Modelling and Analysing ERTMS L3 Moving Block Railway Signalling with Simulink and Uppaal SMC

Davide Basile[0000−0002−7196−6609] [1,2],
Maurice H. ter Beek[0000−0002−2930−6367] [1] (✉),
Alessio Ferrari[0000−0002−0636−5663] [1], and Axel Legay[3]

[1] ISTI–CNR, Pisa, Italy
[2] University of Florence, Florence, Italy
[3] Université Catholique de Louvain, Louvain-la-Neuve, Belgium

Abstract. Efficient and safe railway signalling systems, together with energy-saving infrastructures, are among the main pillars to guarantee sustainable transportation. ERTMS L3 moving block is one of the next generation railway signalling systems currently under trial deployment, with the promise of increased capacity on railway tracks, reduced costs and improved reliability. We report an experience in modelling a satellite-based ERTMS L3 moving block signalling system from the railway industry with Simulink and Uppaal and analysing the Uppaal model with Uppaal SMC. The lessons learned range from demonstrating the feasibility of applying Uppaal SMC in a moving block railway context, to the offered possibility of fine tuning communication parameters in satellite-based ERTMS L3 moving block railway signalling system models that are fundamental for the reliability of their operational behaviour.

1 Introduction

The railway sector is well known for its robust safety requirements, as witnessed by the CENELEC EN 50128 standard [22] for the development of software for railway control and protection systems, which highly recommends the use of formal methods for software systems to be certified at Safety Integrity Levels SIL 3 and SIL 4. In fact, formal methods and tools are widely applied to railway systems [7, 9, 13, 23–25, 28, 30]. Consequently, the railway sector is notoriously cautious about the adoption of technological innovations compared with other transport sectors. Hence, while satellite-based positioning systems are in use for some time now in the avionics and automotive sectors, current railway signalling systems still prevalently use traditional ground-based train detection systems and fixed block distancing. However, the faster trains are allowed to run, the longer their braking distance and the longer the safety distance must be, thus decreasing line capacity. A challenge in the railway sector therefore concerns the development of moving block signalling systems that are as effective and precise as possible [32]. This includes satellite-based positioning, leveraging on an integrated solution for signal outages (think, e.g., of tunnels) and so-called multi-paths, which typically affect satellite positioning in urban environments [12, 46].

The work presented in this paper is one of the outputs of a larger endeavour of the first three authors in the context of the H2020 project ASTRail[4] (SAtellite-based Signalling and Automation SysTems on Railways along with Formal Method and Moving Block Validation) funded by the EU's Shift2Rail[5] initiative. Shift2Rail stimulates the development of safe and reliable technological advances that allow to complete the single european railway area with an ambitious aim: "double the capacity of the European rail system and increase its reliability and service quality by 50%, all while halving life-cycle costs." To this aim, it supports the transition to next generation ERTMS railway signalling systems, including satellite-based train positioning, moving block distancing, and automatic driving [8]. ASTRail makes use of a satellite-based ERTMS Level 3 moving block railway signalling scenario for two different purposes:

- First, in a reduced format, for a trial application of formal modelling and analysis to assess the usability and applicability of formal methods and tools in the railway domain. This assessment is an important issue for the successful uptake of formal methods and tools in the railway industry [7]. In [5], we presented our trial experience in modelling and (statistical) model checking a satellite-based moving block signalling scenario with Uppaal SMC.
- Second, for modelling and validating a more detailed model as a major portion of an integrated system design of moving block signalling with automated driving technologies to provide a rigorous and verified definition of functional, interoperability, and dependability requirements. As part of the assessment, we conducted a survey with railway practitioners to identify the most mature (semi-)formal methods and tools to be used in the railway context [28]. As a result of this survey, a total of 14 tools were carefully reviewed by means of a systematic evaluation based on a set of 34 evaluation features, upon which eight tools were selected for the above mentioned trial application phase, in which we modelled principles of the moving block scenario in all eight tools. Simulink and Uppaal were among the eight selected tools. Specifically, Simulink was considered particularly appropriate for functional requirements elicitation and animation involving domain experts, while Uppaal was considered the appropriate choice for verification of quantitative aspects. More information is available in our contribution [28].

In this paper, we present models of the aforementioned detailed satellite-based ERTMS L3 moving block signalling system model in both Simulink and Uppaal. The Simulink model was obtained from a requirements elicitation and refinement activity performed with the industrial partners of ASTRail, carried out to consolidate an initial set of requirements for the moving block signalling system into an executable specification, after which we developed a corresponding Uppaal model. We report on and draw some lessons from this modelling experience and subsequent analyses with Uppaal SMC. We choose to perform statistical model checking with Uppaal SMC rather than simulation and analysis with Simulink, because we have all the monitoring infrastructure for temporal properties. However, the level of abstraction is the same in both models.

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

[5] http://www.shift2rail.org

We show how Uppaal SMC can assist in fine tuning communication parameters that are fundamental for the reliability of the model's operational behaviour. In particular, we validate that (i) the frequencies of the messages exchanged between the train and its trackside control system as well as (ii) the unit of distance that a train is allowed to proceed based on a movement authority can be set such that the probabilities of failures (like the train exceeding its movement authority, i.e., failing to brake if it lacks permission to proceed) are close to zero. While numerical constraints for (i) and (ii) were previously defined by railway experts, in ASTRail we wanted to explore to which extent Uppaal SMC can be exploited to validate such constraints and to support sensitivity analysis on the parameters.

Related Work We know of several other attempts at modelling and analysing ERTMS L3 signalling systems. Most notably, ERTMS Hybrid L3 systems (using virtual fixed blocks) and its RBC component have recently been modelled and analysed in [2, 4, 15, 41, 44] with Promela/Spin, mCRL2, Electrum, and Event-B. However, none of these permit quantitative modelling and analysis, which are fundamental to demonstrating the reliability of the operational behaviour of satellite-based ERTMS L3 moving block railway signalling system models.

We are also aware of attempts to model stochastic or hybrid models of ERTMS L3 (moving block) scenarios in [31,34,35,38] with Simulink, the bounded model checker HySAT, the probabilistic hybrid automata verifier ProHVer, UML, the symbolic model checker SMV, timed Petri nets and the timed Petri net analyser Tina, generally applying classical (i.e., not statistical) model checking.

We recognise added value in so-called *formal methods diversity*, as advocated in [42,43], according to which, inspired by code or design diversity [40], applying diverse formal methods and tools on replications or different variants of a design may increase confidence in the correctness of the analysis results. Therefore, we believe that this paper contributes to an increased confidence in the reliability of satellite-based ERTMS L3 moving block railway signalling systems. At the same time, we show how multiple formal/semi-formal tools can also play a complementary role to address different needs of the railway development process, namely functional requirements elicitation and verification of quantitative properties.

Outline The rest of the paper is organised as follows. Section 2 introduces the industrial case study: next generation satellite-based ERTMS moving block railway signalling systems. Section 3 describes a Simulink model of the case study, developed in agreement with our industrial partners, followed by a corresponding Uppaal model in Section 4. Section 5 presents an analysis of the case study with Uppaal SMC and Section 6 reports some lessons learned from this modelling and analysis experience. Section 7 concludes the paper and discusses future work.

2 ERTMS L3 Moving Block Railway Signalling

The ASTRail project aims to introduce recent scientific achievements as well as cutting-edge technologies from other transport sectors, in particular avionics and automotive, in the railway sector. The project leverages formal methods and tools for careful analyses of the resulting novel applications and solutions in terms of safety and performance. One of the main focusses of ASTRail concerns the use of the Global Navigation Satellite System (GNSS) [46] for onboard train localisation. While satellite-based positioning systems have been in use for quite some time now in the avionics and automotive sectors, to provide accurate positioning and distancing, the current railway signalling systems are largely based on fixed blocks, implemented by specific trackside equipment along the railway lines. A block is a section of the track between two fixed points, which start and end at signals, with their lengths designed to allow trains to operate as frequently as necessary (i.e., ranging from many kilometres for secondary tracks to a few hundred metres for busy commuter lines). The block sizes are determined based on parameters like the line's speed limit, the train's speed, the train's braking characteristics, drivers' sighting and reaction times, etc. But the faster trains are allowed to run, the longer the braking distance and the longer the blocks need to be, thus decreasing the line's capacity. This is because the railway sector's stringent safety requirements impose the length of fixed blocks to be based on the worst-case braking distance, regardless of the actual speed of the train.

The next generation railway signalling systems no longer rely on trackside equipment for train position detection and train integrity supervision, but an onboard odometry system is responsible for monitoring the train's position and autonomously computing its current speed [32]. By exploiting knowledge of the position of the rear end of the train ahead, a safe zone around the moving train can be computed, thus considerably reducing headways between subsequent trains. The resulting moving block signalling systems allow trains in succession to close up, in principle to the braking distance (cf. Fig. 1).

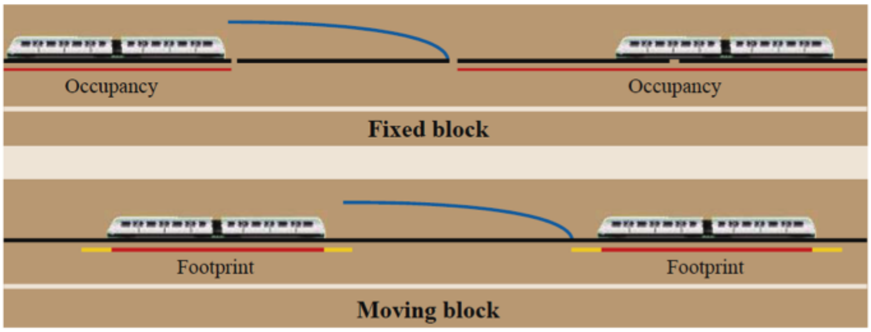


Fig. 1. Safe braking distance between trains for fixed block and moving block signalling (Image courtesy of Israel.abad/Wikimedia Commons distributed under the CC BY-SA 3.0 license)

Moving block signalling allows for more trains to run on existing railway tracks, in response to the ever-increasing need to boost the volume of passenger and freight rail transport and the cost and impracticability of constructing new tracks. For this to work, the precise absolute location, speed, and direction of each train needs to be known. These can be determined by a combination of sensors: active and passive markers along the track, as well as trainborne speedometers. This envisioned future switch to next generation signalling systems would not only optimise the exploitation of railway lines due to the adoption of moving block signalling, but the removal of trackside equipment would result in lower capital and maintenance costs [32]. In ASTRail, the first three authors are involved in the formal modelling and analysis of moving block railway signalling systems by means of different formal methods and tools, and this paper reports on one such experience (cf., e.g., [5, 28]).

ERTMS The European Rail Traffic Management System (ERTMS) [19] is an international standard aiming to enhance safety and efficiency and improve cross-border interoperability of trains in Europe by the replacement of national railway signalling systems with a European standard for train control and command systems. ERTMS relies on the European Train Control System (ETCS), an Automatic Train Protection (ATP) system continuously supervising the train to ensure that safety speed and distances are not exceeded. The ERTMS/ETCS standard distinguishes four levels of operation, depending on the role of trackside equipment and on the way the information is transmitted to/from trains. It is currently deployed on several lines throughout Europe at most in its Level 2.

ERTMS Level 2 ERTMS L2 uses trackside equipment (track circuits) to detect the occupancy of a section of a railway track by trains, determining the location of trains with a coarse granularity. This information is sent to a trackside unit, termed Radio Block Centre (RBC), which sends a Movement Authority (MA) to each train. The MA is computed by summing the free track circuits ahead, meaning L2 is based on *fixed block signalling*. The MA provides a train with the maximum distance it is allowed to travel, the maximum speed (depending on the track) it is allowed to travel at, and data about the track ahead (like temporary speed restrictions and (un)conditional emergency stops). The so-called Onboard Unit (OBU) of each train uses the MA and data stored on board (e.g., the train's braking capability) to compute the braking curve or the dynamic speed profile that determine the speed limit, triggering an emergency brake whenever this limit is exceeded. In L2, so-called Eurobalise responders on the rails of a railway are used for exact train positioning, while the required signalling information is provided to the driver's display by continuous data transmission via GSM-R with the RBC. Further trackside equipment is needed for train integrity detection.

ERTMS Level 3 ERTMS L3 no longer uses trackside equipment for train positioning and train integrity supervision. Instead, the OBU is responsible for monitoring the train's position and computing its current speed through its odometry system. To this aim, the OBU periodically sends the train's position to the RBC and the RBC, in turn, sends back an MA to each train. The MA is computed by exploiting knowledge of the position of the rear-end of the foregoing train, meaning L3 is based on *moving block signalling*. As a result, headways between trains can be considerably reduced, in principle to the braking distance. Actually, L3 as defined in [20] does not explicitly refer to the moving block concept, but it admits any implementation able to periodically provide the RBC with the train positions and using limited trackside equipment. A few pilot implementations, referred to as Hybrid L3 [2,4,15,21,41], use virtual fixed blocks: a line is logically divided into fixed length blocks and the OBU is in charge of communicating, at specific points of the line (virtual balises), the train's position, computed using its onboard odometry system. Moving block signalling based on continuous communication and MA computation is currently implemented in some automatic metros, as part of CBTC (Communication Based Train Control) systems.

Moving Block Scenario The components of the moving block scenario considered in this paper are depicted in Fig. 2. The train carries the Location Unit (LU) and OBU components, while the RBC is a trackside component. The LU receives the train's location from GNSS satellites, sends this location (and the train's integrity) to the OBU, which, in turn, sends the location to the RBC. Upon receiving a train's location, the RBC sends an MA to the OBU (together with speed restrictions and route configurations), indicating the space the train can safely travel based on the safety distance with preceding trains. The RBC computes the MA by communicating with neighbouring RBCs and by exploiting its knowledge of the positions of switches and other trains (head and tail position) by communicating with a Route Management System (RMS). In our scenario, we abstract from an RMS and communication among neighbouring RBCs: we consider one train to communicate with one RBC, based on a seamless handover when the train moves from one RBC supervision area to an adjacent one, as regulated by its Functional Interface Specification [48]. Next to these physical components, there are two temporal constraints for the OBU to respect: the location is continuously updated every 5 seconds, whereas the MA must be continuously updated within 10 seconds. If the OBU does not receive an MA within 10 seconds from the last MA, the OBU is required to force the train to brake.

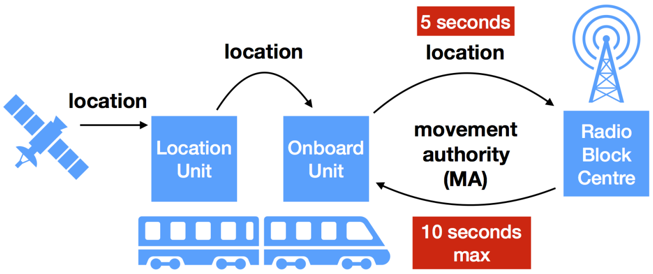


Fig. 2. Overview of ERTMS moving block railway signalling

3 Simulink Model of ERTMS L3 Moving Block

Simulink is a model-based development tool supporting graphical design, simulation, test generation, and code synthesis of dynamic systems.[6] A Simulink model's basic unit is a block, an element that acquires some input and produces some output. Simulink also includes Stateflow, a graphical language inspired by Harel's hierarchical statecharts [37]. Simulink blocks can contain Stateflow statecharts (called charts in Simulink terminology), to represent event-based systems.

[6] http://www.mathworks.com/products/simulink.html

In this section, we present the Simulink model of the moving block system resulting from a requirements elicitation and refinement activity performed with the industrial partners of ASTRail. It is the output of multiple iterations involving the third author and the industrial partners, carried out to consolidate an initial set of requirements for the moving block system into an executable specification. Simulink was selected as preferred tool to support this elicitation activity for two reasons. First, given its previous usage in the railway industry for similar purposes [26,27]. Second, because of the outcome of the assessment reported in [28]. As mentioned in the Introduction, we conducted a survey with railway practitioners to identify the most mature (semi-)formal methods and tools to be used in the railway context, and Simulink was one of the eight selected tools. The model, together with its documentation in HTML format, is publicly available.[7] Here, we show the model's architecture and some excerpts of its behaviour.

Model Architecture Figure 3 reports the architecture of the model, which includes three main Simulink blocks representing the interacting subsystems, namely OBU, LU, and RBC.[8] Each block communicates with the other blocks by means of input/output messages. For example, the label named `location` is one of the outputs of the LU, and it is input to the OBU block. This indicates a virtual channel by which a message is exchanged between LU and OBU, including the current train location. Similarly, `location_to_RBC` is one of the outputs of the OBU block, also serving as input to the RBC block: the OBU location, received from the LU, is passed to the RBC, which, in turn, can compute the `MA` and send it to the OBU. The OBU is also in charge of activating the brake, and the brake's status can be visualised in the `BRAKE_COMMAND` scope element. Similarly, other scope elements are used to visualise a `TIMER`, indicating the time from the last received MA (2.4 seconds in Fig. 3), and `SPACE_TO_EOA`, which is the space from the current position to the end of the MA (996.4 meters). Following the requirements, failure inputs (`OBU_FAIL`, `RBC_FAIL`, and `LU_FAIL`) are associated to each block to simulate external events that may trigger system failures.

Behaviour The behaviour of each block is represented by means of a Stateflow chart. Figure 4 reports an excerpt of the chart representing the OBU behaviour. The excerpt depicts a parallel state (dashed lines indicate parallel states) named `SEND_LOCATION_TO_RBC`, which includes two mutually exclusive substates: one normal state (`SEND_LOC_TO_RBC`) and one failure state (`POSITION_ERROR`). When the system is in the normal state, it continuously checks whether a new location is received. This is performed through the function `check_new_location()`, which is graphically represented as a flowchart inside the state. Whenever a new location is received from the LU (`OBU_REC_location_flg == 1`), it is stored together with the current time stamp. Every five seconds, the location is sent to the RBC, if the location is not older than one second. This is enforced through the condition `after(5, sec) && check_location_fresh [...]`.

The other parallel state named `RECEIVE_MA` takes care of MA reception. Specifically, when an MA is received (`OBU_REC_MA_flg == 1`), it is stored in the

[7] https://github.com/alessioferrari/ASTRail-simulink-models

[8] The full model includes the train's dynamics, not reported here to ease visualisation.

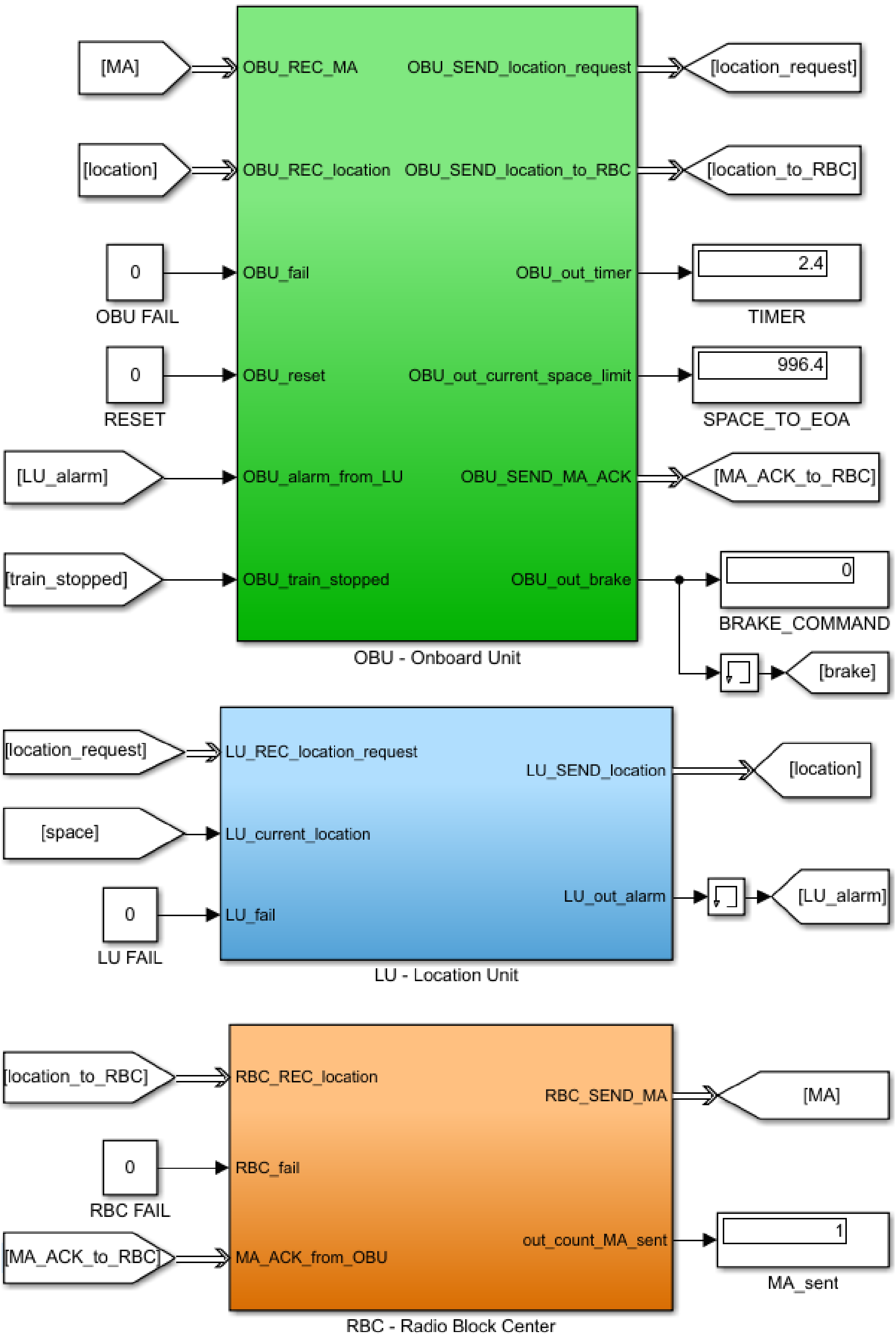


Fig. 3. Architecture of the Simulink moving block model

variable `MA_value`, and the OBU also stores the current location in the variable `MA_reference`. This will be used as a reference to update the variable that indicates how much space is left to the end of the MA (`SPACE_TO_EOA` in Fig. 3), while the train progresses its mission. Then, an ack message is sent to the RBC. The code inside the state `NEW_MA_RECEIVED` continuously updates the value of the variable `OBU_out_timer`, which represents the time that has passed since the last MA was received, and is visualised in the scope `TIMER` of Fig. 3.

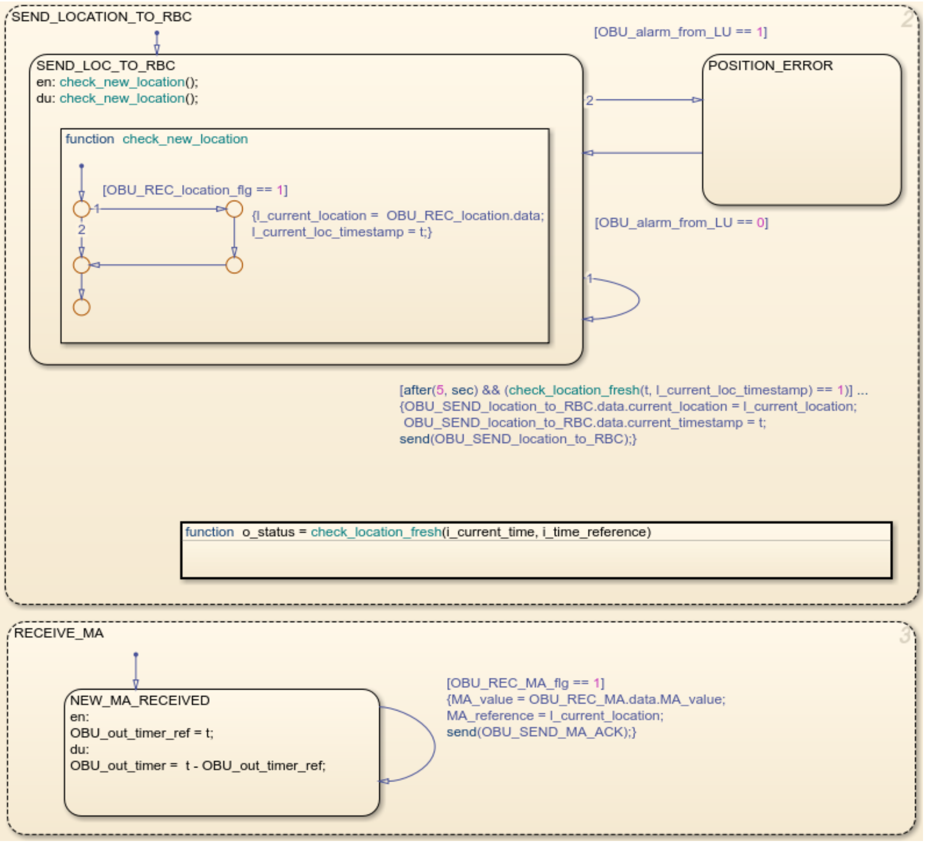

SEND_LOCATION_TO_RBC
2
[OBU_alarm_from_LU == 1]
SEND_LOC_TO_RBC
en: check_new_location();
du: check_new_location();
POSITION_ERROR
2
function check_new_location
[OBU_REC_location_flg == 1]
1
{l_current_location = OBU_REC_location.data;
l_current_loc_timestamp = t;}
2
[OBU_alarm_from_LU == 0]
1
[after(5, sec) && (check_location_fresh(t, l_current_loc_timestamp) == 1)] ...
{OBU_SEND_location_to_RBC.data.current_location = l_current_location;
OBU_SEND_location_to_RBC.data.current_timestamp = t;
send(OBU_SEND_location_to_RBC);}
function o_status = check_location_fresh(i_current_time, i_time_reference)
RECEIVE_MA
3
NEW_MA_RECEIVED
en:
OBU_out_timer_ref = t;
du:
OBU_out_timer = t - OBU_out_timer_ref;
[OBU_REC_MA_flg == 1]
{MA_value = OBU_REC_MA.data.MA_value;
MA_reference = l_current_location;
send(OBU_SEND_MA_ACK);}

Fig. 4. Excerpt of the behaviour of the OBU model

4 Uppaal Model of ERTMS L3 Moving Block

Uppaal SMC [16] is a variant of Uppaal [11], which is a well-known toolbox for the verification of real-time systems.[9] Uppaal models are stochastic timed automata, in which non-determinism is replaced with probabilistic choices and time delays with probability distributions (uniform for bounded time and exponential for unbounded time). These automata may communicate via broadcast channels and shared variables.

[9] http://people.cs.aau.dk/~adavid/smc

Statistical Model Checking (SMC) [1, 39] is concerned with running a sufficient number of (probabilistic) simulations of a system model to obtain statistical evidence (with a predefined level of statistical confidence) of the quantitative properties to be checked. SMC offers advantages over exhaustive (probabilistic) model checking. Most importantly, it scales better, since there is no need to generate and possibly explore the full state space of the model under scrutiny, thus avoiding the combinatorial state-space explosion problem typical of model checking. Moreover, the required simulations can trivially be distributed and run in parallel. This comes at a price. Contrary to (probabilistic) model checking, exact results (with 100% confidence) are impossible to obtain. A further advantage is related to its possible uptake in industry. Compared to model checking, SMC is simple to implement, understand and use, and it requires no specific modelling effort other than an operational system model that can be simulated and checked against (state-based) quantitative properties. In fact, SMC is becoming more and more widely accepted in industry [3, 10, 14, 29, 36, 45].

In this section, we discuss the Uppaal formalisation of the moving block system, derived from the semi-formal Simulink model presented in the previous section. The model is publicly available.[10] Here, we outline the automata constituting the model and describe the one modelling the OBU in more detail.

From Simulink/Stateflow to Uppaal This transformation is simplified by the fact that both formalisms use state machines. While we are aware of other efforts to map Simulink/Stateflow diagrams into Uppaal SMC (cf, e.g., [29]), we encountered some peculiarities to be taken care of to transform the moving block model of the previous section. In particular, Uppaal does not cater for the primitive description of machines with hierarchical states. Moreover, Simulink does not primitively provide concurrency between the processes, i.e., the scheduling is fixed a priori. This is not the case in Uppaal, where there is an interleaving between all possible actions. Actually, the scheduling order was not part of the original ASTRail specification, so this forced scheduling was relieved in Uppaal.

Communication between Simulink blocks is implemented through messages and input/output variables, and through shared variables inside Stateflow charts, whereas in the Uppaal model we use communication via broadcast channels. Simulink/Stateflow diagrams and Uppaal models use different time modelling. In the Simulink model, variables were used that memorise the time difference between events, while the Uppaal model uses clocks that allow to memorise the time elapsed between the various events. Furthermore, the Uppaal model was enriched with probabilities and stochastic events, which were taken from additional specifications of the moving block system by our industrial project partners. We only used rates of exponential delays, since exponential distributions are the only available distributions in Uppaal for unbounded delays due to their memoryless property. Hence, the Uppaal model represents a refinement of the initial semi-formal specification in Simulink. In Section 5, we will see that this allows subsequent verification of properties of interest with Uppaal SMC.

[10] https://github.com/davidebasile/ASTRail

The Uppaal Model The model consists of a number of automata composed as a synchronous product. Below, we list the various components that together form the model, followed by a more detailed description of the main automaton modelling the OBU component. As for the Simulink specification, the model consists of three main entities, namely the RBC, the LU, and the OBU, each represented by a different automaton. Each entity moreover accounts for a probabilistic failure that is modelled through three additional automata, called `RBC_Failure_T`, `LU_Failure_T`, and `OBU_Failure_T`, which model the failure of the respective component. The values of these probabilities are input parameters for the model, thus allowing to analyse several different scenarios, depending on, for example, the devices used. Another task within ASTRail concerns the evaluation of such numbers, input to our model. For the analysis in Section 5, we abstract from these automata generating failures, which is eased by their separate modelling.

The failure of these components was not foreseen in the original Simulink model, where it can be simulated by the manual intervention of the user who wants to analyse the behaviour of the system in case of failure. Note that, in the Simulink/Stateflow specification, failure transitions could be activated by shared variables whose value is assigned by the user.

All components listed next are *templates* in Uppaal, which is a mechanism allowing to instantiate different instances of an automaton. This makes it possible to perform simulations and analyses with a certain number of RBCs, OBUs, and LUs; not fixed beforehand in the model. However, in line with the specification from our industrial partners, we assume that each component communicates with other components of the same index. For instance, RBC_0 always communicates with OBC_0 and never with OBC_1, who communicates with RBC_1. In reality, an RBC will have different threads, each one communicating with one train; each of these threads is an automaton. For simplicity, in the next section we will analyse the system considering only one OBC, one LU, and one RBC.

Furthermore, this model is parametric and highly customisable. It is possible to analyse different operational scenarios of the ASTRail moving block system by instantiating the individual parameters of the model. For instance, it is possible to customise the frequency of each of the various messages such as the frequency of requesting the location or the frequency of sending the MA. It is also possible to specify the size of the MA in terms of meters. Moreover, it is possible to model the acceleration of the train, as well as its average speed. By changing these parameters, we can perform different evaluations of the properties of interest, as we will show in the next section, so as to fine tune the setup of these parameters.

We briefly describe the model's components, followed by details of the OBU.

`OBU_MAIN_GenerateLocationRequest_T`: This automaton is the initial component that starts the system interactions and takes care of generating every few seconds a request for a new location to be sent to the LU.

`LU_MAIN_T`: This automaton models the LU. Its behaviour involves receiving a new position request from the OBU and replying with the current train location (computed via GNSS).

`OBU_MAIN_SendLocationToRBC_T`: This automaton, depicted in Fig. 5, is the main component of the OBU, and as such it performs a variety of operations.

The first operation is the reception of the position by the LU. Subsequently, with a certain frequency, this component sends the received position to the RBC. The same component moreover receives the MA from the RBC (after sending its position). Finally, it implements one of the safety mechanisms present in the system specification. In particular, at each instant of time, the model checks that the train's position has not exceeded the MA received from the RBC; if it has, it will enter a failure state. All components listed so far provide the possibility to enter a failure state if one of the probabilistic failures foreseen by the corresponding probabilistic automata occurs.

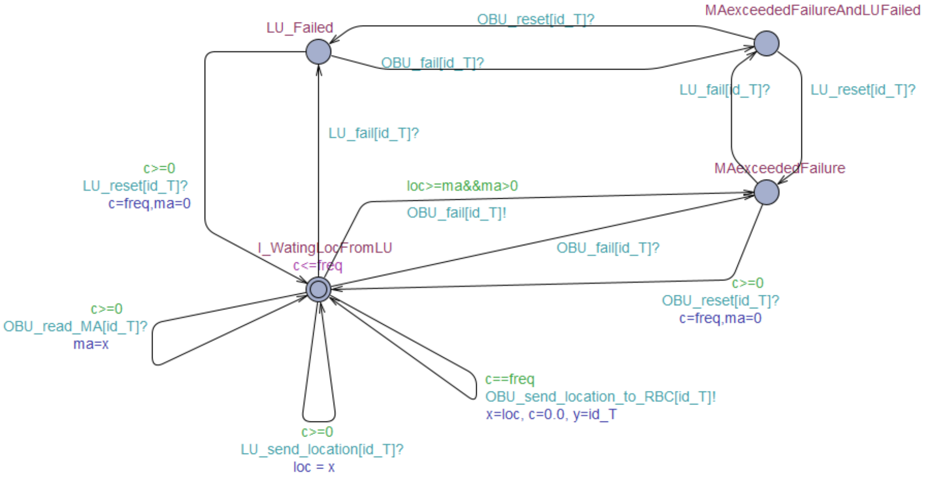


Fig. 5. The automaton `OBU_MAIN_SendLocationToRBC_T`

`RBC_Main_T`: This automaton models the RBC. It receives the MA request from the OBU. Once this request is received, the RBC sends a certain number of times the MA message until the corresponding acknowledgement from the OBU is received or the number of attempts is exceeded. Also this component, like all others, enters a failure state if one of the aforementioned errors occurs.

`OBU_MAIN_ReceiveMA_T`: This is the last automaton modelling the logic of the OBU. It receives an MA from the RBC and sends back a corresponding acknowledgement message. This component implements an additional safety mechanism of the system specification by means of a timer that counts the time passed from the reception of the last MA. In the event that this timer is exceeded, an alarm is emitted and a failure state (`TimeOutFail`) is entered.

`TRAIN_ATO_T`: This is a special component that was introduced to model more accurately the behaviour of a train. In particular, this component models the movement of the train, its speed, and the acceleration and deceleration that are triggered by approaching the limit described by the MA. This automaton also deals with simulating braking curves when a particular failure state is reached. In particular, the position of the train is stated in an unidimensional space and identified by one coordinate. Figure 6 shows the speed of the train and its sudden braking the moment it exceeds the MA.

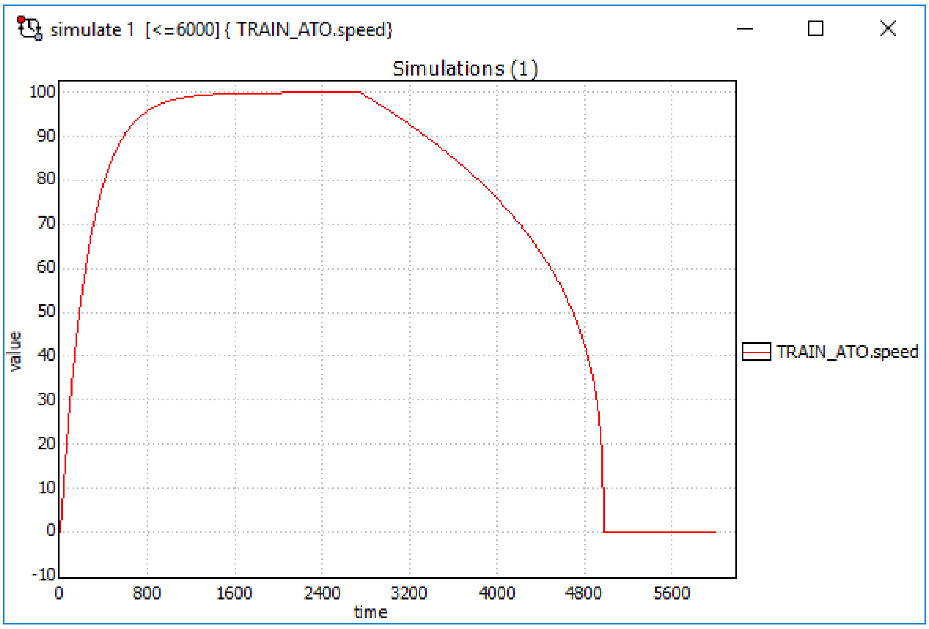


Fig. 6. A simulation showing the speed of the train in m/s

The OBU Model This automaton, depicted in Fig. 5, has four states. The initial state is the nominal state `I_WatingLocFromLU`, drawn with two circles, while the other three states represent system failures that are due to failure of the LU (`LU_Failed`), failure to receive the MA (`MAexceededFailure`), or both failures together (`MAexceededAndLUfailed`). The initial state has three outgoing transitions that have the same initial state as their target state (i.e., loops). The initial state also has an invariant to guarantee that the initial state's clock `c` is always less than or equal to the `freq` parameter, which represents the frequency of sending the location to the RBC.

In order of execution, the first transition to be performed is the one with signal `LU_send_location [id_T]?`. This action represents the reception of the position from the LU; `loc = x` represents the assignment of the variable `loc` that reads from the buffer variable `x` used to implement value passing. The transition with guard `c == freq` is activated exactly when the guard is satisfied, i.e., when the clock reaches the `freq` parameter. This transition implements a periodic operation which is carried out every instant of time `freq`. The action is that of sending the position data to the RBC. The sending operation is transmitted via the signal `OBU_send_location_to_RBC [id_T]!`, while the assignment of variables is `x = loc, c = 0.0, y = id_T`; i.e., the value `loc` of the location and the unique train identifier `id_T` are stored in the buffer variables, and the clock `c` is reset. Similarly, `OBU_read_MA [id_T]?` performs the reception from the RBC of the MA stored in the variable `ma`.

The outgoing transition from `I_WatingLocFromLU` to `MAexceededFailure` is activated by the guard `loc> = ma && ma> 0`; i.e., it is activated when the train position exceeds the MA. In this case, a failure signal is sent via the `OBU_fail [id_T]!` channel. Transitions in other failure states likewise encode reception of failure signals arriving from the LU. Finally, note that once the system restart message is received via the reset channel, the initial MA value is set to zero.

5 Analyses of ERTMS L3 Moving Block

Next to standard model-checking queries concerning reachability and deadlock-freedom, Uppaal SMC allows to check (quantitative) properties over simulation runs of an Uppaal model (i.e. a network of stochastic timed automata). For instance, Uppaal SMC supports the evaluation of the probability estimation $\mathbb{P}_M(\Diamond_{x\leq t}\, p)$ over a model M, where x is a clock, $t \in \mathbb{N}$, and p is a state predicate. Moreover, $\Diamond_{x\leq t}\, p = \mathit{true}\ U_{x\leq t}\ p$, in which U is a time-bounded Until operator of the form $p_1\ U_{x\leq t}\ p_2$, which is satisfied if p_1 holds on a simulation run *until* p_2 is satisfied, and this must happen before clock x exceeds time bound t. Apart from bounding over time, which may result in non-termination, we may bound runs for a number of discrete steps, which guarantees termination of the simulation. For a given model in Uppaal SMC, the query $\mathtt{Pr}[<= \mathtt{N}](<> \mathtt{p})$, where $\mathtt{N} \in \mathbb{N}$, is satisfied if $<> \mathtt{p}$ holds on a simulation run of at most $\mathtt{N}$ discrete steps.

We provide two temporal logic formulae to evaluate measures of interest of the moving block system. Both measure the probability of the Uppaal model entering a failure state within 1000 steps, namely when the train's position exceeds the MA (ϕ_1) or when the timeout for the reception of a new MA is exceeded (ϕ_2):

$$\phi_1 \overset{\mathsf{def}}{=} \mathtt{Pr}[<= 1000](<> \mathtt{OBU\_MAIN\_SendLocationToRBC.MAexceededFailure})$$

$$\phi_2 \overset{\mathsf{def}}{=} \mathtt{Pr}[<= 1000](<> \mathtt{OBU\_MAIN\_ReceiveMA.TimeOutFail})$$

We now show the potential of Uppaal SMC to analyse the modelled system for these properties of interest. The model has a myriad of possible parameters to fine tune. Here we limit ourselves to two different parameter setups, allowing to demonstrate the tool's effectiveness in confirming or rejecting parameter values.

We used academic version 4.1.19 (rev. 5649) of Uppaal SMC, with the probabilistic deviation set to 0.01, the probability of false negatives and false positives set to 0.005 and 0.5, respectively, and the probability uncertainty set to 0.005.

As mentioned before, the experiments instantiate one OBC, one LU, and one RBC (i.e. the experiments are performed with one train communicating with an RBC). Moreover, the automata generating probabilistic failures have been deactivated.

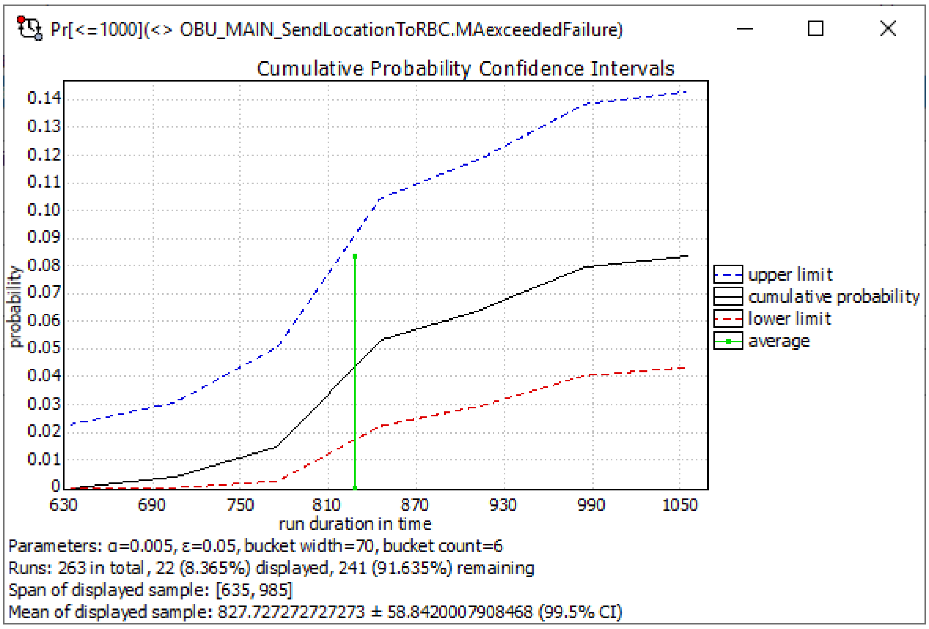
Table 1 contains the parameter values used in the experiments. The first experiment serves to confirm the correctness of the system specification received from the domain experts, which concerns both quantitative aspects (e.g., the MA size and the communication frequencies) and qualitative aspects (i.e., failure states). Our formalisation in Uppaal confirms that with the given parameter values the possibility to reach one of the failure states is indeed very low. More precisely, Uppaal SMC reports with 99.5% confidence the same interval $[0, 0.00998576]$ obtained from 597 runs after just under eight minutes.

Table 1. Parameter values used in the experiments

component (abbreviated)	parameter	value		description
		exp 1	exp 2	
RBC_Main	freq	1000s	1000s	frequency of sending MA to OBU before ack
RBC_Main	ma	1000m	500m	size of MA (in meters from current location)
OBU_MAIN_SendLocToRBC	freq	0.5s	5.0s	frequency of sending location to RBC
OBU_MAIN_GenerateLocReq	freq	0.5s	0.5s	frequency of sending location request to LU
OBU_MAIN_ReceiveMA	OBU_out_timer	10s	10s	timeout for receiving MA from RBC

We set up a second experiment to show that Uppaal SMC can also be used to reject parameter values that do lead to a high probability of failure and thus to hazardous scenarios. In this experiment there are less frequent updates of the train's position to the RBC and a tighter MA. Our formalisation in Uppaal confirms this to be an inappropriate parameter setup, as the probability for the train to exceed the MA (as expressed by formula ϕ_1) becomes high. Uppaal SMC reports with 99.5% confidence the interval $[0.0430205, 0.14268]$ obtained from 263 runs after approximately three minutes (cf. further details in Fig. 7).

This shows that further varying the parameters values, in principle the positive results of the first experiment could be improved. This would require more experiments and close interaction with the domain experts to understand which parameter values could theoretically be changed, without violating physical limits or fundamental requirements (e.g., an OBU cycle may take at most 500ms).


Pr[<=1000](<> OBU_MAIN_SendLocationToRBC.MAexceededFailure)

Parameters: α=0.005, ε=0.05, bucket width=70, bucket count=6
Runs: 263 in total, 22 (8.365%) displayed, 241 (91.635%) remaining
Span of displayed sample: [635, 985]
Mean of displayed sample: 827.727272727273 ± 58.8420007908468 (99.5% CI)

Fig. 7. The cumulative probability confidence interval of experiment 2

6 Lessons Learned

In this section, we report some lessons learned from our modelling and analysis experience of a satellite-based ERTMS L3 moving block railway signalling system with Simulink and Uppaal SMC.

Formal Evidence The analysis of two fundamental properties of the satellite-based ERTMS L3 moving block railway signalling system with Uppaal SMC, described in Section 5, provided further evidence for the applicability of Uppaal SMC in the railway domain (cf. also, e.g., [5, 6, 14, 29]). In particular, we showed the tool's potential for fine tuning communication parameters in satellite-based ERTMS L3 moving block railway signalling system models that are fundamental for the reliability of their operational behaviour. Given a specific parameter setup, we showed how to use Uppaal SMC to confirm or reject parameter values. The analysis of the Uppaal model for the parameter setup provided by our industrial partners confirmed the (desired) very low possibility to reach one of the failure states. Further analysis showed the capability of Uppaal SMC to detect a bad parameter setup.

Complementarity of Tools The starting point was a model in Real-Time UML [18, 47] and a set of requirements, both provided by our industrial partners [5]. The requirements elicitation and refinement activity performed with the industrial partners, as briefly outlined in Section 3, confirmed Simulink as an appropriate tool for the initial phases of a development process. Its simulation and debugging facilities increase confidence in the initial design and facilitate interaction with the industrial partners, thus allowing to consolidate a final set of requirements. Not surprisingly, the resulting model and requirements turned out to be far more detailed than the Real-Time UML model and the initial set of requirements. In this initial phase, the focus was on the elicitation and animation of functional requirements. At the same time, the probabilistic aspects initially introduced in the Real-Time UML model [5] could not be expressed in Simulink, thus requiring the usage of Uppaal. This confirms the need to introduce formal methods diversity [43] to properly address all the functional, quality and process-related aspects related to the development of railway signalling systems.

Transformation and Refinement The transformation of the Simulink model into an Uppaal model, described in Section 4, required us to revisit in particular the communication among the different processes, removing the fixed scheduling through shared variables in favour of action interleaving via broadcast channels. Moreover, in the Uppaal model time is no longer modelled by memorising the time difference between events in variables, but by explicit clocks. Most notably, Uppaal allows to model events to occur with certain probabilities and to consider stochastic timed behaviour, which we used to enrich the initial model according to additional specifications of the moving block system provided by our industrial partners. As such, the Uppaal model represents a refinement of the initial semi-formal Simulink specification into a more formal specification amenable to quantitative analyses.

Challenges We presented only some preliminary analyses in Section 5. Further properties of interest would require a more complex model with more than one train and more than one RBC, next to running more systematic experiments. Moreover, it remains to further vary the parameter values to investigate whether the parameter setup provided by our industrial partners can be improved. However, while it is not too difficult to use Uppaal SMC to either confirm or reject a parameter setup, it is much more difficult to use it to find an optimal parameter setup. We believe this requires profound knowledge of the statistical model-checking algorithms underlying the tool as well as of the tool's functionality, combined with expert knowledge from the railway domain concerning the physical limits of certain parameter values as well as best practices from the field.

7 Conclusion

In this paper, we have presented an experience in modelling a satellite-based ERTMS L3 moving block signalling system from the railway industry with Simulink (upon close interaction with the domain experts) and Uppaal and in performing preliminary analyses of the Uppaal model with Uppaal SMC (to be continued in close interaction with the domain experts). In the previous section, we have reported some lessons learned from this experience.

Future Work We plan to extend the model with more actors. This would allow us to consider properties like deadlocks (e.g., following [14], we could model several trains and use SMC to verify deadlock avoidance under intra communications). The work could also be extended by using Uppaal Stratego [17], an SMC and learning-based tool, to synthesise best routes to avoid deadlocking and match performance objectives (e.g., arrival delays). This would require a drastic modification of the model to introduce measure of performances. We could also see if Uppaal Stratego can be used for the optimisation of the model's parameters.

Finally, it would be worth to consider cyber attacks, e.g., by modelling the attacker and attacks with attack trees and combine the new model with that of the train. The result could be analysed via the Uppaal extension for cyber security [33]. Note, however, that this would be a major challenge as it would require a model of potential attacks (and thus know attacks typically kept secret).

Acknowledgements This work was partially funded by the Tuscany Region project SISTER (SIgnalling & Sensing TEchnologies in Railway application) and by the EU project ASTRail (SAtellite-based Signalling and Automation SysTems on Railways along with Formal Method and Moving Block validation), which received funding from the Shift2Rail Joint Undertaking under the EU's H2020 Research and Innovation programme under Grant Agreement No. 777561. The content of this paper reflects only the authors' view, and the Shift2Rail JU is not responsible for any use that may be made of the included information.

We thank our colleagues in the Formal Methods and Tools lab at ISTI-CNR and our project partners for discussions on the models analysed in the paper. We thank the four anonymous reviewers for their suggestions to improve the paper.

References

1. Agha, G., Palmskog, K.: A Survey of Statistical Model Checking. ACM Trans. Model. Comput. Simul. **28**(1), 6:1–6:39 (2018)
2. Arcaini, P., Ježek, P., Kofroň, J.: Modelling the Hybrid ERTMS/ETCS Level 3 Case Study in Spin. In: ABZ. LNCS, vol. 10817, pp. 277–291. Springer (2018)
3. Arnold, A., et al.: An Application of SMC to continuous validation of heterogeneous systems. EAI Endorsed Trans. Indust. Netw. & Intellig. Syst. **4**(10) (2017)
4. Bartholomeus, M., Luttik, B., Willemse, T.: Modeling and Analysing ERTMS Hybrid Level 3 with the mCRL2 toolset. In: FMICS. LNCS, vol. 11119. Springer (2018)
5. Basile, D., ter Beek, M.H., Ciancia, V.: Statistical Model Checking of a Moving Block Railway Signalling Scenario with Uppaal SMC. In: ISoLA. LNCS, vol. 11245, pp. 372–391. Springer (2018)
6. Basile, D., Di Giandomenico, F., Gnesi, S.: Statistical Model Checking of an Energy-Saving Cyber-Physical System in the Railway Domain. In: SAC. pp. 1356–1363. ACM (2017)
7. Basile, D., et al.: On the Industrial Uptake of Formal Methods in the Railway Domain. In: iFM. LNCS, vol. 11023, pp. 20–29. Springer (2018)
8. ter Beek, M.H., Fantechi, A., Ferrari, A., Gnesi, S., Scopigno, R.: Formal Methods for the Railway Sector. ERCIM News **112**, 44–45 (2018)
9. ter Beek, M.H., Gnesi, S., Knapp, A.: Formal methods for transport systems. Int. J. Softw. Tools Technol. Transf. **20**(3), 355–358 (2018)
10. ter Beek, M.H., Legay, A., Lluch Lafuente, A., Vandin, A.: Statistical Model Checking for Product Lines. In: ISoLA. LNCS, vol. 9952, pp. 114–133. Springer (2016)
11. Behrmann, G., et al.: UPPAAL 4.0. In: QEST. pp. 125–126. IEEE (2006)
12. Beugin, J., Marais, J.: Simulation-based evaluation of dependability and safety properties of satellite technologies for railway localization. Transport. Res. C-Emer. **22**, 42–57 (2012)
13. Boulanger, J.L. (ed.): Formal Methods Applied to Industrial Complex Systems — Implementation of the B Method. John Wiley & Sons (2014)
14. Cappart, Q., et al.: Verification of Interlocking Systems Using Statistical Model Checking. In: HASE. pp. 61–68. IEEE (2017)
15. Cunha, A., Macedo, N.: Validating the Hybrid ERTMS/ETCS Level 3 Concept with Electrum. In: ABZ. LNCS, vol. 10817, pp. 307–321. Springer (2018)
16. David, A., Larsen, K.G., Legay, A., Mikučionis, M., Poulsen, D.B.: Uppaal SMC tutorial. Int. J. Softw. Tools Technol. Transf. **17**(4), 397–415 (2015)
17. David, A., et al.: On Time with Minimal Expected Cost! In: ATVA. LNCS, vol. 8837, pp. 129–145. Springer (2014)
18. Douglass, B.P.: Real-Time UML. In: FTRTFT. LNCS, vol. 2469, pp. 53–70. Springer (2002)
19. EEIG ERTMS Users Group: ERTMS/ETCS RAMS Requirements Specification — Chapter 2 - RAM (30 09 1998)
20. EEIG ERTMS Users Group: System Requirements Specification v3.6.0 - SUBSET-026 (15 06 2016)
21. EEIG ERTMS Users Group: Hybrid ERTMS/ETCS Level 3: Principles (14 07 2017)
22. European Committee for Electrotechnical Standardization: CENELEC EN 50128 — Railway applications – Communication, signalling and processing systems – Software for railway control and protection systems (01 06 2011)
23. Fantechi, A.: Twenty-Five Years of Formal Methods and Railways: What Next? In: SEFM. LNCS, vol. 8368, pp. 167–183. Springer (2013)
24. Fantechi, A., Ferrari, A., Gnesi, S.: Formal Methods and Safety Certification: Challenges in the Railways Domain. In: ISoLA. LNCS, vol. 9953, pp. 261–265. Springer (2016)

25. Fantechi, A., Fokkink, W., Morzenti, A.: Some Trends in Formal Methods Applications to Railway Signaling. In: Formal Methods for Industrial Critical Systems: A Survey of Applications, chap. 4, pp. 61–84. John Wiley & Sons (2013)
26. Ferrari, A., Fantechi, A., Gnesi, S., Magnani, G.: Model-Based Development and Formal Methods in the Railway Industry. IEEE softw. **30**(3), 28–34 (2013)
27. Ferrari, A., Fantechi, A., Magnani, G., Grasso, D., Tempestini, M.: The Metrô Rio case study. Sci. Comput. Program. **78**(7), 828–842 (2013)
28. Ferrari, A., et al.: Survey on Formal Methods and Tools in Railways: The ASTRail Approach. In: RSSRail. LNCS, vol. 11495, pp. 226–241. Springer (2019)
29. Filipovikj, P., et al.: Simulink to UPPAAL Statistical Model Checker: Analyzing Automotive Industrial Systems. In: FM. LNCS, vol. 9995, pp. 748–756. Springer (2016)
30. Flammini, F. (ed.): Railway Safety, Reliability, and Security: Technologies and Systems Engineering. IGI Global (2012)
31. Fränzle, M., Hahn, E., Hermanns, H., Wolovick, N., Zhang, L.: Measurability and safety verification for stochastic hybrid systems. In: HSCC. pp. 43–52. ACM (2011)
32. Furness, N., van Houten, H., Arenas, L., Bartholomeus, M.: ERTMS Level 3: the Game-Changer. IRSE News **232**, 2–9 (April 2017)
33. Gadyatskaya, O., et al.: Modelling Attack-defense Trees Using Timed Automata. In: FORMATS. LNCS, vol. 9884, pp. 35–50. Springer (2016)
34. Ghazel, M.: Formalizing a subset of ERTMS/ETCS specifications for verification purposes. Transport. Res. C-Emer. **42**, 60–75 (2014)
35. Ghazel, M.: A Control Scheme for Automatic Level Crossings Under the ERTMS/ETCS Level 2/3 Operation. IEEE Trans. Intell. Transp. Syst. **18**, 2667–2680 (2017)
36. Gilmore, S., Tribastone, M., Vandin, A.: An Analysis Pathway for the Quantitative Evaluation of Public Transport Systems. In: IFM. LNCS, vol. 8739, pp. 71–86. Springer (2014)
37. Harel, D.: Statecharts: a visual formalism for complex systems. Sci. Comput. Program. **8**(3), 231–274 (1987)
38. Herde, C., Eggers, A., Fränzle, M., Teige, T.: Analysis of Hybrid Systems Using HySAT. In: ICONS. pp. 196–201. IEEE (2008)
39. Larsen, K.G., Legay, A.: Statistical Model Checking — Past, Present, and Future. In: ISoLA. LNCS, vol. 8802, pp. 135–142. Springer (2014)
40. Littlewood, B., Popov, P., Strigini, L.: Modeling Software Design Diversity: A Review. ACM Comput. Surv. **33**(2), 177–208 (2001)
41. Mammar, A., Frappier, M., Tueno Fotso, S.J., Laleau, R.: An Event-B Model of the Hybrid ERTMS/ETCS Level 3 Standard. In: ABZ. LNCS, vol. 10817, pp. 353–366. Springer (2018)
42. Mazzanti, F., Ferrari, A.: Ten Diverse Formal Models for a CBTC Automatic Train Supervision System. In: MARS. EPTCS, vol. 268, pp. 104–149 (2018)
43. Mazzanti, F., Ferrari, A., Spagnolo, G.O.: Towards formal methods diversity in railways: an experience report with seven frameworks. Int. J. Softw. Tools Technol. Transf. **20**(3) (2018)
44. Nardone, R., et al.: Modeling Railway Control Systems in Promela. In: FTSCS. CCIS, vol. 596, pp. 121–136. Springer (2016)
45. Puch, S., Fränzle, M., Gerwinn, S.: Quantitative Risk Assessment of Safety-Critical Systems via Guided Simulation for Rare Events. In: ISoLA. LNCS, vol. 11245, pp. 305–321. Springer (2018)
46. Rispoli, F., et al.: Recent progress in application of gnss and advanced communications for railway signaling. In: RADIOELEKTRONIKA. pp. 13–22. IEEE (2013)
47. Selic, B.: The Real-Time UML Standard: Definition and Application. In: DATE. pp. 770–772 (2002)
48. UNISIG: FIS for the RBC/RBC handover, version 3.1.0 (15 06 2016)