

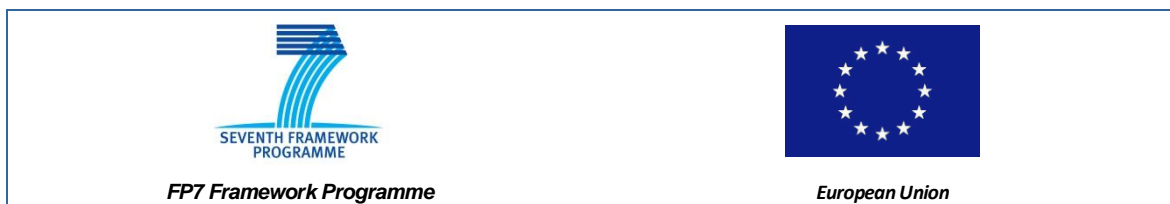
DELIVERABLE D1.1—WORK PACKAGE 1

RAILWAY CASE STUDY DEFINITION

ADVANCE

Grant Agreement: 287563
Date: March 21st 2013
Author: Fernando Mejia, Alstom
Status: Final version
Reviewer: Michael Butler, University of Southampton
Reference: D1.1
Issue: 2

Partners / Clients:



Consortium Members:



Contents

- 1. Introduction3
- 2. Interlocking systems.....4
- 3. Verification of interlocking systems.....5
 - 3.1 Non formal safety verification5
 - 3.2 Formal safety verification6
 - 3.2.1 Event-B approach6
 - 3.2.2 Proofer approach.....6
- 4. Description of the studied solution.....12
 - 4.1 Architecture, design, development and verification principles12
 - 4.2 Development and verification processes.....12
 - 4.3 Rationale13
- 5. Case Study Activities, Process and Methods14
 - 5.1 Demonstration of technical feasibility.....14
 - 5.2 Assessment and Certification.....15
 - 5.3 Full application of ADVANCE methods and tools15
- 6. Evaluation Criteria.....16
- 7. References16

1. Introduction

WP1 of ADVANCE is addressing system-level analysis of safety critical railway interlocking supported by the ADVANCE methods and tools. This deliverable defines the railway interlocking case study being developed in WP1 and the rationale for the approach being followed. Current practice in the railway industry involves verification of specific interlocking configurations based on fixed track layout and route data. The aim in WP1 is to develop an approach based on verification of generic models of interlocking using ADVANCE methods and tools. We outline an existing approach to verification in railway interlocking based on fixed data and outline some weaknesses of the approach. This leads to the rationale for the generic approach being taken in ADVANCE.

To meet the demands of its possible customers, Alstom Information Solutions (ATIS) intends to develop a solution, proposed initially by Systerel, that would permit formal, exhaustive proof that interlocking systems, independent of their size, complexity and implementation technology, contribute to ensure passenger safety by avoiding collisions and derailments of trains due to mismanagement of points and aspects of signals.

Presently, the only industrial certified technology for formal verification of interlocking systems is based on model checking tools developed by Prover Technology¹. This is the case in particular for the Proofer² approach developed and applied by RATP³, Thalès⁴ and Prover Technology for the safety verification of interlocking systems of Paris subway ([1]). This technology has undisputable advantages, it is fully automatic and exhaustive, can be easily integrated in development process and can be applied by non-experts in formal methods. However, in our opinion, it has two major drawbacks. It is highly sensitive in terms of execution time to the size and complexity of the instantiated⁵ interlocking systems that it treats. And it has to take into account very accurately the behaviour of the interlocking execution engine, i.e. the software interpreter that executes the interlocking system code, which, in some cases, can be very complex.

To avoid those drawbacks the proposed solution relies on mathematical proof of generic Event-B ([2]) and Classical-B ([3]) models taking into account safety requirements and is independent from the size and complexity of particular switching stations and from the behaviour of the execution engine of interlocking systems.

The objectives of the case study are to demonstrate the technical feasibility of the proposed solution, to define a process involving ADVANCE methods and tools that would permit certification of the final product in a cost effective way based on reusable verified formal models and to support the consortium in bringing the ADVANCE methods and tools to a technology readiness level permitting to develop the solution in an industrial context. Safety certification is the main driver for the use of formal methods in the railways domain and the development of a strategy for future use of ADVANCE methods and tools in certification is an important goal for WP1. Achieving full certification for the interlocking system addressed in WP1 is outside the scope of ADVANCE since it would require considerably more time to achieve than is available in ADVANCE.

¹ Prover Technology (<http://www.prover.com>) provides solutions for formal verification of signalling systems

² Proofer is the name of an industrial project led by RATP that used the Prover technology.

³ Paris Transport Operator (<http://ratp.fr>)

⁴ Thalès Transport (<http://www.thalesgroup.com/transport>) provides railway signalling system

⁵ Interlocking systems customized with the specific points, signals, routes and so on of a particular switching station.

The document is organized as follows. The next section describes briefly railway signalling systems and in particular interlocking systems. The third section deals with non-formal and formal verification processes of interlocking systems. In particular it describes precisely the Proofer methodology. The fourth section describes the alternative solution proposed by ATIS and Systerel and its verification method based on proof. The fifth section defines the activities that will be carried out in order to achieve the objectives of the case study. The sixth and last section defines the criteria of evaluation of the case study.

2. Interlocking systems

An automatic train management system involves essentially three main components: an automatic train supervision system (ATS), an automatic train control system (ATC) and an interlocking system (IXL).

The ATS system gives to railway operators the global view of the railway system (position of trains, configuration of the track, and so on) and the means to control the destination and the timetables of trains and to ensure good traffic flow. ATS system is not safety-critical because responsibility for safety remains with the IXL and ATC systems.

The ATC system controls and protects train operation. It involves the automatic train operation subsystem (ATO) and the automatic train protection subsystem (ATP).

The ATO subsystem assists, and sometimes replaces, train drivers to operate trains optimally according to timetables, energy consumption, precise stop of trains in stations and so on. ATO system is not safety-critical because local responsibility for safety remains with ATP subsystem.

The ATP subsystem protects the movement of trains, the opening and closing of train and platform doors, the power supply in case of evacuation and so on. Part of an ATP system is installed on board of trains; part is installed in a trackside technical room. ATP subsystem is safety-critical.

The IXL system controls the trackside devices (points, signals, track circuits, various key locks) according to the orders of the railway operator and the movement of trains. Typically, when the railway operator requests the IXL to set and lock a route, the IXL, first checks that this route is compatible with other routes in operation (no risk of collision); second, commands switches along the route to the adequate position; third, lights the aspect of the signal allowing trains to enter the route when all the switches along the route are locked in the adequate position. As and when a train moves along a route, the IXL system releases the locked devices behind the train and lights the aspect of the signal at the entrance of the route forbidding the trains to enter that route. IXL system is safety-critical.

From a design point of view, an IXL system can be considered as a generic abstract machine providing a predefined set of instructions (called "IXL principles") that must be instantiated in order to customise the IXL system for a particular switching station. An example of IXL principle is the rule that governs the control of points. Another example is the rule that governs the release of locked routes. It is important to distinguish between IXL principles and IXL safety properties: safety properties are global system-level properties of a network (such as collision avoidance) while IXL principles are local rules that apply in individual switching points and signals. Existing work on IXL verification is focused on verifying that the principles are correctly implemented. In ADVANCE we are formalising system-level safety properties in Event-B and using refinement to verify that local IXL principles achieve those system-level safety properties.

Distinct formalisms, each with its own execution model, are used to define the dynamic behaviour of principles: Boolean equations, Petri nets, SCADE (cf.[4]), SSI language ([5]), etc. In any case and whatever the formalism, the instance of the IXL system for a particular switching station is implemented by software made of two parts:

- A specific part corresponding to the IXL principles instantiated for the particular switching station.
- A generic part that interprets the instantiated IXL principles, in other words, the execution engine of the formalism used to define the IXL principles.

The execution engine is validated once for all because it is independent from any particular station. By contrast, every instance of the IXL principles for a particular station must be validated separately.

3. Verification of interlocking systems

Interlocking systems are safety-critical given that a failure on their part may result in death or severe injury of people. This is why they are developed and verified with the utmost rigor in order to avoid systematic errors that might result in a catastrophic failure.

ATIS uses the SCADE Suite to develop safety-critical software of IXL systems. This suite allows the interlocking principles to be specified rigorously as communicating automata and to be translated into executable code with a certified translator. Thus at the level of individual IXL units it is possible to guarantee that software of a basic IXL principle implements the specified behaviour of that basic IXL principle.

But, at the global level it is difficult to demonstrate that software resulting from the combination of instantiated basic automata implements exactly the system-level safety properties and is free from errors. The conventional verification methodology based on factory and on-site tests discloses most of the errors but it cannot claim that it discloses all the errors. Only a formal verification can claim to do so.

3.1 Non formal safety verification

ATIS interlocking systems have been installed and certified in numerous countries although no formal techniques were applied for system or software verification. Confidence results from the application of a certified process compliant with CENELEC⁶ standards EN50126, EN50128 and EN50129. This process involves:

- Management plans, including quality, engineering, requirements, configuration, documentation, change control, safety, verification and validation plans;
- Safety Management, including safety verification and validation strategy, safety analyses, hazard log;
- Document Management;
- Requirements Management, including requirements identification, capture and traceability;
- Configuration management;
- Change control management;
- Tool Management;
- Test Management, including regression test strategy, factory and on-site tests.

⁶ Standards applicable in railways for system and software development
Work Package: 1 – Deliverable: D1.1

Safety requirements are identified, apportioned and traced during safety analyses conducted at distinct phases of system development and validation. Preliminary Hazard Analysis (PHA) identifies potential hazards faced by the system and defines system level safety requirements preventing or avoiding potential hazards. System Hazard Analysis (SHA) defines safety requirements for system functions, checks that they close safety requirements of PHA and apportions them between subsystems. Interface Hazard Analysis defines requirements for interfaces between subsystems. All requirements resulting of safety analyses and their closure are logged in the Safety Hazard Log.

In the end, factory and on-site tests as well as validation conducted by qualified personnel provide evidence that interlocking systems meet safety requirements.

3.2 Formal safety verification

3.2.1 Event-B approach

Prior to ADVANCE, some attempts have been made to prove that an Event-B model of an IXL system meets system safety requirements. They face two difficulties. The first one is to establish a formal provable link between the actual behaviour of IXL principles and system-level safety requirements. The former are very concrete and operate at a local level while the latter are rather abstract and global. This is a challenge that we are addressing in ADVANCE by improving the expressivity of the modelling language and by improving support for model decomposition. The second difficulty is that it is hardly possible, and may be impossible with the available tools, to prove that an existing IXL implementation satisfies an Event-B specification. The best that may be possible do is to prove a generic model of the IXL software satisfies the safety properties. But then it is difficult to demonstrate the equivalence between the proved software model and the actual operating interlocking system. As will be explained in Section 4, we are addressing this challenge.

3.2.2 Proofer approach

RATP, Thalès and Prover Technology developed the Proofer methodology for formal verification of IXL systems. Proofer methodology is based on model checking tools and was successfully applied by for the verification of several IXL systems of Paris subway.

Thus it is important to understand this methodology if we want to propose an equivalent verification methodology in spite of different underlying techniques.

3.2.2.1 Proofer Process

The Proofer process involves five steps.

1. The first step identifies the properties the IXL system must meet in order to prevent any undesired event leading to death or severe injury of people. It is a hazard analysis that discloses the immediate causes of the catastrophic accidents, refines them according to the functional specification and identifies the subsystems that are responsible according to the architecture specification.

The results of this step are the IXL safety properties expressed in terms of high-level concepts arranged in fault trees. The figure below, extracted from [6], represents the fault tree resulting from the first phase.

2. The second step refines informally the fault tree describing the high-level IXL safety properties in terms of system concepts in order to define the IXL safety properties in terms of the IXL principles.
3. The third step creates three models that put together form the model of an instantiated IXL system to be verified. By instantiated we mean fixing the specific track layout, specific routes, specific points and specific signals. These models are: the model of the instantiated IXL safety

properties, the model of the instantiated IXL principles and the model of the environment in which the IXL system operates. The latter describes the track layout and the behaviour of trackside devices. The models are written in a Linear Temporal Logic language handling Boolean variables.

4. The fourth step is a non-certified proof process. Its purpose is to correct and complete the models created in the previous step. It is an assisted iterative analysis depicted in Figure 3 below. These models are written in Linear Temporal Logic handling Boolean variables.

First, the three models are translated automatically into a model representing the actual execution tree of the instantiated IXL system. This model is written in TECLA, the language of the proof tool developed by Prover Technology.

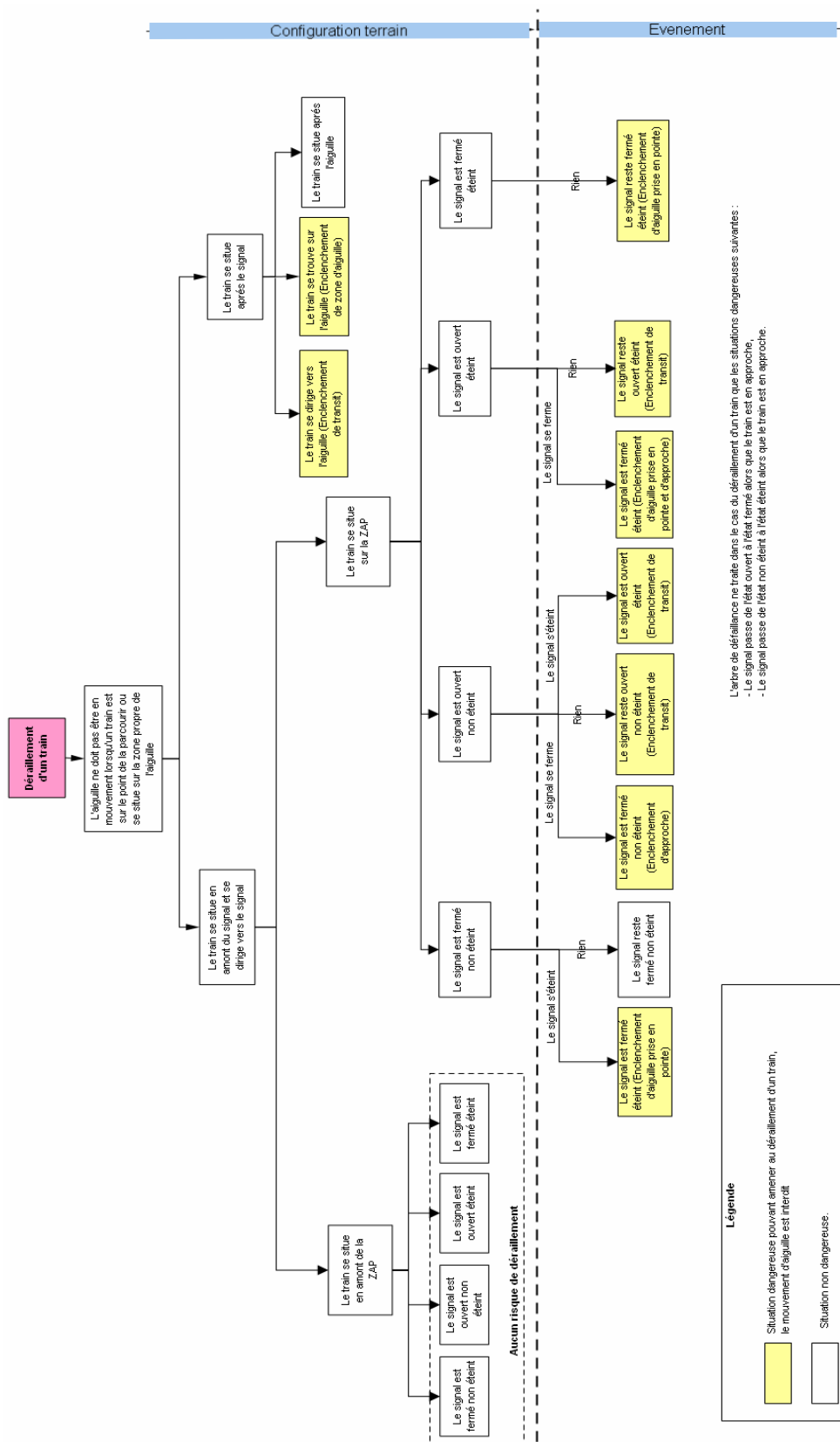


Fig. 2: Fault tree resulting from System Hazard Analysis

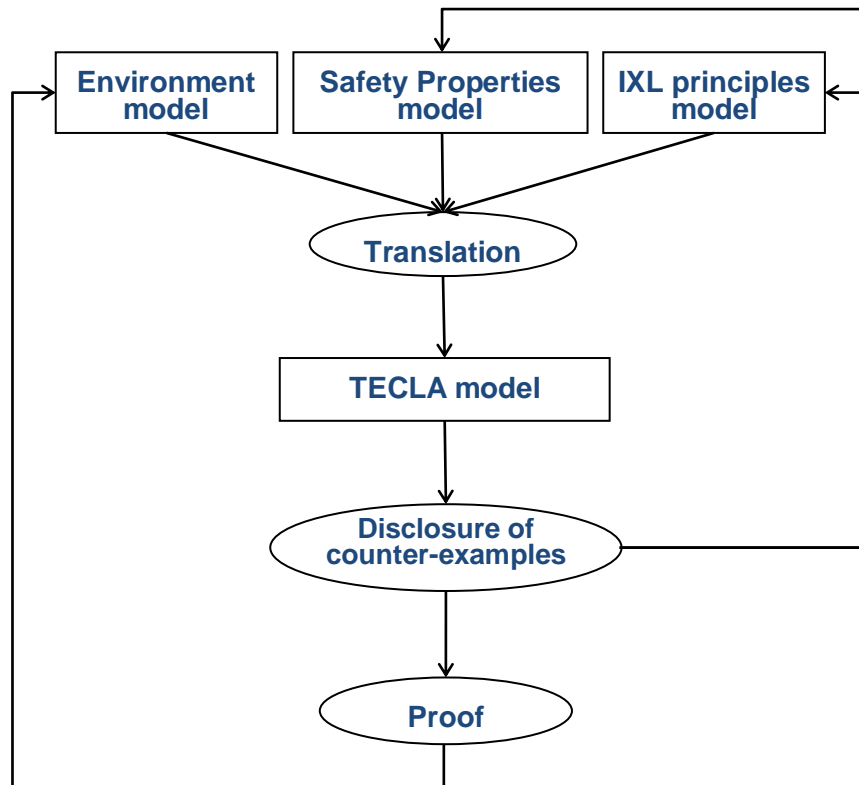


Fig. 3 : Proofer non certified proof process

Then, this proof tool verifies automatically up to a predefined depth of the execution tree whether the computed states of the IXL principles meet the safety properties. If this partial proof discloses a state that violates a safety property the tool delivers the counter-example, i.e. the sequence of states of the IXL principles resulting in that faulty state. A human verifier analyses then this sequence in order to determine the origin of the flaw, correct the appropriate model and, if necessary, the specification of the IXL system. This analysis can be performed by a railway engineer unaware of Linear Temporal Logic because the sequence of states reflects meaningful signalling situations.

The comprehensive proof of the models starts when no counter-example is disclosed by the partial proof. If the comprehensive proof fails the tool exhibits again the disclosed counter-example and the verifier requests the modification of the concerned models. The fourth step terminates when the comprehensive proof discloses no counter-example. Its outputs are correct models of IXL safety properties, of IXL system environment and of the IXL principles.

5. The fifth step is the certified proof process. Its purpose is to verify with a certified tool that, considering the properties and the environment models developed in the previous step, the model of the instantiated IXL principles developed in the previous step and the model of the actual executable programs of the IXL system are equivalent. This step is a fully automated certified activity depicted in Figure 4 below.

The models of the IXL principles, of the IXL safety properties and of the environment are first, translated into a high-level language (HLL) and then expanded into a low-level language (LLL). This corresponds to left branch of Figure 4.

The model of the binary code of the IXL software together with the models of the IXL safety properties and of the IXL environment are in turn translated into the high-level language and then expanded into the low-level language. This corresponds to right branch of Figure 4.

Thus, the low level models of the instantiated IXL principles and of the actual executable programs of the IXL software are brought into a common formalism.

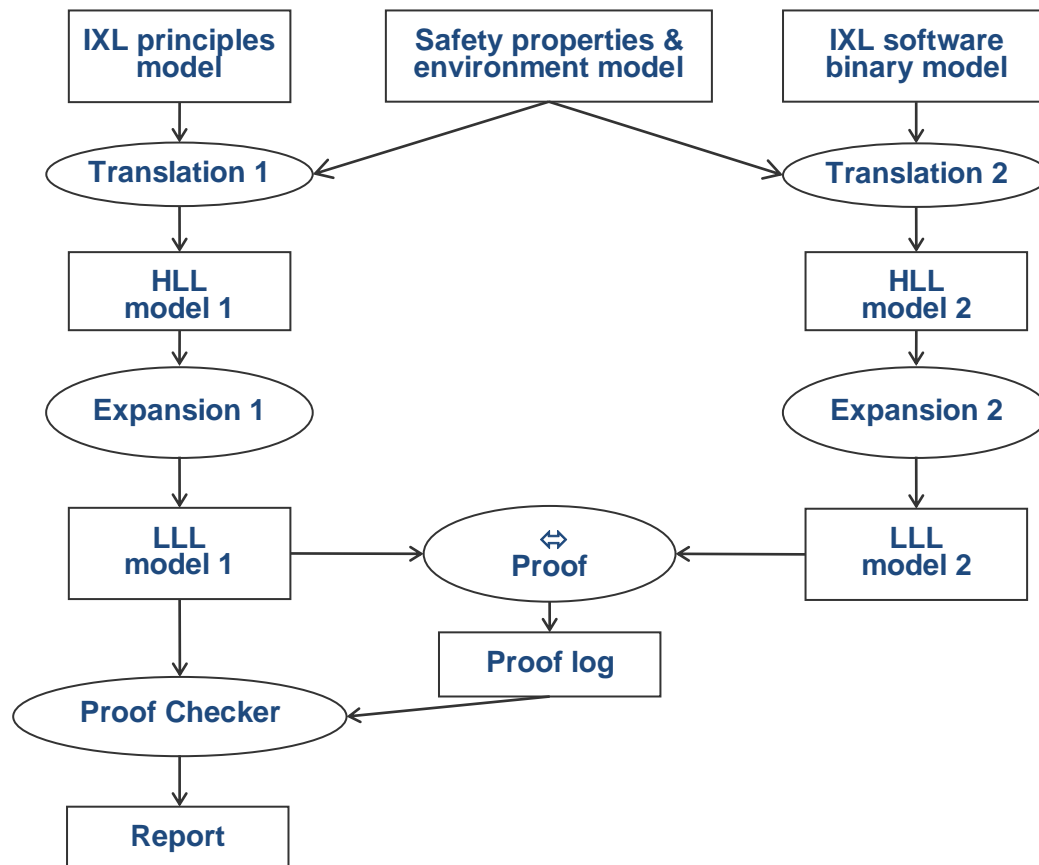


Fig. 4: Proofer certified proof process

The proof tool verifies then that the two low-level representations are equivalent. It delivers a proof log that encodes the proved models and the proof tree that traces the proof steps applied during the verification.

Finally, a proof checker tool verifies the proof log in order to check that it encodes the correct models and that the proof tree traces a correct proof.

The fifth step produces a verification report attesting the correctness of the proofs.

3.2.2.2 Proofer Tools

The tools used for the Proofer process were developed by Prover technology.

The translator used in the fourth step is not certified.

The same proof tool is used for steps four and five. It is not a certified tool.

The translators used in the fifth step are distinct and were developed by distinct teams. One is written in C language, the other in OCCAM language.

Just as the translators, the expansion tools used for the fifth step are distinct and were developed by distinct teams. One is written in C language the other in OCCAM language.

The proof checker tool used for the fifth step is the only certified tool used in the Proofer process.

The diversified development of the translators and expansion tools and the certification of the proof checker tool means that the fifth step is as a certified activity as determined by RATP.

3.2.2.3 Analysis of Proofer methodology

The Proofer methodology has several indisputable advantages.

- It allows checking exhaustively and with a level of assurance difficult to achieve with other methodologies that the instance of an IXL system for a particular switching station complies with safety requirements.
- It is intrusive neither in the development nor in the execution of the IXL system since it does not introduce any new activity for the development of the system and does not add any piece of software to the existing one.
- It is highly automated and can be applied by appropriately trained railway engineers unaware of theory and techniques of proof.

But it has also several drawbacks.

- It treats instances of the IXL system for a particular switching station. Consequently, even though it proved to be effective for moderately complex switching stations, it is almost certain that it will not be able to treat instances of the IXL system for relatively complex or complex switching stations whose state space is considerably larger and will certainly exceed the capacities of the tools supporting the technology. In particular, it has been noticed that the size of the state space grows exponentially with the number of routes that pass by the same point. In brief, the Proofer methodology cannot treat all switching stations and this is recognised by the organisations involved in this approach.
- It is intimately related to the internal design of the IXL principles and to the execution model of the definition language of the IXL principles. Therefore, for each definition language of IXL principles it is necessary to recreate the models of the safety properties, of the IXL environment and of its execution model and to develop specific translators.
- Step 4 (assisted iterative analysis) of the Proofer process uses Bounded Model Checking. One therefore needs to find a good bound (the depth of the execution tree), which is difficult to tune up. If the bound is too large, the verification takes a very long time. If it is too low, spurious liveness counter-examples are produced by the tool.
- The approach based on instantiated IXL systems mixes together three concerns: the IXL principles, IXL safety properties and the track topology. Therefore, in step 4 of the Proofer process, the same properties are replicated several times. For instance, a property that concerns a point is replicated as many times as we have points in the switching station. For simple configurations, intermediate lemmas can be generated automatically for each copy to achieve the proof. But for complex systems, it is more difficult to automate lemma generation, and proving several times the same property is time consuming and might not be economically feasible in the long run.
- It consumes many person and machine resources.
- It requires the development of several complex tools.

The first two drawbacks are considered sufficiently important to prevent the adoption of the Proofer methodology without investigating an alternative solution that overcomes them. The next section presents the principles of the solution that we shall investigate in the case study.

4. Description of the studied solution

In this section we present a solution allowing to prove formally that interlocking systems comply with safety requirements that is deeply inspired by the architecture, the design principles and the development and verification methods of the ATC and ATP systems presented in section §2.

4.1 Architecture, design, development and verification principles

In section §2 we saw that an ATC system involves a safety-critical ATP subsystem ensuring that trains are operated safely. Following this architecture we propose to develop a new interlocking system component ensuring that the controls issued by the IXL do not result in hazardous situations. We shall call this system the Interlocking Dynamic Controller (IXL-DC) in the sequel.

In this architecture, the IXL-DC system gets the controls issued by the IXL and verifies that they are safe. For instance, when the IXL controls the aspect of a signal authorizing trains to enter an area of the track, the IXL-DC must ensure that this area is compatible with all other authorized areas and that all points in this area are in a safe position and locked.

The ATP subsystem is designed as a generic component parameterized by a set of predefined constants representing the characteristics of trains (length, maximum and minimum speeds and accelerations, etc.), and the track layout (stations, stabling areas, slopes, turnaround areas, etc.). This design pattern simplifies the verification of the instance of the ATP for a particular line. The generic part is verified once for all and only the values of the actual parameters of the ATP have to be verified regarding the properties they must meet. We adopt the same design pattern for the IXL-DC. The generic part of IXL-DC will implement its dynamic behaviour and will be parameterized by a predefined set of constants representing the characteristics of the controlled and monitored trackside devices (points, signals, key locks, etc.) and the track layout.

ATIS uses Classical-B to develop the safety-critical software of its ATP systems. The existing formal process of ATIS addresses correctness of software implementation of the individual ATP units, but demonstration of safety at system-level is done outside the formal Classical-B process. This process involves three main steps. First, the expected behaviour of the software is formally specified in Classical-B and this specification is proved consistent. Then the formal specification is refined into concrete Classical-B programs and refinement is proved correct. Finally, Classical-B programs are automatically translated into Ada programs by a certified translator. Thus there is formal evidence that programs comply with their specification, that is to say that safety-critical software of ATP system meets their expected behaviour. Inspired by that approach, we propose to develop IXL-DC formally with Event-B and Classical-B. In ADVANCE, we are using Event-B to demonstrate formally at **system level** that IXL-DC ensures the system level safety requirements and we intend to refine this Event-B model of IXL-DC into a Classical-B model of IXL-DC. Then starting from the latter it will be possible to develop the software of IXL-DC formally in Classical-B as this is usually done.

From the verification point of view, safety analysis and verifications ensure that the Classical-B model of ATP is relevant and complete, that the proof is correct and that the overall process meets certification requirements. In a similar way, we propose that safety analysis and verifications ensure that the Event-B model is relevant and complete and that the proposed process meets certification requirements.

4.2 Development and verification processes

According to the previous section the development process of the IXL-DC that we intend to implement is very similar to that of the ATP, except that it includes a formal activity at system

specification level. Roughly speaking the complete process involves six steps. Figure 6 below illustrates this process.

The purpose of the first step is to create and prove an Event-B model of the part of the signalling system dealing with reservation, protection and release of the routes for trains.

The purpose of the second step is to refine and decompose progressively this initial system model into several Event-B models each of which dealing with a particular part of system.

In the third step the Event-B models of the IXL-DC and of its parameters are translated into Classical B models.

The fourth step covers the development of the IXL-DC software⁷. During this step the Classical-B model of the IXL-DC is refined stepwise into Classical-B programs.

The fifth step automatically translates the Classical-B programs into executable Ada programs using an existing certified translator.

The sixth step is the generation and verification of the actual parameters of the IXL-DC for a particular switching station. The purpose of this verification is to ensure that values of actual parameters IXL-DC comply with the formal parameters model. Our intention is to automate this with enhancements to the ProB tool being developed in ADVANCE.

Only the three first steps of this process will be achieved in the ADVANCE project. We shall neither develop IXL-DC software formally nor generate and verify actual parameters for a particular switching station. These activities are out of the scope ADVANCE methods and tools and are already well known and practiced in ATIS and Systemel.

4.3 Rationale

We believe that the solution presented above is technically feasible; worthwhile from the methodological and economical points of view and that the previously proposed process meets certification requirements.

Technical feasibility of the IXL-DC concept consists in the ability to create and prove an abstract Event-B model involving the IXL-DC and fully representative of the safety requirements and of the static and dynamic features of the environment where IXL-DC operates.

We believe that this is possible, but the case study has to confirm our belief, because IXL-DC will be designed explicitly to verify safety requirements. Therefore, unlike IXL, we shall have a strong correspondence between the concepts of the requirements and the concepts of the events and we shall not even consider some concepts of IXL related to control trackside equipment. In short, this will allow us to create a simpler model than the one of IXL.

Other technical aspects do not present particular difficulties. The proposed architecture has been implemented for the ATC system. Scalability is not an issue because proof deals with proof rules and predicates abstracting properties of data rather than with actual values of data, therefore proof of IXL-DC is independent from the size and complexity of switching stations. And finally, ATIS has long experience in developing software in Classical-B.

We believe the solution worthwhile from the methodological point of view because, compared with the Proofer methodology, it is independent from the implementation technology of IXL, its process is

⁷ Actually this step is decomposed into the steps of the software development process.

simpler and does not require other tools than those already available either in Rodin (tools dealing with Event-B models) or in ATIS development workbench (tools dealing with Classical-B models).

From the economical point of view, the cost of development and verification of the IXL-DC and of the new architecture would be compensated by savings: savings in development of specific formal verification tools and savings in the safety verification of instances of IXL. Indeed, since IXL-DC would be verified once for all, safety verification of an instance of the IXL for a particular switching station would be reduced to verify compliance of values of actual parameters of IXL-DC with properties of formal parameters. We intend to do this automatically using the enhancements to the ProB tool being developed in ADVANCE.

We believe, but again our belief has to be evaluated by WP1, that the process presented above meets certification requirements of CENELEC standards for railway safety. Our main argument is that our process provides a formal basis to system activities that until now were informal and leaves the other activities unchanged. Indeed, in current ATIS safety verification process and in Proofer process, refinement of system level safety requirements into IXL level safety requirements is informal. In our process, system level safety requirements are proved consistent, they are formally refined into IXL-DC level requirements and the refinement is proved correct. All other activities of our process meet certification requirements because we keep the methods and practices recommended by CENELEC standards and widely accepted by railway industry and transport authorities: safety analysis based on Fault Trees (FT) and Failure Modes Effects Analysis (FMEA), formal development of safety-critical software in Classical-B, data generation and/or verification based on formal models.

5. Case Study Activities, Process and Methods

We describe in this section the activities that will be achieved during the case study.

5.1 Demonstration of technical feasibility

The purpose of this activity is to demonstrate the technical feasibility of the IXL-DC concept and of the use of ADVANCE methods and tools for verification of system-level safety of the IXL-DC concept. According to what was said above, our goal is to create an abstract Event-B model involving the IXL-DC and truly representative of safety requirements and of the environment where IXL-DC operates.

The inputs of this step are the documents of a driverless CBTC system operational since 2008. We shall extract from those documents the functional and safety requirements on interlocking systems.

The Event-B model will be made of two parts. One part will deal with dynamic aspects of the system and the other part will deal with static aspects. The former part can be considered as the specification of a parameterised component and the later as the specification of its formal parameters.

The Event-B model will be validated by means of animation using the ProB tool. Our goal is to validate the actual dynamic behaviour of the IXL-DC with respect to expected behaviour.

During the feasibility study we shall start proving the general purpose theories of the model and we shall assess informally the “provability” of the model. We shall achieve the proof of the model once the validation by animation and the safety assessment activity described in the following section will be completed.

We shall transform manually the Event-B model into a Classical-B model. Then, according to the latter, we shall develop informally a proof-of-concept prototype that we intend to test with actual IXL factory tests.

5.2 Assessment and Certification

The objective of assessment activity is to verify that the Event-B model developed in the previous activity meets all functional and safety requirements. In other words the Event-B model fits for industrial application.

Inputs of assessment activity are the system documents and the Event-B model. The result of this activity is an assessment report.

Functional assessment will be carried out on the basis of traceability between system specification documents, Event-B models and animation scenarios.

Safety assessment will be carried out on the basis of analyses done with the classical methods employed in the railway industry and recommended by CENELEC standards such as event trees and fault trees. However, for experimental purposes, in conjunction with WP5, we will apply the System Theoretic Process Analysis (STPA) method proposed recently ([7]).

The objective of the certification activity is to define a strategy for integration of the ADVANCE methods and tools into a certified process.

The inputs of certification activity are documentation of the ATIS certified process, CENELEC standards and the process methods and tools developed in ADVANCE. The results of this activity are the document describing the process and the recommendations to follow to make it certifiable according to CENELEC standard.

5.3 Full application of ADVANCE methods and tools

The purpose of this activity is to use the methods and tools developed in the ADVANCE project to implement the activities presented above and to bring them at an industrial level. We intend to achieve the following tasks.

- Develop an initial model with Rodin platform (started in Demonstration of technical feasibility);
- Animate the initial model using ProB (started in Demonstration of technical feasibility);
- Prove the initial model using Rodin's prover and the THEORY plug-in (started in Demonstration of technical feasibility);
- Refine stepwise the initial model in order to introduce information progressively;
- Prove refinements using Rodin's prover and the THEORY plug-in;
- Validate the final model according to functional and safety requirements using testing based on model animation scenarios executable by ProB;
- Decompose the final model into its architectural parts using the Composition / Decomposition plug-in;
- Document the models and ensure traceability using ProR.
- Validate the hypotheses used for specifying and modelling the system by running side-by-side a continuous mathematical model of train together with a discrete model of the interlocking using co-simulation. ATIS already has Matlab models of train movements. The aim will be to use the multi-simulation framework being developed in ADVANCE to enable the Matlab simulation to provide discrete train movement stimulus to drive the discrete model of the IXL.

The inputs of this activity are system documents. The results of this activity are the documented models, the proof theories and the test scenarios.

6. Evaluation Criteria

We shall reach our crucial objective if we succeed in creating and proving a representative Event-B model of the IXL-DC compliant with safety requirements, in refining and decomposing the Event-B models to separate architectural units and in translating the Event-B model of the IXL-DC unit into a Classical-B model suitable for formal development to software. Indeed, this ensures that the solution based on the concept IXL-DC to formally prove that an instance of IXL system for a switching station meets safety requirements is effective and independent of the complexity of the switching station and from the implementation technology of the IXL.

A second criterion is the technological level reached by the methods and tools developed in the ADVANCE project through the completion of the case study. This criterion can be measured by the list of methods and tools provided by ADVANCE that were actually used and useful for the achievement of the tasks listed above.

A third criterion is the contribution of the case study to improve the formalism, methods and tools developed in WP3, WP4, and WP5 of the ADVANCE project. This criterion can be measured by the list of improvements proposed in during the case study and the list of propositions adopted.

7. References

- [1] Industrialising a proof-based verification approach of computerised interlocking systems. S. Behnia, A. Mammar, J.-M. Mota, N. Breton, P. Caspi et P. Raymond. WIT Transactions on The Built Environment, Vol 103, © WIT Press 2008.
- [2] Modeling in Event-B – System and Software Engineering. J.-R. Abrial. © Cambridge University Press 2010.
- [3] The B Book – Assigning programs to meanings. J.-R. Abrial. © Cambridge University Press 1996.
- [4] The SCADE Suite. <http://www.esterel-technologies.com/products/scade-suite/>
- [5] Solid State Interlocking (1st. Edition ed.). D H Stratton. © IRSE 1988.
- [6] PROOFER, démonstration de la sécurité d’une application ferroviaire de signalisation en mode nominal et modes dégradés par la preuve formelle. Pierre Chartier – RATP, Séminaire de Valorisation GO2, « Gouvernance et Transports en commun », Paris 3 et 4 décembre 2009.
- [7] Engineering a Safer World: Applying Systems Thinking to Safety, Nancy Leveson. MIT Press, January 2012.