prior to site operation. Model checking is particularly suitable to verify that systems always

satisfy a set of properties and since the development of the interlocking system is already performed, formal development is not adapted. This is why model checking has been favoured to other formal methods. The use of model checking in Alstom's interlocking verification process will be presented in the following sections of this article. The expected benefits are numerous and they will improve the competitiveness of the system on which the formal methods were used.

Exhaustive and Unambiguous. As previously mentioned, formal methods are founded on mathematical theories. They aim at building precise models of the system or software under development. Their common objective is to eliminate any ambiguity or imprecision which may come from the use of natural language in order for the results to be unambiguous. They are sound and ensure that the system is exhaustively verified or proved. Consequently, using formal methods eases the approval process of the system.

Shortened Time-to-Market and Costs Reduction. In the railway domain, the majority of safety-related faults which are discovered late in the lifecycle of the project (which are therefore problematic) are linked to unlikely scenarios that were not foreseen beforehand. The completeness of the proof or verification obtained with the use of formal methods allows discovering these safety-related scenarios early in the development lifecycle. The necessary modifications can then be made earlier and the amount of required rework will be limited. This implies a substantial amount of time saved and costs reduction at project level.

2.3 Limitations

Despite the significant advantages they present, formal methods also have their limitations. The first limitation is that the proof (or verification) performed formally is based on a set of safety properties that are manually determined beforehand. If this set of properties is erroneous or incomplete, the value of the formal proof (or verification) is of little use. It is therefore crucial to establish a robust process to list the necessary and sufficient safety properties to be proven. The process Alstom uses is presented in Sect. 5. Moreover, formal methods are efficient when they contribute to the safety demonstration of the system. However, when it comes to proving non safety-related properties, these are very complicated to determine as they must include all possible functional requirements cases to be provable.

3 State of the Art

Interlocking systems, with their inherent boolean nature and overwhelming combinational complexity have been a privileged target of model checking techniques. Pioneering work started as soon as sufficiently powerful model-checker software came into existence in the early 2000s (*e.g.* [1,3,8]). All these contributions were

analyzing manually crafted models of interlocking systems, somewhat distant from the real installed safety critical systems. Moreover, at that time, it had always been concluded that the huge state space of these systems could not be handled without over-simplifications and/or splitting and compositional verification techniques.

Over the past decade, model checking techniques have matured, increasing the analysis power. After 2010, a renewed interest arose, leading to novel attempts to solve the problem (*e.g.* [4–6]). Great progress has been made that demonstrates the feasibility of the formal safety verification of real-world interlocking systems.

In this article, following the Systerel Smart Solver (S3) workflow presented in a previous article (see [2]), we describe the use of these techniques in an industrial and normative context.

4 Technical Issue

The formal safety verification of an interlocking system requires both a solution to perform the analyses and the safety properties to be analysed under a number of environment constraints. This section focuses on the formal verification solution, and Sect. 5 describes the safety properties.

Following the description given in [2], an S3 formal safety verification solution involves the development of a translator from a given interlocking application (given in its specific language/format) to a model of this application in HLL, the S3 tool-chain input modelling language [7]. This model of the safety critical application shall be *sound*, in that it shall preserve the semantics of the real application, so that any property proved on this model is valid on the real application. A second translator is also needed to translate a description of the track layout controlled by the given interlocking application. These data, usually given in some form of database, contain the objects present on the tracks (*e.g.* signals, points, routes, ...), and relations between these objects (*e.g.* origin signal of a route, points of a route, ...). They are translated in HLL as hierarchical enumerations of objects, and predicates on these objects. These two translators are specific to the given family of interlocking applications.

The obtained HLL models can then be concatenated with the desired safety properties and environment constraints formalized in HLL to obtain the analysis model. This model can then be analyzed by the standard S3 tool-chain. It is first given to an expander tool that transforms the HLL model into a semantically equivalent model in LLL, a purely boolean subset of HLL suitable for S3 analysis. Two main types of analyses may be performed.

Bounded Model Checking (BMC). In this type of analysis, scenarios of increasing length are investigated in order to find counter-examples of some of the provided safety properties. Such a scenario, exercising the inputs of the interlocking application (*i.e.* its sensors) in a way compatible with the provided environment constraints, leads the application from its initial state to a state

violating a safety property. For each safety property, the result of this analysis is thus either a scenario violating the property, or the assurance that this property holds for every scenario up to a given length (the higher the length, the longer the analysis will take). This type of analysis is the first one to be attempted on a new system or a new property, until no more counter-examples can be found, and the BMC has reached a length large enough to have an intimate conviction that the non-violated properties hold.

Induction over Time. In this mode, the analysis engine attempts to prove that some safety property is valid, which means that there exist no scenario, whatever the length, that leads the interlocking application to a violation of this safety property (e.g [2,4]). This is performed using standard induction over the length of the scenario. A first analysis shows that the property holds for every scenario of length 1, and a second shows that if the property holds in some state of the system, it will hold in any state reachable from this state in one transition of the interlocking application. When these two analyses are successful, the property is proved valid. If the first analysis fails, a counter-example to the property has been found (similarly to the BMC analyses). However, when the second analysis fails it either means that the property can be falsified with a long scenario (longer than the length reached by a BMC on this property), or more often, that the property is non-inductive. This means that the analysis engine has found a scenario (called a step-counter-example) starting from a state of the system in which the property holds, and which leads with a single transition of the interlocking application to a state violating this property. This means that this starting state is unreachable from the initial state of the system. The way to deal with these non-inductive properties is by developing induction enforcing lemmas, as explained in Sect. 5.

The analysis process starts by using the BMC strategy repeatedly and correcting either the expression of the safety properties or the bugs found in the interlocking application until no more counter-examples are found for a large length. The process then reverts to an induction strategy, used iteratively to find all lemmas until all properties are proved.

However, in the EN50128 normative context, this is not sufficient. This standard asks for some insurance on the results of the verification (T2) tool.

To achieve a high degree of confidence compatible with EN50128, a second set of translators is developed in an independent way (different development team and different programming language), a second independent expander from HLL to LLL is also used. The resulting LLL models of the two translations expanded by the two expanders are combined by a tool that creates a new LLL file expressing that the two models are *sequentially-equivalent* (*i.e.* provided with the same inputs sequences, they produce the same output sequences). This resulting LLL file is then given to the S3 analysis engine to prove the equivalence. Moreover, the S3 analysis engine is equipped with a proof-log/proof-check mechanism, such that for each proof that it finds (proofs of equivalence and of the safety properties), it outputs a proof-log file containing this proof expressed in a formalized

proof system, and the correctness of each proof-log is independently verified by a simple proof-checker software.

Therefore, the S3 solution is compatible with an EN50128 T2 verification tool certification.

5 Industrial Process

5.1 Determination of Safety Properties

The first step towards proving that an interlocking system is safe through model checking is to determine the safety properties that this system must satisfy. These safety properties must be as high-level as possible in order to maintain a black box approach and remain independent from the design of the interlocking system. Thus, the safety properties are less likely to be biased and to hide a possibly dangerous scenario. The identification of the adequate safety properties is performed through the "Deductive Identification of Safety Properties" which is a three-step process.

Deductive Tree Analysis. First, a top-down analysis is conducted. It aims at identifying a comprehensive set of high-level functional safety properties to be satisfied by the interlocking system. It is performed independently from the detailed design, i.e. with a black-box approach, knowing only the external interfaces of the interlocking. Thanks to a user-level knowledge of the functions the system must implement and to the definition of its scope, the influence of the system on its external environment is studied based on the two following criteria: What are the hazards that are likely to occur in the scope of the interlocking system? How can the interlocking protect against these hazards by use of its means of interaction with its external environment? This identification of prohibited scenarios allows modelling the hazards associated with the functional behaviour of the interlocking system. The properties, thus specified, ensure that the system does prevent these hazards from occurring.

Failure Modes and Effects Analysis (FMEA). The previous deductive approach has the advantage of being completely independent from the product. However, some risks can originate from the design choices. This is why a FMEA, which is inductive (or bottom-up), is performed. Instead of focusing on the hazards and looking for the possible causes, it aims at determining the possible effects of a failure of each function performed by the interlocking system and defining mitigations should the risk be safety-related.

Convergence. In order to ensure the completeness of the list of safety properties, the two sets of requirements coming both from the deductive and the inductive analyses are traced. This ensures that the high level properties of the system do cover all possible hazards related to the interlocking system.

The safety properties are the result of this traceability. They are based on the wording of the requirements coming from the deductive analysis. Should there be a requirement from the inductive analysis that cannot be traced with any requirement of the deductive analysis, a new safety property is added, based on the formalization of this requirement. Once this last task is performed, the output is a complete set of safety properties expressed in natural language that will be proven with model checking after being formalized.

5.2 Modelling

Environment. In order to adequately simulate the inputs of the interlocking system, a model of its environment is created. This model describes the behaviour of the systems interfaced with the interlocking by constraining their outputs which are inputs of the interlocking system. This prevents impossible scenarios from being considered and allows the proof to focus on realistic ones. For instance, an impossible scenario could be a train not moving continuously along the track.

The environment of the model can also include a similar system to the one that is being proved (two systems managing different geographical parts of the track). In that case, each system is proved separately. If some hypotheses must be made on the behaviour of the first system to prove the second, they must be proved when performing the proof of the first system. As the interlocking conditions are different in the two systems, this methodology does not create any reasoning loop and the proof of both systems stands. The asymmetrical conditions come from track layout deployment rules.

Safety Properties. The model also includes the safety properties that have been previously established. These properties rely on refined concepts that must be formally modelled in order to rigorously remove any ambiguity that could be introduced by using natural language.

Interlocking System. The model of the interlocking application and the track layout data are obtained as described in Sect. 4.

Modelling Risks. In order for the proof to be effective and reliable, some precautions must be taken during the modelling phase. It is necessary for the model to be as permissive as possible. It must allow all possible scenarios to occur, otherwise a safety-related hazard could be missed during the proof process. Thus, the constraints on the inputs must be carefully defined and checked with this risk in mind.

5.3 Proof Process

The proof process is described in the Fig. 1. In this process there are two manual tasks:

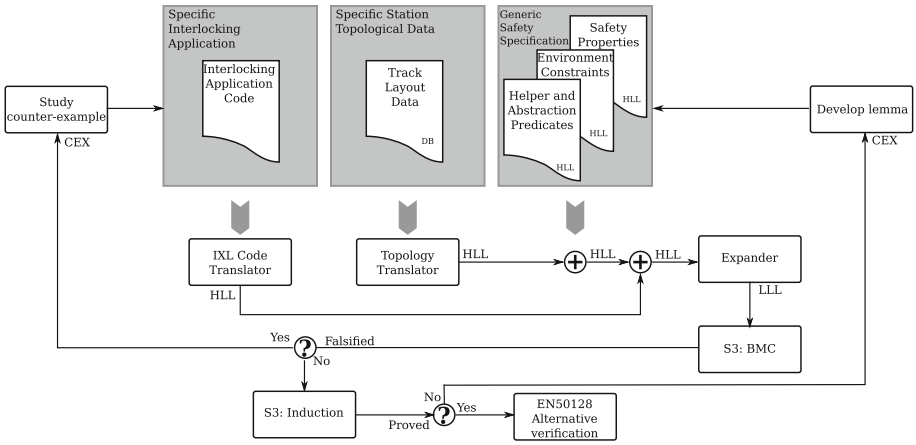

Fig. 1. S3 formal verification process implemented at Alstom

Study of Counter-Examples. This task consists in exploring the counter-example step by step to understand why a property is falsified. The bounded model checking can find problems of three different categories:

- Environment modelling error: the model of the railway environment is too abstract. The error is not reproducible on the track. Some constraints have to be added to remove this behavior. In this case, the environment model has to be fixed.
- Data error: there is an error in the data (chaining error, definition of flanking, ...) or the data of the interlocking are not compatible with some interlocking principles usage restrictions. In this case, the track layout data has to be fixed.
- Principles errors: error in the interlocking principles or the principles are not compatible with the specific track configurations. In this case, the Specific interlocking application has to be fixed.

Development of Lemmas. When a property is not proved, a step-counter-example is generated. This counter-example is used to develop a lemma. A lemma is a relation that holds between variables of the system. It is thus similar in essence to a safety property, except that a property is usually expressed only in terms of the inputs and outputs of the system whereas a lemma may also rely on internal variables. Also, a safety property characterizes some aspect of the safety of the system, whereas a lemma may be more general. Looking at the unreachable state found in a step-counter-example, together with the knowledge of the system design and especially of the principles ensuring its safety, it is usually rather simple to express a relation between variables of the model that eliminates this unreachable state.

5.4 Insertion of Model Checking into Alstom's Pre-existing Process

Interlocking Development Process. Alstom's classical interlocking development process is based on the development of a generic product which is then customized through data to the different specific applications required by commercial projects. This separation between generic product and specific application ensures a high level of reusability of the different activities involved in the development of the specific application (design, validation, safety demonstration). Regarding formal verification, it means that the model of the environment and safety properties, which is based only on the generic product principles and functions, is applicable to all specific applications based on the same version of the generic product. When the interlocking principles are updated, usually in order to incorporate a new functional gap for a client, so is the generic product which in turn means the model has to be adapted as well. The proof, however, is always performed on a specific application as it requires the instantiated principles of the interlocking system.

Occurrence in the Lifecycle of the Interlocking System. The proof is performed on instantiated principles. It can therefore occur as soon as the first version of these instantiated principles is available. The modelling phase can start earlier though, during the design phase, as long as the principles and functions of the system have been defined. The proof must then be repeated for each update of the instantiated principles and system data.

6 Results

6.1 Technical Results

The S3 formal verification solution has been applied on several Smartlock 400 GP interlocking applications on multiple subway lines (Amsterdam, Lusail, Guadalajara...). On the larger stations (1312 routes, 235 points, 398 signals, and 587 secondary detection devices), the analysis took up to 24 h of CPU time on an intel i5-4670. While this duration is acceptable for the long BMC runs used to find falsification of the safety properties, it reveals a burden when it comes to the development of lemmas. For this phase of the project, a custom utility tool has been developed to allow the splitting of a station on a small sub-region of its track layout to allow for faster analysis time. However the final proofs are obviously performed on the whole track-layout.

A total of 114 properties have been formalized, and 533 lemmas were needed to ultimately prove these properties. During the analysis, a total of 5 iterations have been needed to mature the environment modelling (driven by 5 environment modelling errors). The various long BMC runs have unveiled 3 data errors (mainly around the definition of the flanking of points), and a single unlikely principle error. After correction of all these errors, all considered interlocking applications have been proved to respect the safety properties.

6.2 Related to the Industrial Process

As implemented, model checking allows a non intrusive verification of the system design. Thus, the design process of the interlocking system is not impacted by the introduction of formal verification in the safety demonstration process. It is only the verification and validation activities that are impacted as the model checking proof can replace the safety-related tests performed for the interlocking system, that is to say about 30% of the required tests. Indeed, the model of the interlocking system is obtained by two independant translators and a proof of equivalence between the two translations is established, ensuring the model is totally compliant with the source code. Incidentally, the safety demonstration can be provided more quickly when using model checking compared to the classical verification process. This compensates the additional modelling work required by the model checking process.

Moreover, the use of model checking has proven to be beneficial as valid counter-examples (whether they were related to the data or the interlocking principles) were found earlier than with the classical process on the different lines it was tested on. This confirms that introducing model checking in the interlocking system verification and validation process does have added value.

7 Conclusion

Introducing model checking in Alstom's verification and validation process of interlocking systems has proven to be efficient as safety-related counter-examples have been discovered more quickly than with the traditional process. In addition, model checking is performed on a set of instantiated interlocking principles, whereas the traditional verification process uses the generic principles. Therefore, using model checking provides additional confidence in the safety demonstration compared to the traditional process because the proof uses the real data of the specific application. Today, this approach is applied on Alstom's largest interlocking project and the computation time is shorter than a day.

However, the sensibility to complex stations is linked to the lemmas identification for reuse. Indeed, the lemmas that must be defined for the inductive proof can be difficult to find as they must be adequate for all track configurations existing in the specific application. This means they could have to be modified when switching from one application to another.

Overall, this new process was deemed beneficial and will be used on new Alstom interlocking systems in the future.

References

1. Bernardeschi, C., Fantechi, A., Gnesi, S., Mongardi, G.: Proving safety properties for embedded control systems. In: Hlawiczka, A., Silva, J.G., Simoncini, L. (eds.) EDCC 1996. LNCS, vol. 1150, pp. 321–332. Springer, Heidelberg (1996). https://doi.org/10.1007/3-540-61772-8_46

2. Breton, N., Fonteneau, Y.: S3: proving the safety of critical systems. In: Lecomte, T., Pinger, R., Romanovsky, A. (eds.) RSSRail 2016. LNCS, vol. 9707, pp. 231–242. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-33951-1_17
3. Eisner, C.: Using symbolic model checking to verify the railway stations of Hoorn-Kersenboogerd and Heerhugowaard. In: Pierre, L., Kropf, T. (eds.) CHARME 1999. LNCS, vol. 1703, pp. 99–109. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48153-2_9
4. Haxthausen, A.E., Peleska, J., Pinger, R.: Applied bounded model checking for interlocking system designs. In: Counsell, S., Núñez, M. (eds.) SEFM 2013. LNCS, vol. 8368, pp. 205–220. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-05032-4_16
5. James, P., et al.: Verification of solid state interlocking programs. In: Counsell, S., Núñez, M. (eds.) SEFM 2013. LNCS, vol. 8368, pp. 253–268. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-05032-4_19
6. Mota, J.L., et al.: Safety demonstration for a rail signaling application in nominal and degraded modes using formal proof. In: Formal Methods Applied to Industrial Complex Systems, pp. 71–113, July 2014. https://doi.org/10.1002/9781119004707.ch4
7. Ordioni, J., Breton, N., Colaço, J.L.: HLL vol 2.7 modelling language specification. Other STF-16-01805, RATP, May 2018. https://hal.archives-ouvertes.fr/hal-01799749
8. Winter, K.: Model checking railway interlocking systems, February 2002. https://doi.org/10.1145/563857.563836