A Model-based Approach for the Modeling and the Verification of Railway Signaling System

Racem Bougacha[1], Abderrahim Ait Wakrime[1], Slim Kallel[2], Rahma Ben Ayed[1]
and Simon Collart-Dutilleul[1,3]

[1] *Institut de Recherche Technologique Railenium, F-59300, Famars, France*
[2] *ReDCAD, University of Sfax, Tunisia*
[3] *IFSTTAR-Lille, 20 Rue Elisée Reclus BP 70317, 59666 Villeneuve d'Ascq Cedex, France*

Keywords: Railway Signaling System, Model-Driven Engineering, Formal Methods, Verification, Model Checking.

Abstract: Railway Signaling System aims at resolving interoperability problems of railway signaling and train control/-command. An European system is taken place to guarantee interoperability between international borders. Such complex systems require a high-level of safety. We propose an approach for modeling and verifying railway signaling systems. Our approach consists of graphical modeling such systems and automatically generating formal specification in Event-B. It is based on model-driven engineering techniques. We propose model-to-model transformation to generate Event- B model from UML class diagram profiled with safety and railways concepts. A textual Event- B code is also automatically generated using model-to-text transformation. The proposed approach allows to formally verify the safety of railway signaling systems based on model checking and animator tool.

1 INTRODUCTION

European countries launched a major system named ERTMS/ETCS (Schön et al., 2013) to harmonize the variety of railway signaling in Europe. This system serves to guarantee the interoperability of European railway network and thus by eliminating the need of the locomotives changes at border points between countries. In addition, its aim is to improve maintenance costs, reliability and performance. ERTMS/ETCS is a control-command and signaling system and it is composed of many software and material components which they communicate with each other.

Safety is a major challenge for this type of systems given the complexity and serious consequences that may arise from analysis and design errors. Particularly, in ERTMS/ETCS system, safety requirements is the central concern in the development process (Ait Wakrime et al., 2018; Beugin et al., 2010). For this purpose, safety ensures the proper functioning of such system and guarantees traffic safety. Likewise, some railway operating rules are defined in order to respect some specific functions and their implementations regarding the safety requirements. Modeling of these railway operating rules requires the definition of some concepts and the relationships between them. In this context, we define a railway operating rule as a sequence of authorizations to execute some operations (actions) by a set of users (human or software actors) according to roles (driver, traffic agent, etc.) which assign the responsibilities granted to users.

In response to this need, this paper addresses two main issues, namely (1) Non-existence of a holistic approach for modeling safety railway signaling systems. This issue is stemming from, on the one hand, the existence of safety rules defined in an informal way which does not allow its maintainability and its definition in a formal way to be checked and validated. On the other hand, the research works who tried to define safety aspects for railway and the ones who tried to adapt security aspects (especially access control) of information system to railway domain still couldn't support all the railway safety specification. (2) Inadequacy of the existing formal approaches to specify these systems since railway systems are based on communication with signals exchange. The latter describes the interaction between the different entities

Bougacha, R., Ait Wakrime, A., Kallel, S., Ben Ayed, R. and Collart-Dutilleul, S.
A Model-based Approach for the Modeling and the Verification of Railway Signaling System.
DOI: 10.5220/0007728403670376
In *Proceedings of the 14th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE 2019)*, pages 367-376
ISBN: 978-989-758-375-9

that manage the railway system and which describes critical systems where there is no place for errors.

Basically, we aim to guarantee valid railway signaling systems. In order to achieve this goal, we propose a holistic approach for modeling and formally verifying railway signaling systems. This process is composed of three main steps: (i) Graphical Modeling of railway systems. First, we present a UML profile that describes safety aspects of railway system to provide a generic extension mechanism for specializing and customizing UML class diagram meta-model in order to introduce our new safety concepts defined in the previous meta-model. (ii) Model Transformation. First we create two new meta-models. The first one is Event-B meta model, which is an extension of the B-Method meta-model, while the second one is Safety Railway meta-model describing safety aspects of railway system. Thereafter, we propose also an automatic model-to-model transformation from UML class diagram to new models conforming to proposed meta-model. (iii) Event-B Generation & Formal verification. First, we aim at this step to generate a textual specification containing Event-B code using a model-to-text transformation in order to apply a formal verification on it with model checker supporting the generated specification. Thereafter, the generated Event-B code is introduced in model checker and animator to detect errors and to prove the correctness of systems.

Our approach offers a better way to reduce the complexity by defining railway system formally. It aims to represent their needs for safety by introducing their safety concepts. Based on model-driven engineering, our approach provides modular and maintainable safety specification that could be verified formally and which guarantee the safety of such systems.

The paper is structured as follows: Railway signaling system ERTMS/ETCS is described in the next Section. Section 3 presents our approach. Graphical modeling of our approach is detailed in Section 4. In Section 5, model transformation of the proposed approach is presented. Section 6 describes the generation of our Event-B specification and its validation using model checking and animation. Section 7 discusses the related work. Section 8 concludes the paper.

2 RAILWAY SIGNALING SYSTEM ERTMS/ETCS

The European Rail Traffic Management System/European Train Control System (ERTMS/ETCS) aims at resolving interoperability problems of railway signaling and train control/command (Schön et al., 2013). ERTMS/ETCS is an European system to manage railway traffic and to guarantee interoperability between international borders. Indeed, it allows to provide a compatible signaling systems among countries. ERTMS/ETCS in Europe has a significant benefits regarding safety, cost, interoperability, accessibility and maintenance.

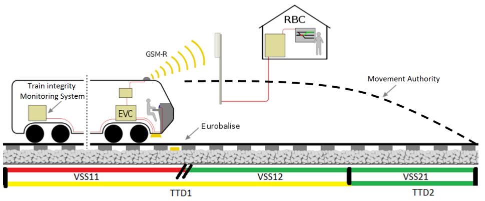

Figure 1: ERTMS/ETCS.

The Figure 1 presents Hybrid ERTMS/ETCS system. Trackside Train Detection (TTD) is divided into several Virtual Sub-Sections (VSS). A TTD is a section determined by a conventional trackside train detection system like track-circuits or axle-counters. VSS is used to define a MA before assigning it to the train. Movement of the train depends on the VSS state and the train location is provided according to the VSSs. The Figure 2 shows an extract of Unified Modeling Language (UML) class diagram of studied system.

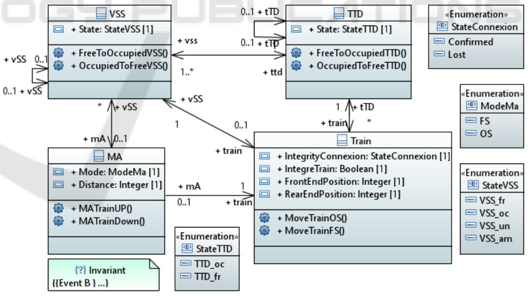

Figure 2: An extract of class diagram of Hybrid ERTMS/ETCS.

VSS guarantees the safety of the system since it allows the spacing between a foregoing train and the chasing train. The presence of a train detected by a TTD in a given VSS, makes the VSS state *Occupied* (the red VSS in Figure 1). If a TTD did not detect a train in a VSS, the VSS state becomes *Free* (the green VSS in Figure 1). When a TTD is not sure that a train is hidden behind another train at the same VSS, the VSS state becomes *Ambiguous*. State *Unknown* represents the case where the TTD doesn't know the position train and is not sure that the VSS state is *Free*. In the nominal situation, one train owns a MA to move to a given point as a supervised movement. This point is determined according to the position of the chasing trains in terms of VSSs.

3 THE PROPOSED APPROACH

Our approach is intended to model railway signaling system, which is a critical system that needs a high level of safety, security and integrity. The main contribution targeted by our approach is to integrate safety properties and apply its aspects on the functional specification of the railway system.

3.1 Safety Properties of Railway System

Safety is a major challenge for critical systems given the complexity and serious consequences that may arise from analysis and design errors. It is defined as a property of a system, which does not in any way endanger neither persons nor its environment. Another definition is a removal of unacceptable risk. Particularly, in ERTMS/ETCS system, safety requirements is the central concern in the development process (Beugin et al., 2010; Ait Wakrime et al., 2018) since human errors, unforeseen failures, and different combinations of unfavorable situations may negatively influence the safety of various elements or even of the whole railway system. Safety of a railway system is dependent on the balance of *Man*, *Machine* and *Environment* (Cicmancova, 2013).

For this purpose, safety ensures the proper functioning of such system and guarantees traffic safety. Different properties need to be identified and verified, *e.g.* : (a) Controlling that the train does not exceed the maximum authorized speed nor the different speed limits assigned to the various track sections. (b) Checking that each train is in the right direction and does not exceed the limited authorized area. (c) Checking if a train is attributed to zone and that a zone contains one and only train. (d) Checking that the MA of each train is correctly established, that sufficient protections have been put in place to prohibit the entry of other trains in an area assigned to a specific train and that the points of the different switches are locked in the correct position so that the train moves along the planned itinerary.

Likewise, some railway operating rules are defined in order to respect some specific functions and their implementations regarding the safety requirements. In fact, the railway operating rules defines the train movement authorizations which are specified with a set of constraints, software and human system actors. To analyze these movement authorizations several questions (Ben Ayed et al., 2014) arise: Who does what? What are the authorizations? Who are the responsible of these authorizations? Which resources are affected by these authorizations? What actions are enabled by these authorizations? What are the constraints related to these authorizations?

So the essential concepts that we could extract from these questions are: System actors, authorizations, authorization's responsible, concerned resources, and authorized actions and associated constraint.

We can see following these concepts a rapprochement with the RBAC (Ferraiolo et al., 1995) model concepts describing security concepts of information systems and the safety concepts of railway systems, noting that:

- A system actors correspond to RBAC users.
- Responsibilities accorded to actors correspond to RBAC roles accorded to RBAC users.
- The authorization concept corresponds to RBAC permission concept.
- Authorized actions correspond to RBAC operations.
- Concerned resources correspond to RBAC objects.
- Authorization constraints correspond to permissions constraints.

Following this correspondence, a designer can profit from a standardized model such as RBAC model developed by the National Institute of Standards and Technology (NIST) in order to define railway safety concepts instead of starting from scratch. Unfortunately, RBAC concepts does not cover all safety aspects required by the railway system. Therefore, we add a new concept called *Mode*, which will precise the circulation mode of a train. This concept is specialized with two types *Normal* or *Degraded* and each of which offers a different degree of supervision and protection. The *Normal* mode describes that the system is working as planned, while the *Degraded* mode represents situation where all or part of the system should work without their usual resources.

Adding to that, and in order to argue the use of the *Normal* or the *Degraded* mode, specifying contexts that depends on the circumstances in which these modes are granted is required.

The *context* in railway system can depend on the time, the location, the activation of an action by a user or the history of actions. According to literature, and in correspondence to our requirement of the concept *context*, a similar concept was defined in the ORBAC model (El Kalam and Deswarte, 2006), which satisfies the same properties needed for the *context* in railway system. Therefore, we adapt ORBAC *context* to the railway system. An ORBAC *context* allows to switch from static permissions to specific dynamic permissions to concrete circumstances in which systems grant users permissions to perform actions on objects. In the railway system, the *context* is used to express notions on which ERTMS/ETCS specifications are based, such as the different levels of ERTMS, the mode situations, the conditions of transition from a procedure to an other, etc. The different *context* types described in (Cuppens and Miège, 2003) are: (i) *Temporal context*: defines the period of validity at which the subject is requesting to the system. (ii) *Spatial context*: depends on a network belonging or a geographical position of the subject. (iii) *User-declared context*: depends on the subject objective. (iv) *Prerequisite context*: depends on the domain specific characteristics (subject, the action and the object). (v) *Provisional context*: it depends on the history of the actions that the subject had the privilege to perform. However, in order to define all the required context of railway system, train movement constraints needs in addition of these types a new context named *Spatio-Temporal* that we defined to describes the dynamic train movement, *e.g.* speeds constraints.

Finally, in order to define safety aspects in modular way we have introduced the ORBAC organizations concept, which allows to model the structure of the real organizations. In our study, ERTMS/ETCS can be considered as a parent organization. Any specific line in a country equipped with ERTMS/ETCS is a sub-organization that can inherit the generic safety rules of the parent organization, and also add or remove rules and thus define its own railway operating rules. All this point are illustrated in Figure 3 describing the new safety model for railway system.

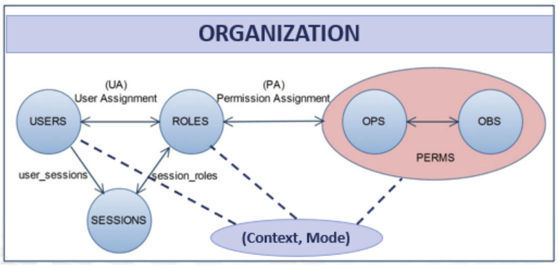

Figure 3: the proposed safety model.

3.2 Approach in a Nutshell

Our approach is based on model-driven engineering (MDE) approach and it presumes a three step process as shown in Figure 4: Graphical modeling, Model transformation and Event-B generation & formal verification. In the first step, we propose a UML profile which is inspired from SecureUML (Lodderstedt et al., 2002) used to enrich class models in UML with stereotypes and constraints specific for safety of railway system. The designer specifies:

- The model of the railway system using UML class diagram to define the functional part, which describes the different entities of the system.
- The non-functional part, which uses functional entities to specify the safety properties on it using a specialization of the UML class diagram components applied by the UML profile.

The second step corresponds to an automatic transformation which consists of mapping the class diagrams elaborated in the first step to a target models:

- The functional class diagram is transformed to an Event-B model which contains several elements of the Event-B notation. Since there is no standard that describes and groups the components of Event-B specification, a new meta-model has been proposed and which the generated Event-B model is conform to.
- Non-functional class diagram profiled with safety properties is transformed to a railway safety model which contains the different components describing safety aspects of railway system. To regroup this components, we proposed a new meta-model called *Safety Railway Meta Model*, so that we can navigate on railway specific concepts that will be more intuitive than navigating on UML concepts for our domain. Therefore our railway safety model will be conforms to the proposed meta-model.

Finally, the last step is composed of two parts: Event-B generation and formal verification. The Event-B generation consists of a model to text transformation which generates a textual specification from the models created in the second step.

- We proposed a template that generates Event-B textual specification, which describes the functional specification of the system, from the Event-B model obtained in the previous step.
- A second template is used to generate another Event-B textual specification, from the railway safety model generated in the previous step, which applies safety aspects on the functional concepts by filtering the access to the functional model.

In order to detect the design errors and to discover some invariant violation, the formal verification step of the generated specification is important. Therefore, we used a model checker and animator which support our specification. Using the model checker executed with different strategies, we guarantee the correctness of our specification.

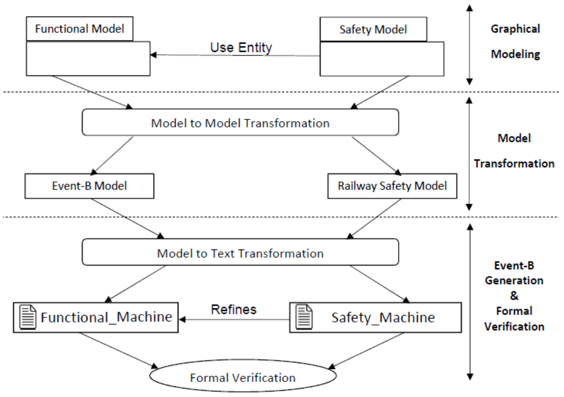

Figure 4: The proposed approach.

4 GRAPHICAL MODELING

The graphical modeling consists of modeling of functional and non-functional aspects of the railway system. It focuses on different entities that should be existed in the system. In our approach, this step is defined with a UML class diagram to specify, view, understand, and document the system in a simple way. A class diagram is one of the leading diagrams in a UML modeling. It allows to dissect the system by showing its components (classes).

To define a railway system, its functional description should contains a set of elements which describes the software and hardware parts interacting together to complete the specification of such a system. So as our input functional class diagram, the designer should model all components of a railway system like the Train, the Driver, the Rail, the Movement Authority, etc and the interaction between them.

To define non-functional railway system specification, a UML class diagram is used to describe and precise the application of safety aspects on the functional entities. To precise and specialize UML elements with safety properties a UML profile called *Railway SecureUml Profile* is proposed.

4.1 Railway SecureUML Profile

We proposed a UML profile as an extension of the UML class diagram for modeling safety railway system. This profile, inspired from SecureUML (Lodderstedt et al., 2002), represents the essential concepts of safety of railway system previously explained in 3.1. Our profile defines all safety aspects of railway system using RBAC concepts, *context* and *organization* of ORBAC model and the new concept of *Mode* dedicated to railway system. In addition, our profile contains a set of OCL constraints to impose some restrictions on the defined stereotypes.

As we can see in the Figure 5, for example, the stereotypes *Role* and *User* extends the Meta-Class *Class*, while the *organization* extends the Meta-Class *Class* and the Meta-Class *Property* too. In addition, the different types of *Context* and *Mode* are represented as enumeration and are introduced in the profile as properties for the actions which extends the Meta-Class *Operation*. *User*, *Role* or *Organization* are extensions of Meta-Class *Class*. This categorization of Meta-Class *Class* allows to remove the ambiguity of its different specialization and facilitate the mapping to the target concepts. In order to give more completeness to the definition of this constraint, it should be expressed in a formal and standardized way by using the OCL language.

Figure 6 shows the application of *Railway secureUML profile* on entities of functional model of the Figure 2, exactly *Train* and *MA* entities. In our example, the three roles *Driver*, *OnboardSafetyManagement* and *TracksideSafetyManagement* are modeled by a UML class. Each role belongs to an organization like *ERTMS*. The permissions are drawn as an associations class. The permission *Driver_TrainPerm* defines a permission for the role *Driver* to access the *Train* with the actions *MoveTrainOS* and *MoveTrainFS*. The permission *OSM_MAPerm* defines a permission for the role *OnboardSafetyManagement* to access the *MA* with the action *ValidateMA*. It is the same for the permission *TSM_MAPerm*, see Figure 6.

5 MODEL TRANSFORMATION

In the first part of this step, we adopt model-to-model transformation using a simple *java* code. In fact, the first part of this transformation consists of automatically map functional class diagram describes in the *Graphical Modeling* step into an Event-B model. Event-B formal method is chosen because it specifies all the system functionalities and complexity and it provides a formal specification based on *Set theory* and *Predicate logic* which guarantee rigorously the correctness and express the semantics involved in such systems. Event-B is an extension of the B-method (Abrial, 2010b), which models only the software part of system, contrary to Event-B which defines the software and the hardware parts (Abrial, 2010a) of system such as railway system.

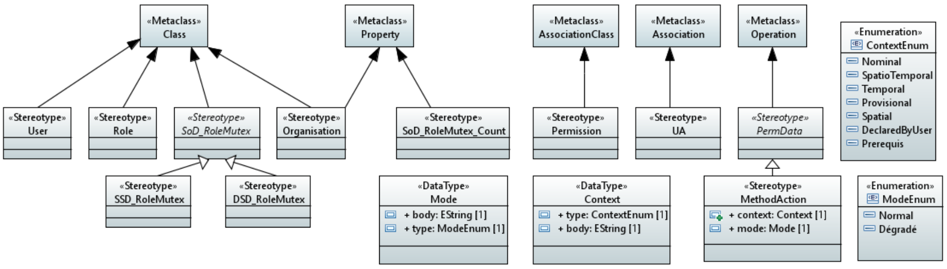

Figure 5: Railway SecureUml Profile.

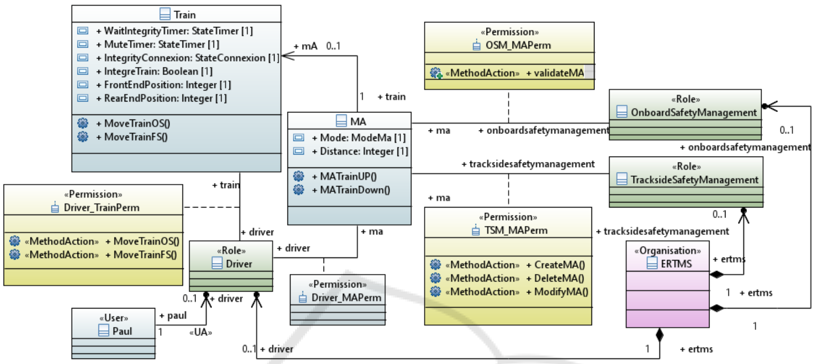
Train
+ WaitIntegrityTimer: StateTimer [1]
+ MuteTimer: StateTimer [1]
+ IntegrityConnexion: StateConnexion [1]
+ IntegreTrain: Boolean [1]
+ FrontEndPosition: Integer [1]
+ RearEndPosition: Integer [1]
+ MoveTrainOS()
+ MoveTrainFS()
+ mA 0..1
1 + train
MA
+ Mode: ModeMa [1]
+ Distance: Integer [1]
+ MATrainUP()
+ MATrainDown()
«Permission»
OSM_MAPerm
«MethodAction» + validateMA
+ ma
+ onboardsafetymanagement
«Role»
OnboardSafetyManagement
0..1
+ onboardsafetymanagement
+ tracksidesafetymanagement
«Role»
TracksideSafetyManagement
+ ma
0..1
+ train
«Permission»
Driver_TrainPerm
«MethodAction» + MoveTrainOS()
«MethodAction» + MoveTrainFS()
+ driver
«Role»
Driver
+ driver
+ ma
«Permission»
TSM_MAPerm
«MethodAction» + CreateMA()
«MethodAction» + DeleteMA()
«MethodAction» + ModifyMA()
+ tracksidesafetymanagement
«Organisation»
ERTMS
+ ertms
1 + ertms
1
«Permission»
Driver_MAPerm
0..1 + driver
«User»
Paul
+ paul
1
«UA»
0..1 + driver
1
+ ertms

Figure 6: Permissions to move train and MA management in the specified context and mode.

5.1 Event-B Meta Model

A standardized meta-model for Event-B is still not available. However, the one proposed in the Rodin Platform (Abrial et al., 2010) is of high complexity and cannot cover our needs. Therefore, we propose the meta-model depicted in Figure 7 that shows clearly Event-B meta-classes and structures and which is an extension of the B-Method meta model (Idani, 2006) used in the B4MSecure Platform (Idani and Ledru, 2015).

In Event-B meta-model, *EventB_SPEC* is the root meta-class. It is composed of zero or many *CONTEXT* and zero or many *MACHINE*. The extension proposed between our meta-model and B-Method meta-model is that our root meta-class "*EventB_SPEC*" inherits from the root meta-class of B-Method "*BSpec*". A *CONTEXT* describes the static part of a system, it is composed of a zero or many *CONSTANTS* and zero or more *AXIOMS* that they could be obligatory in the presence of constant because they defines the constants types. Adding to that, a context could be extended with zero or one context and it could be seen with a machine.A *MACHINE* describes the dynamic part of a system, it is composed of zero or more *VARIABLES*, *INVARIANTS*, *EVENTS*, *VARIANTS* and *ASSERTIONS*. *VARIABLES* represents the list of state variables of the model. *INVARIANTS* represents the typing predicates of the various variables and the predicates that the variables should obey. *ASSERTIONS* shows the different assertions which have to been proved within the machine. *EVENTS* represents the list of the various events related to the model. *VARIANTS* appears in a machine containing some convergent events. Adding to that, a machine could refines or be refined by another machine and it could see zero or many contexts.

Safety properties are principal aspects that should be respected in railway system specially and critical systems generally. That's why, the specification of such a system should be enriched with safety aspects. Therefore, in the second part of this transformation step, we used the standardized MDE-based model-to-model transformation language which is QVT. This transformation is to automatically map UML class diagram profiled with our proposed profile *Railway SecureUML profile* into safety model using transformation rules in order to define and represent the different safety properties required for railway system in a single model. Therefore, after specializing UML class diagram of the non-functional part with safety properties, a model type with all these types of safety elements and their constraints of railway system is required.

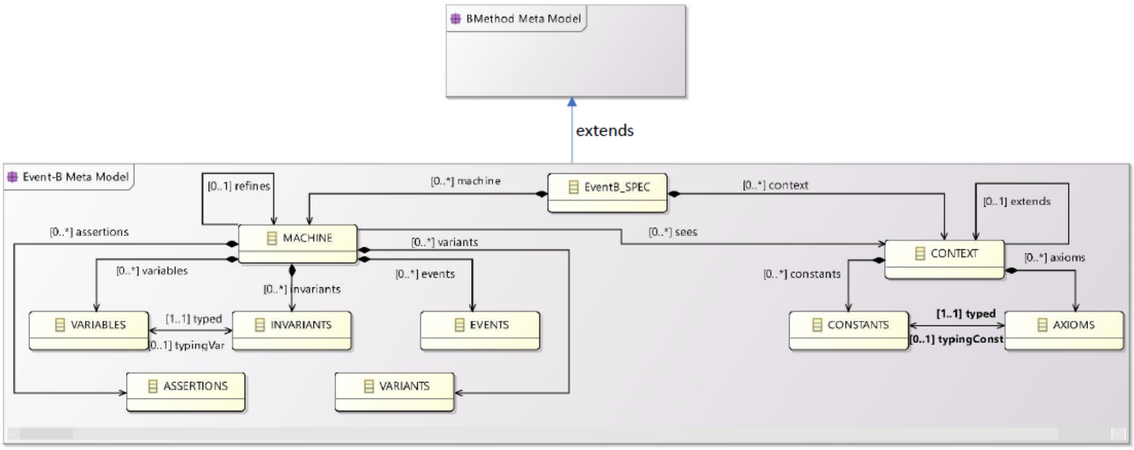

Figure 7: Event-B Meta Model.

5.2 Safety Railway Meta Model

The meta-model proposed in the B4MSecure Platform (Idani and Ledru, 2015) represents the concepts of RBAC. This meta-model does not cover the safety properties required for railway system, so its adaptation for railway system will suffer of the lack of safety properties. Therefore, we propose the meta-model illustrated in Figure 8 that introduces clearly Safety Railway meta-classes and structures.

Our meta-model is an extension of the RBAC meta-model used in the B4MSecure Platform (Idani and Ledru, 2015) in which *Policy* is the root meta-class and contains all the principal components of RBAC. Adding to that, and to cover all the complexity of the safety railway system we add the two ORBAC concepts as described in the Section 3.1 which are *Organization* and *Context*. The policy could contain zero or more Organizations and the *Context* is in a relation with the *Mode* which is a new concept that we added to cover all behaviors of the railway safety aspects. However, the proposed new type of *Context Spatio-Temporal* which describes the dynamic train movement, for example, speeds constraints, in order to cover all behaviors of train movement constraints, is defined as an inheritance from the ORBAC *Context*.

6 EVENT-B GENERATION & FORMAL VERIFICATION

The Event-B generation step consists of a model-to-text transformation in which, textual specification is generated representing *Event-B code*. The Event-B generation process is composed of two parts. The first corresponds to the generation of functional specification code including functional entities and concepts of railway system from the *Event-B Model* created in the transformation step and does not contains any safety related code. The second part correspond to the generation of the Event-B textual specification including railway safety aspects which will be generated automatically from the *Railway Safety model* created in the previous step. The generated Event-B safety code, will integrate modularly functional specification to verify if the specified safety constraints and authorizations are satisfied.

Using Acceleo (Musset et al., 2006), we create a new template, which represents the different aspects of Event-B. Our template describes that each component in Event-B model (step 2) will be translated to the Event-B language that describes functional textual specification with Event-B code.

Likewise, we propose to generate an Event-B textual specification to implement the safety concerns in the specification of the railway system application. Thus, we define Acceleo template to generate an Event-B code from these concerns according to the Event-B code. Our template describes that components in safety model will be translated to the Event-B components that describes the non-functional safety specification which refines the functional specification with safety behaviours. Thus, for example, a transformation rule creates the different Events including the permissions that allow the execution of the treatment to be performed by the Event.

6.1 Verification and Validation

We adopted the verification of our generated formal specification using ProB model-checker and animator. After adding manually the safety invariants by a competent person in Event-B (either at the level of

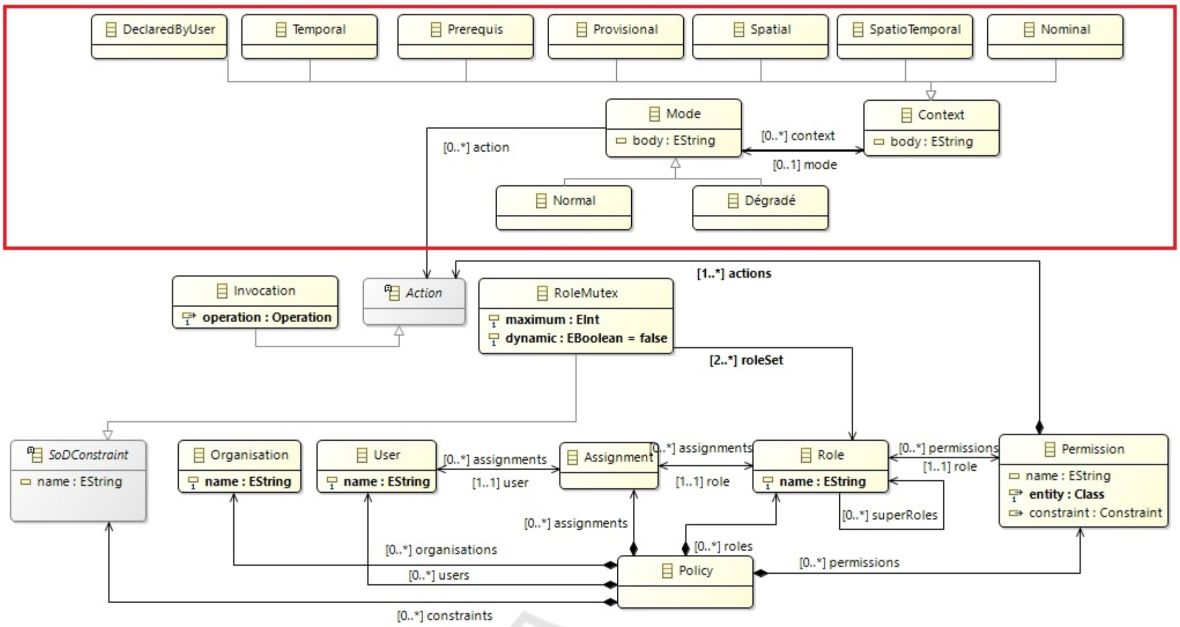

Figure 8: Safety Railway Meta Model.

the first step of our approach, exactly in the *Invariant* constraint of Figure 2, or after specification generation), the correctness of the model is proved in order to achieve a reliable system prior to animation. In the sequel, we use ProB to discover some errors and invariant violation during the model animation or during the verification using model checking.

We considered a model with finite state spaces in terms of number of Train, TTDs and VSSs. As a model example, we have defined two trains, two TTDs when each one consists of four VSSs. In addition, two safety invariants are added to avoid trains accidents under normal and degraded modes, Figure 9. The first one represents that the sets of VSSs of MAs are mutually disjoint. The second one checks that the VSSs of the MA are all obligatorily free.

MACHINE $HybridERTMSETCS3$
INVARIANT
\\Invariant 1
$\forall(tr1,tr2).(tr1 \in Train \wedge tr2 \in Train \wedge tr1 \neq tr2 \Rightarrow$
$MA\_VSS(Train\_MA(tr1)) \cap MA\_VSS(Train\_MA(tr2)) = \emptyset) \;\wedge$
\\Invariant 2
$\forall ma.(ma \in MA \wedge MA\_Mode(ma) = FS \Rightarrow$
$\forall vss.(vss \in VSS \wedge vss \in MA\_VSS(ma) \Rightarrow VSS\_\_State(vss) = VSS\_fr))$
END

Figure 9: Safety invariants.

7 RELATED WORK

Several works have been proposed to combine UML with formal methods. The paper (Zafar and Alhumaidan, 2011) proposed an integration of UML and formal methods for systematic development of computer systems. They developed an approach which link UML to Z notation which defines a relationship among fundamentals of these techniques. Associations, generalization, aggregation relationship and composition of class diagram are chosen at this integration. In fact, an automatic generation of specification from diagrams will be much useful to capture the hidden semantics under the UML notations and to provide a formal mathematical description of the main domain that could be exact and well verified. This work focuses on generating formal specification for functional aspects of information system softwares and does not intended to define the safety aspects of an application. In addition, this work is specified only for Z specification and could not be used in another context or formal methods. The work (Anastasakis et al., 2010) designed an approach that relies on a MDA techniques to transform UML models to Alloy to deal with the analysis of UML models and identify design faults within a specification. This work consists of analyzing UML models captured as class diagrams, enriched with OCL statements, modeling various constraints on the system and using a model driven approach, UML models are automatically transformed to corresponding Alloy representations.This work is specified for Alloy generation in information system that's mean it supports only software parts contrary to our approach, it could not be support both software and hardware parts.

In (Distefano et al., 2011) an evaluation methodology is defined to validate the performance of a UML model. It consists on a mapping from PCM (Performance Context Model) which collect Schedulability, Performance, and Time Specification for the performance annotations into UML models to the petri nets. they provides transformation rules to derive a Petri net from a PCM and to validate their approach they provides an in-depth analysis of a web application for music streaming.This approach is appropriated for functional specification of the UML model specification and could not include safety aspects. In (Snook and Butler, 2006) a strategy is proposed which converts different package, specification and entities of UML to a single B component that support model refinement and could be proved with proof tools. The result is a formally precise variant of UML that can be used for refinement based, object-oriented behavioral modeling. This result is appropriated for software parts while BMethod is destinate only for softwares management, it generate only functional specification and non the safety aspects and it is not destinate for railway systems.

The work (Idani and Ledru, 2015) provides Model Driven Engineering that transforms the functional and secure aspects of the application to formal specification. In fact, it translates a functional UML class diagram and graphical modeling of an access control policy using a UML profile for RBAC inspired from SecureUML model into formal B specifications. Their approach is destinate to the information system because B specification is used only to manipulate software system contrary to railway system which contains both software and hardware parts. In (Siyuan and Hong, 2015) the feasibility to transform models from UML to Event-B is showed. In this paper, a transformation approach was found to map UML activity diagram into Event-B model including the basic mapping of the two types of activity flow. It consists of transforming all the actors will be transformed into the context of Event-B, The basic activities will be transformed into both the context and machine of Event-B and the activity flows will be transformed into the machine. Contrary to our approach which its input is a class diagram this work's input is an activity diagram adding to that this work generate only functional aspects of a domain application without including safety aspects necessary to model railway systems safety aspects.

In this paper, a platform is developed to formally reason about UML class diagram. In fact, this platform operates with the *Papyrus* graphical modeling tool with respect to UML Meta Model as defined by the OMG (Object Management Group). Thereafter, it translates the obtained model into Event-B specifications in order to formally reason about it. In the same context, other tools have been proposed, in the literature, to transform UML class diagram to Event-B like UML-B (Snook and Butler, 2006) and iUML-B (Snook, 2014). These tools are based on their own UML modeler and they provide a UML-like graphical front-end, which makes difficult their correlation with other UML-based tools unlike our platform. Furthermore, these tools do not consider secure policy profile.

8 CONCLUSION

We have introduced a model-based approach where a UML class diagram, with its secure policy profile, can be converted to Event-B models and one can check safety properties to prevent train collisions. This approach can account for both functional and safety aspects through the adoption of model-to-model and model-to-text transformations. A formal verification, using model checking and animation, is also performed to check the consistency of the model. This allows the invariants, typing and safety properties invariants, to studied and they are preserved by all events.

There are several ongoing work to enrich our approach. Our future objective will focus on a transformation of other structure, behavior and interaction diagrams like component diagram, activity diagram and sequence diagram. Another future work is to enrich Railway SecureUML Profile by other secure policies like priority, validity and duration of permissions.

ACKNOWLEDGEMENTS

This work is funded by the NExTRegio project of IRT Railenium. The authors would like to thank SNCF Réseau for its support. We also thank German Vega for his support in the implementation part of this work.

REFERENCES

Abrial, J. (2010a). Modeling in event-b: System and softwareengineer.

Abrial, J.-R. (2010b). *Modeling in Event-B: System and Software Engineering*. Cambridge University Press, New York, NY, USA, 1st edition.

Abrial, J.-R., Butler, M., Hallerstede, S., Hoang, T. S., Mehta, F., and Voisin, L. (2010). Rodin: an open toolset for modelling and reasoning in event-b. *International journal on software tools for technology transfer*, 12(6):447–466.

Ait Wakrime, A., Ben Ayed, R., Collart-Dutilleul, S., Ledru, Y., and Idani, A. (2018). Formalizing railway signaling system ertms/etcs using uml/event-b. In *International Conference on Model and Data Engineering*, pages 321–330. Springer.

Anastasakis, K., Bordbar, B., Georg, G., and Ray, I. (2010). On challenges of model transformation from uml to alloy. *Software & Systems Modeling*, 9(1):69.

Ben Ayed, R., Collart-Dutilleul, S., Bon, P., Idani, A., and Ledru, Y. (2014). B formal validation of ertms/etcs railway operating rules. In *International Conference on Abstract State Machines, Alloy, B, TLA, VDM, and Z*, pages 124–129. Springer.

Beugin, J., Filip, A., Marais, J., and Berbineau, M. (2010). Galileo for railway operations: question about the positioning performances analogy with the rams requirements allocated to safety applications. *European Transport Research Review*, 2(2):93–102.

Cicmancova, S. (2013). safety and risk as part of railway system. Technical report, Transactions of the VSB - Technical university of Ostrava.

Cuppens, F. and Miège, A. (2003). Modelling contexts in the or-bac model. In *Computer Security Applications Conference, 2003. Proceedings. 19th Annual*, pages 416–425. IEEE.

Distefano, S., Scarpa, M., and Puliafito, A. (2011). From uml to petri nets: The pcm-based methodology. *IEEE Transactions on Software Engineering*, 37(1):65–79.

El Kalam, A. A. and Deswarte, Y. (2006). Multi-orbac: A new access control model for distributed, heterogeneous and collaborative systems. In *8th IEEE International Symposium on Systems and Information Security*, page 1.

Ferraiolo, D., Cugini, J., and Kuhn, D. R. (1995). Role-based access control (rbac): Features and motivations. In *Proceedings of 11th annual computer security application conference*, pages 241–48.

Idani, A. (2006). *B/UML: Bridging the gap between B specifications and UML graphical descriptions to ease external validation of formal B developments*. Theses, Université Joseph-Fourier - Grenoble I.

Idani, A. and Ledru, Y. (2015). B for modeling secure information systems - the B4MSecure platform. In *International Conference on Formal Engineering Methods*, pages 312–318. Springer.

Lodderstedt, T., Basin, D., and Doser, J. (2002). Secureuml: A uml-based modeling language for model-driven security. In *International Conference on the Unified Modeling Language*, pages 426–441. Springer.

Musset, J., Juliot, É., Lacrampe, S., Piers, W., Brun, C., Goubet, L., Lussaud, Y., and Allilaire, F. (2006). Acceleo user guide. *See also http://acceleo.org/doc/obeo/en/acceleo-2.6-user-guide. pdf*, 2.

Schön, W., Larraufie, G., Moëns, G., and Poré, J. (2013). Railway signalling and automation. *Work in three volumes. La Vie du Rail*.

Siyuan, H. and Hong, Z. (2015). Towards transformation from uml to event-b. In *IEEE International Conference on Software Quality, Reliability and Security-Companion*, pages 188–189. IEEE.

Snook, C. (2014). iUML-B statemachines. In *Proceedings of the 5th Rodin User and Developer Workshop, 2014*, pages 29–30. Univ. of Southampton.

Snook, C. and Butler, M. (2006). UML-B: Formal modeling and design aided by UML. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 15(1):92–122.

Zafar, N. A. and Alhumaidan, F. (2011). Transformation of class diagrams into formal specification. *International Journal Computer Science and Network Security*, 11(5):289–295.