A UML APPROACH FOR MODELING AND VERIFYING OF RAILWAY SIGNALLING SYSTEMS SPECIFICATIONS

Zaibi Kais, Mohamed Sallak, Walter Schon, Subeer Rangra, Roberto Sacile

10^{th} *International Conference on MOdeling, Optimization and SIMlation - MOSIM14 - November 5-7-2014- Nancy - France "Toward circular Economy"*

A UML approach for modeling and verification of Railway signalling Systems specifications

Z. KAIS, M. SALLAK, W. SCHON, S. RANGRA

Heudiasyc CNRS UMR 7253
Compiegne University of Technology (UTC)
60200 Compiegne - France
mohamed.sallak,walter.schon,zaibi.kais@utc.fr

R. SACILE

University of Genova
Department of Computer science, Bioengineering,
Robotics and Systems engineering (DIBRIS)
16145 Genova - Italy
roberto.sacile@Unige.it

ABSTRACT: *This paper proposes a UML based approach for the modeling and the verification of Railway signalling Systems specifications. Particularly, we consider the European Rail Traffic Management System (ERTMS) and the European Train Control System (ETCS) specifications. First, the architecture of ERTMS/ETCS is described. The validation and verification procedure is also introduced. Then, class, sequences and use case diagrams related to the technical specifications of ERTMS/ETCS are presented. A case study from the technical specification of ERTMS/ETCS which represents the operation of "Establishing a communication session" between ERTMS/ETCS On-board equipment and RBC (Radio Block Center) to initiate a communication session is also proposed. Finally, a formal verification using B method is proposed to show how to verify some safety properties of railway signalling systems and to complete the verification procedure performed using UML.*

KEYWORDS: *Railway signalling systems, ERTMS/ETCS, UML, verification.*

1 Introduction

European Rail Traffic Management System/European Train Control System (ERTMS/ETCS) is a widely implemented railway signalling system in Europe. ERTMS/ETCS is a platform to guarantee the interoperability across different countries and manufacturers by creating a single Europe-wide standard for train control and command systems. It has two components, the first component being ETCS, which is a standard for train control systems, and the second component being the Global System for Mobile communications-Railways (GSM-R), which is an international wireless communications standard for railway communication and applications. ERTMS/ETCS has three levels. ERTMS/ETCS Level 1 and ERTMS/ETCS Level 2 are widely applied in Europe. ERTMS/ETCS Level 3 is currently under development.

The railway standards (EN50126 2000, ERTMS/ETCS 2010b, ERTMS/ETCS 2010a) define the procedure for verification and validation of railway systems. Particularly, the System Requirements Specification (SRS) (ERTMS/ETCS 2010a) developed by the European Railway Agency (ERA) defines the system requirements for the ERTMS/ETCS. This specification often offers multiple solutions on how to implement a specific function. It therefore contains both mandatory and optional requirements. Particularly, the Chapter 3 in (ERTMS/ETCS 2010b) specifies the system principles and specifications of ETCS/ERTMS applied to software used in On-board and trackside subsystems. However, the ERTMS/ETCS System Requirements Specification (ERTMS/ETCS 2010b) and the ERTMS/ETCS Functional Requirements Specification (ERTMS/ETCS 2010a) are written in natural language. This is a major issue when dealing with such systems since, by nature, literal specifications often hold ambiguities as they can be subject to different interpretations.

Several methods were proposed to model railway systems in order to verify their compliance with specifications defined in standard (EN50126 2000, ERTMS/ETCS 2010b, ERTMS/ETCS 2010a). Most of the modelling representations for ETRMS/ETCS were made in B language (Fantechi, Fokkink & Morzenti 2013) or using Petri nets (Barger, Schön & Bouali 2009) which are difficult methods to understand by railway engineers and need some theoretical background. The Unified Modeling Language (UML) is a well known recognized, powerful and leading diagrammatic modeling language. Nowadays, UML is becoming a standard modeling language for the hard-

ware and software Industries. However, few works were proposed to model railway systems using UML. In (Zimmermann & Trowitzsch 2009), the authors described behavioral modeling of systems with UML State Machines and a transformation method into corresponding Stochastic Petri Nets to perform reachability analysis. They consider a part of the communication between trains and RBC. The methods have been implemented as a prototype extension of the TimeNET tool including a specific graphical editor for UML State Machine models. In (Bernardi, Flammini, Marrone, Mazzocca, Merseguer, Nardone & Vittorini 2013), the authors addressed the definition of a Model-Driven approach for the evaluation of RAM (Reliability, Availability, Maintainability) attributes in railway applications to automatically generate formal models. The approach is based on the usage of UML profiles at the conceptual representation level of the system for the automatic generation of formal models. In (Jabri, El koursi, Lemaire & Bourdeaud'huy 2009), the authors presented methods dedicated to the generation of tests scenarios for the validation of ERTMS communication components based on functional requirements, UML models and Petri nets. In (Mecitoglu & Soylemez 2013), UML formalism was employed to design a railway signalling system simulator and a SCADA system. The developed simulator can also help to validate a formal design based on automata. In (Ghazel 2014), the author proposed a mechanizable formalization of a subset of ERTMS/ETCS specifications relative to ETCS modes and transitions based on class diagram model and formal SMV model.

The present work is an attempt of to apply UML methodology to formalize and verify specifications related to the functioning of principal systems used in ERTMS/ETCS. UML class, sequence and activity diagrams related to ERTMS/ETCS will be designed. The originality of this work is that first it proposes a real case study concerning a specification from the UNISIG standard (ERTMS/ETCS 2010b). Secondly, the UML proposed methodology introduces several types of diagrams for modeling ERTMS/ETCS (both On-board and trackside subsystems). Finally, the verification of specifications is based on verification tools used in UML without using other formal methods such as Petri nets or B method. In this work, we argue that ERTMS/ETCS and railway systems in general can benefit from the introduction of a formal modelling approach. Thus, we propose that UML would be a suitable paradigm for modelling ERTMS/ETCS.

2 Architecture and environment of ERTMS/ETCS

The European Railway Traffic Management System (ERTMS) is a major industrial project developed in order to enhance cross-border inter-operability through Europe by creating a single standard for railway signalling. This project was developed by 8 UNIFE (Association of the European Rail Industry) members: Alstom Transport, Ansaldo STS, AZD Praha, Bombardier Transportation, Invensys Rail Group, Mermec, Siemens Mobility, and Thales. It was also supported by the European Union (EU), railway stakeholders and the GSM-R industry. By the end of 2012, more than 62000 km of railway tracks and 7500 vehicles are either already running or being equipped with ERTMS in 38 countries around the world.

ERTMS has two basic components:

- European Train Control System (ETCS) which is an automatic train protection system (ATP) to replace the existing national ATP-systems.
- GSM-R which is a radio system for providing voice and data communication between the track and the train, based on standard GSM using frequencies specifically reserved for rail application.

The architecture and the environment of the ERTMS/ETCS are defined in the the UNISIG SUBSET-026-02 (ERTMS/ETCS 2010b) which a mandatory document related to ERTMS/ETCS requirement specification.

2.1 ERTMS/ETCS Levels

ERTMS/ETCS has three levels. ERTMS/ETCS Level 1 (Fig. 1) is superimposed on the existing signalling system. The transmission of information from the track to the train-borne system is totally dependent on balises which are installed in the track. The driver controls the train according to the lineside signals. In ERTMS/ETCS Level 2 (Fig. 2), the information is transmitted by radio. The authority and track description are displayed directly in the cab for the driver, so lineside signals are no longer needed. Balises are used as positioning beacons to help the train to determine its position via sensors. In ERTMS/ETCS Level 3 (Fig. 3), the train integrity checking is done by the train itself, so track circuits are no longer needed. Balises are used to update position information and transmit position and integrity data back to the interlocking via GSM-R.

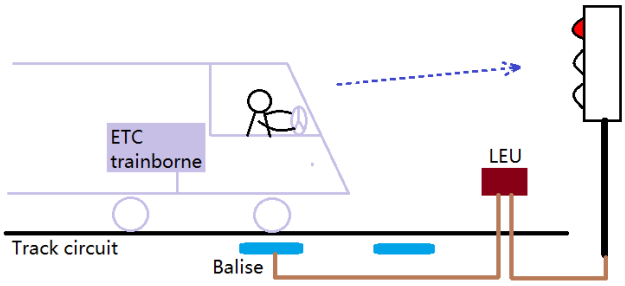


Figure 1: ERTMS/ETCS Level 1

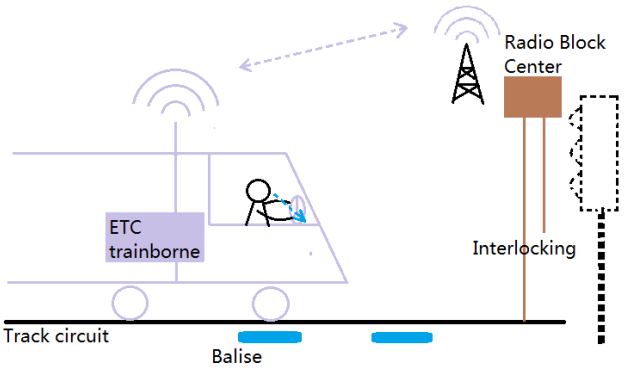


Figure 2: ERTMS/ETCS Level 2

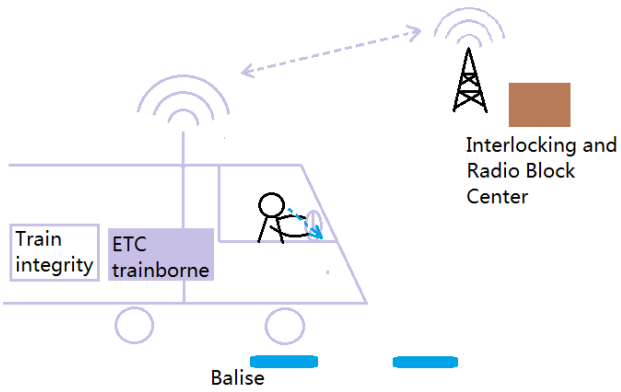


Figure 3: ERTMS/ETCS Level 3

2.2 Architecture of ERTMS/ETCS

Due to the nature of the required functions, the proposed architecture of ERTMS/ETCS system defined in (ERTMS/ETCS 2010b) has two sub-systems:

- On-board sub-system.
- Trackside sub-system.

Figure 4 represents the architecture of the ERTMS/ETCS. The interfaces in brackets are not required for interoperability.

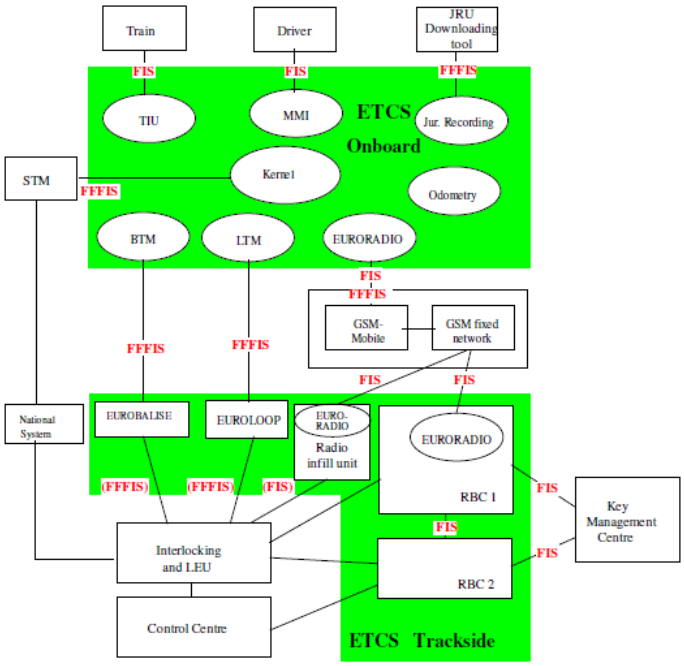


Figure 4: UNISIG ERTMS/ETCS reference architecture

2.2.1 Trackside subsystem

Depending of the ERTMS/ETCS level, the trackside sub-system can be composed of:

- Balises: are electronic beacons or transponders placed between the rails in order to send messages from trackside to the on-board sub-system and are based on existing Eurobalise specifications. Each balise transmits a telegram and the combination of all telegrams defines the message sent by the balise group.
- Lineside electronic unit: generates telegrams to be sent by balises on basis of information received from external trackside systems.
- Radio communication network (GSM-R): is used for the bi-directional exchange of messages between on-board sub-systems and RBC or radio infill units.
- Radio Block Centre (RBC): is a computer-based system that elaborates messages to be sent to the train on basis of information received from external trackside systems and on basis of information exchanged with the on-board sub-systems.
- Euroloop: operates only on ERTMS/ETCS Level 1 lines. It provides signalling information in advance as regard to the next main signal in the train running direction.

- Radio infill unit: operates only on ERTMS/ETCS Level 1 lines. It provides signalling information in advance as regard to the next main signal in the train running direction.

2.2.2 On-board sub-system

Depending of the ERTMS/ETCS level, the on-board sub-system can be composed of:

- The ERTMS/ETCS on-board equipment: is a computer-based system that supervises the movement of the train to which it belongs, on basis of information exchanged with the track-side sub-system. It is composed of:
  - Kernel which comprises the whole Eurocab, the interface equipment with the GSM-R, the data transmission equipment with Eurobalise, Euroloop and Euroradio, and the interface with the lineside signalling systems (interlocking, signals) and other on-board systems (braking systems).
  - BTM (Balise Transmission Module) which is an interface used to receive telegrams from balises and to provide power to balises.
  - RTM (Radio Transmission Module) which provides a bidirectional interface with the Trackside system via a mobile terminal.
  - DMI (Driver Machine Interface) or MMI (Man Machine Interface) which provides a bidirectional interface with the train driver. It displays information and instructions to the driver, and the driver reacts to them.
  - TIU (Train Interface Unit) which provides a bidirectional interface with the train-borne equipment.
  - Odometer which measures train speed and distance since last balise. In order to control train movement, the kernel has to interface with odometer.
- The on-board part of the GSM-R system: The GSM Radio system is neither developed nor standardized within the frame of the UNISIG requirements specifications (ERTMS/ETCS 2010b).
- Specific transmission modules for existing national train control systems.

2.3 ERTMS/ETCS environment

The environment of ERTMS/ETCS system is composed of:

- the train, which will then be considered in the train interface specification;
- the driver, which will then be considered via the driver interface specification;
- other onboard interfaces.
- external trackside systems (interlockings, control centres, etc.), for which no interoperability requirement will be established.

3 RAMS requirements specification for ERTMS

The RAMS (Reliability, Availability, Maintenability and Security) requirements specification for ERTMS are defined in the RAMS requirement specification - Chapter 2 (ERTMS/ETCS 1998) developed by the ERTMS User Group. These RAMS requirement are based on the requirements defined in the CENELEC EN50126 (EN50126 2000) and adapted to the ERTMS/ETCS requirement system specification defined by the UNISIG (ERTMS/ETCS 2010b).

The conformity of ERTMS/ETCS to the RAM requirements is performed in 4 steps:

- The identification of the mission Profile.
- The definition of RAM Requirements.
- The definition of criteria of RAM V&V.
- The definition of requirement for the ERTMS/ETCS RAM programme.

3.1 Mission Profile identification

The mission profile of ERTMS/ETCS introduces the conditions corresponding to the accomplishment of the system mission. The mission of the ERTMS/ETCS is to supervise the movement of trains for each application level, and to ensure their safety. It should be noted that the considered system for these RAMS requirement is composed of the ERTMS/ETCS which equipped the train and the ERTMS/ETCS trackside and lineside equipment encountered during 1 hour of trip in the worst case.

3.2 The definition of RAM Requirements

The ERTMS/ETCS RAMS Requirements Specification (ERTMS/ETCS 1998) provides the RAM requirement for the whole ERTMS/ETCS and the 3 subsystems: Onboard, Trackside and Lineside. Because no experiences are at present recognizable in European Railways at an acceptable experience level, there is no related data on GSM-R in this specification.

3.3 The definition of criteria of RAM V&V (Verification and Validation)

The RAM V&V is based on the evaluation of the RAM Demonstration Test results or, where testing is not applicable for practical or economic reasons, of the documental proof of the fulfilment of RAM targets, in order to establish the compliance with the System RAM Requirements. Particularly, The acceptance criteria are conditioned to the adequacy of the RAM Validation Report, which purpose is to document the success, or the unsuccess, of the Reliability Demonstration Tests or of the documental proof.

3.4 The definition of requirement for the ERTMS/ETCS RAM programme

The ERTMS/ETCS RAM Programme is a set of activities to be performed along the ERTMS/ETCS Lifecycle for ensuring that the RAM Requirements stated for the system are fulfilled at each development phase. The RAM Programme aims to identify the system RAM Requirements and the activities of analysis, verification and demonstration, to be developed by the subjects responsible for performing activities related to one or more ERTMS/ETCS Lifecycle phases, for ensuring the compliance with the above requirements.

4 UML models of ERTMS/ETCS ans formal verification

4.1 UML Diagrams

UML is a visual modeling language. It has been implemented to simplify and consolidate the many object-oriented methods. The object-oriented analysis (OOA) uses object modeling techniques to analyze system requirements. It considers the world as a set of objects with data structures and behaviors. Indeed, the idea that a system can be regarded as a population of interacting objects, each of which is an atomic unit of data and functionality is the foundation of object technology. The object-oriented world is based on the following concepts: objects, classes, inheritance and aggregation. For example:

- An object has a state that is not a set of circumstances describing it.
- A class is a collection of similar objects with the same attributes and the same methods.

For a given class, the created objects are called instances of the class. Each instance will have its own identity. It is possible for objects to be composed of other objects; it is an aggregation or composition relationship. When the destruction of the compound results in the destruction of the components, it is a strong aggregation that is the composition. However, the aggregation is not the only way in which two objects may be linked. An object can be a specialization of another object by inheritance. In addition, UML offers a variety of diagrams to express different views of the system:

- a static view of the system through the class diagram, the object diagram, the component diagram and deployment diagram.
- a dynamic view through the sequence diagram, the state machine diagram, the activity diagram and the collaboration diagram.
- a functional view through the diagram use case.

4.2 Formal verification

Formal methods are increasingly becoming necessary and in some cases mandatory for developing safety critical software of railway signaling systems. Formal specifications allow for a mathematical definition, manipulation and reasoning, facilitating the rigorous testing procedures. In our opinion, whereas UML diagrams allow us to verify at each step of construction diagrams that all the diagrams are free from syntax errors and are coherent, formal methods can be used for the verification of some safety properties of railway signaling systems. Thus, the purpose of our work, in addition of the verification performed when constructing UML diagrams, is the use of B method as another support tool to express and verify some of the safety properties relevant to the railway signaling systems.

B is a formal method introduced by J-R. Abriel (Abrial 1996) that covers the complete life cycle of software development. The main feature of a B development process is that it proves that the final code implements its formal specification. The B notations are based on set theory and generalised substitutions. The B method enables an incremental development process which consists of an abstract specification, followed by some refinement steps. The final refinement corresponds to an implementation. The correctness of the construction is obtained by the verification of the proof obligations (POs) associated to each step of the development. The abstract machine is composed of a set of variables, invariant properties of those variables, and operations. The set of variable values represents the state of the system, and can be modified by operations which must preserve its invariant (see (Abrial 1996), for more details). The POs are generated automatically by software tools such as Atelier B, B4free, B-Toolkit, etc. The check of these POs is performed using the same tools either through an automatic or an interactive proof.

4.3 ERTMS modeling with UML

Modeling ERTMS/ETCS involves several actors. In order to model a process or a scenario, it is necessary to model the behavior of each object involved in a given operating procedure. The identification of stakeholders to define use cases knowing that UML use case determines an action performed by the system and producing an observable result by one of the actor's sequence. An operating procedure of the ERTMS system corresponds to a UML use case. Furthermore, the verification of a component such as the European Vital Computer (EVC) as a reactive component interacting with the external environment requires modeling the behavior of each object communicating with it. The UML sequence diagram identifies the different interactions between objects. The UML class diagram models the static structure of the system by identifying the properties of each object. In the sequel, ERTMS/ETCS systems will be described with UML through their static structures and dynamic behaviors.

4.3.1 UML Static Diagrams

In order to validate and verify the ERTMS/ETCS specifications, we have first to construct a static global view of the ERTMS/ETCS and its environment. The class diagram is the main building block of static diagrams used in UML. Indeed, the considered class diagram, in this section, represents the static structure of ERTMS/ETCS that describes the structure of ERTMS/ETCS by showing the ERTMS/ETCS's classes, their attributes, operations, and the relationships (composition, aggregation and inheritance) among objects. The overall diagram of the whole ERTMS/ETCS was not presented here because of its huge structure. We choose only to represent in Figure 5 a class diagram with principal components.

We provide the classes, attributes and operations of EVC, control center, interlocking system, FPGA, voter, GSM-R card, WAN card, CPU, bus, power supply, and Triple Modular Redundancy (TMR) architecture for the RBC. The top part of the classes contains the name of the class. The middle part contains the attributes of the class. The bottom part gives the methods or operations the class can take or undertake. For example, in the RBC class, the attributes (Id-RBC, Name-RBC, etc.) and operations (Accept-Opening-Session(), Close-Session(), etc.) of RBC are defined. The relationships between the RBC and its components (CPU, TMR, Voter, FPGA, Bus, WAN Card, GSM-R Card, and power supply) are based on compositions whereas the relationships between the RBC, control center, EVC and interlocking are based on associations. Each composition is as a filled diamond shape on the containing class end of the tree of lines that connect contained class to the containing class. Each association is represented as a line. The ends of these associations are adorned with role names and multiplicity. The relationships between the TMR architecture and its components (CPU, Voter, FPGA) are based on aggregations. Each aggregation is represented by a hollow diamond shape on the containing class end of the tree with a single line that connects the contained class to the containing class.

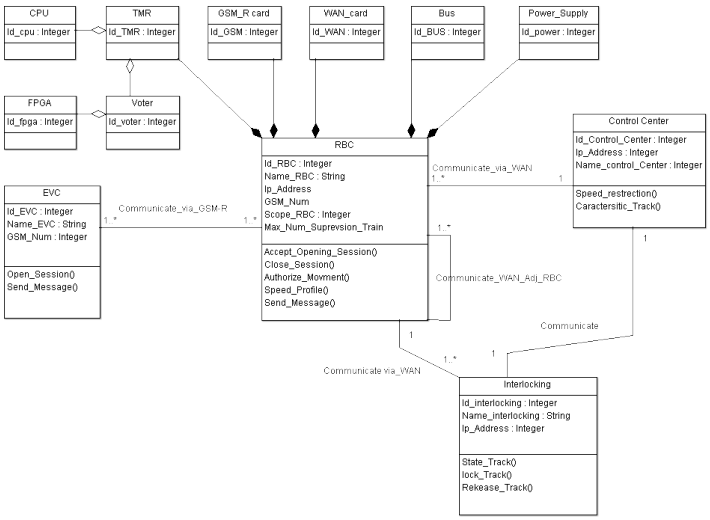


Figure 5: Class diagram of ERTMS architecture

4.3.2 UML Dynamic Diagrams

The use cases are used to represent ERTMS/ETCS at a higher level. They represent the user's interaction with the ERTMS/ETCS subsystems. They can portray the different types of users of ERTMS/ETCS subsystems and the various ways that they interact with the system. In Figure 6, we present a use case of a managing traffic demand. The actors are EVC, Interlocking, Control center and RBC. Then, we present a sequence and an activity diagrams in Figures 7 and 8.

The sequence diagrams show how ERTMS/ETCS subsystems operate with one another and in what order. They are a construct of a Message Sequence Chart and shows ERTMS/ETCS subsystems interactions arranged in time sequence. In Figure 7, we represent the sequence digram of the procedure initiated by the driver in order to receive a MA (Movement Authority) from the control center.

The activity diagrams represent workflows of step-

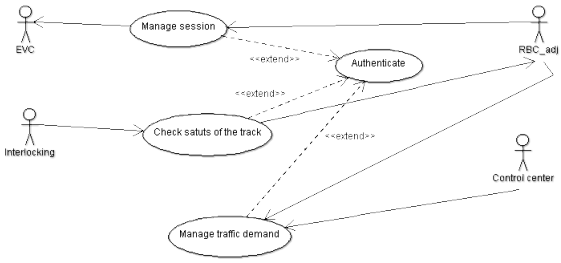


Figure 6: A use case of managing traffic demand

wise activities of traffic permit request and the corresponding actions of ERTMS/ETCS procedures. In Figure 7, we present an activity diagram of traffic permit request. The rounded rectangles represent the following actions: Logon processing, Verification of the itinerary, etc. The diamonds represent decisions about state track. The black circle represents the start (initial state) of the workflow which corresponds to the Opening session received. The encircled black circle represents the end (final state) which corresponds to closing session or reservation of track.

4.3.3 Verification of UML models

All the diagrams of his work were performed using ArgoUML. We then use the modules "Design critics" of ArgoUML. They are simple agents that continuously execute in a background thread of control. They analyze the design of each diagram of ETRTMS/ETCS as we are working and suggest possible improvements. These suggestions range from indications of syntax errors, to reminders to return to parts of the design that need finishing, to style guidelines, to the advice of expert designers.

Many critics offer to automatically improve the design. Critics are controlled so that their suggestions are relevant and timely to the design task at hand, based on information in Argo'UMLs user model. Critics never interrupt the designer, instead they post their suggestions to the designer's "to do" list. This allows us to verify at each step of construction diagrams that all the diagram are free from syntax errors.

5 Case Study: Management of Radio Communication

5.1 Description of the case study

In this case study, we consider the operation of "Establishing a communication session" between ERTMS/ETCS On-board equipment and RBC to initiate a communication session. This specification is presented in the ERTMS/ETCS requirements specification - Chapter 3 (ERTMS/ETCS 2010b). We aim to model and formalize the above specification using UML models. It should be noted that the radio In-fill Unit shall never initiate a communication session. Furthermore, only communication sessions between an ERTMS/ETCS On-board equipment and a trackside equipment (RBC or Radio In-fill Unit) are considered here. The on-board shall establish a communication session:

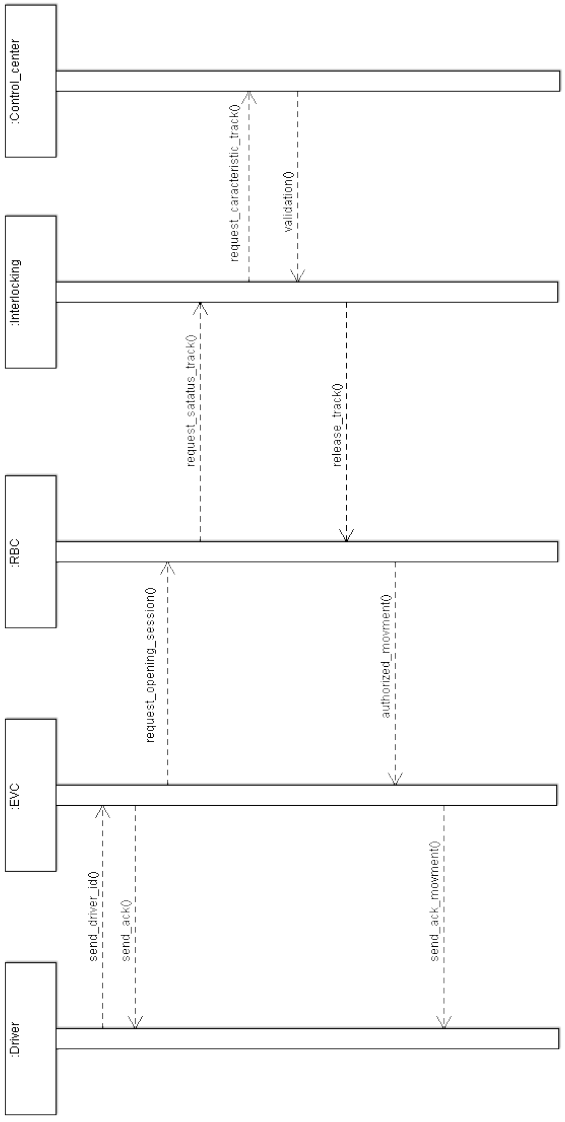


Figure 7: A sequence diagram of managing traffic demand

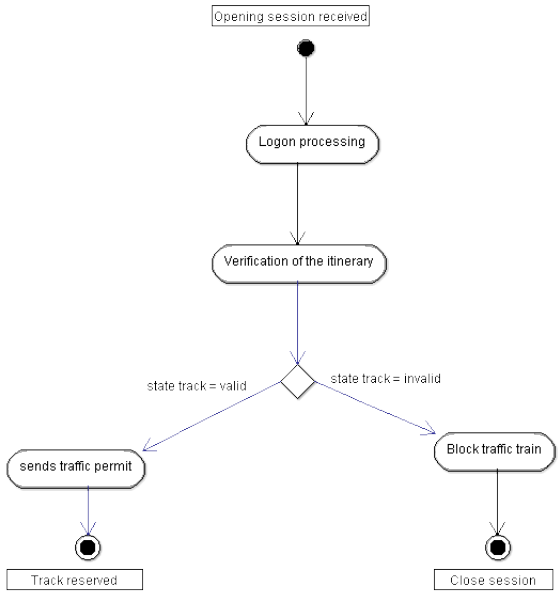


Figure 8: Activity diagram of Traffic permit request

- At Start of Mission only if level 2 or 3.
- If a mode change, not considered as an End of Mission, has to be reported to the RBC only if level 2 or 3.
- If the driver has manually changed the level to 2 or 3.
- When a Start of Mission procedure, during which no communication session could be established, is completed in level 2 or 3.

The order to contact an RBC shall include:

- The identity of RBC.
- The telephone number of RBC.
- The action to be performed (establish/terminate the session).

If the order to establish a communication session with an RBC is received and accepted by ERTMS/ETCS On-board equipment already in session with another RBC, the existing communication session shall be terminated and the new one shall be established. The order to contact an Accepting RBC shall be part of the RBC transition order and shall include:

- The identity of the Accepting RBC.
- The telephone number of the Accepting RBC.
- Whether this applies also to Sleeping unit.

The order to contact a Radio In-fill Unit shall include:

- The identity of Radio In-fill Unit.
- The telephone number of Radio In-fill Unit.
- The action to be performed (establish/terminate the session).

If the establishment of a communication session is initiated by On-board, it shall be performed according to the following steps:

- The On-board shall request the set-up of a safe radio connection with the trackside. If this request is part of an on-going Start of Mission procedure, it shall be repeated until successful or a defined number of times.
- As soon as the safe radio connection is set-up, the On-board shall send the message Initiation of communication session to the trackside.
- When the on-board receives the system version it shall consider the communication session established and:
  - If one of its supported system versions is compatible with the one sent by trackside, it shall send a session established report, including its telephone numbers, to the trackside.
  - If none of its supported system versions is compatible with the one sent by trackside, it shall send a version independent message indicating ”No compatible version supported”. It shall inform the driver and then shall terminate the communication session.
- When the trackside receives the session established report or the information that no compatible system version is supported by the on-board, it shall consider the communication session established.

If the establishment of a communication session is initiated by RBC, it shall be performed according to the following steps:

- The trackside shall request the set-up of a safe radio connection with the on-board.
- As soon as the safe radio connection is set-up, the trackside shall send the message Initiation of communication session to On-board.

- When the on-board receives the information, it shall consider the communication session established and send a session established report to the trackside.
- When the trackside receives the session established report, it shall consider the communication session established.
- In case the RBC is the initiator, the first message from RBC to On-board shall have the timestamp set to "unknown";
- In the case the RBC is the initiator, there is no need to verify the compatibility of the system versions and for the on-board to send its telephone numbers, because the on-board is obviously already known to RBC.
- An order to contact RBC may contain a special value for the RBC identity indicating that the on-board shall contact the last known RBC (i.e., using the stored RBC ID/phone number, if any); the phone number indicated in the order shall be ignored by the on-board equipment.
- If there is no RBC ID/ phone number stored on-board, the order to contact RBC shall be ignored.

5.2 UML models

The class diagram shown in Figure 9 represents RBC, Radio, On-board subsystems, and their main attributes and methods which are required to establish a communication between RBC and On-board via Radio. So each subsystem should have a unique reference (Id-Subsystem) to be identified and to avoid ambiguity; they also have their own methods and attributes. The class association "Connection" which is represented by the dotted line is a part of an association relationship between RBC, Radio, and On-board. It provides additional information (Id-connection and Data-connection) about the relationship between RBC, Radio, and On-board in order to establish connection. Particularly, we show that the connection can't be established if one of the three subsystems is missing.

As shown in the sequence diagram of Figure 10, since RBC already knows the Identifier of On-board, the communication has been established for RBC which sends a timestamp "unknown" to EVC located in On-board. Then, EVC has to search RBC phone number in its memory card and sends a request to Control center to get authorization to move, if On-board didn't find the phone number, the communication session is not established because EVC is not autonomous in the situation of contact a new RBC. In this case, On-board needs to receive an order to send a request for recovering RBC phone number. Then,

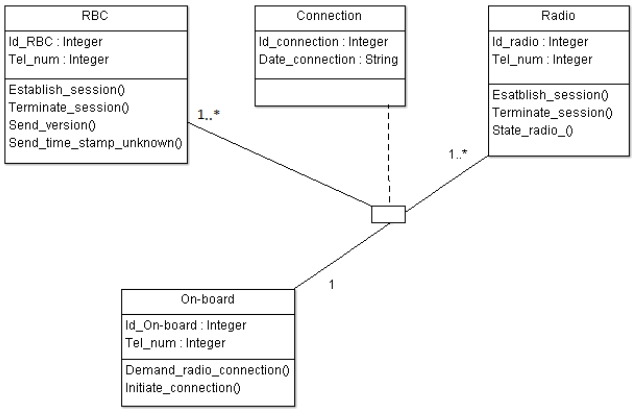


Figure 9: Class diagram of "Establishing a communication session"

RBC verifies the itinerary and validate or not the request in the form of movement authorized.

To establish a communication initiated by On-board with Trackside (cf. Figure 11), On-board sends first a message. Then, RBC transmit his version with RBC phone number. In case of compatible version, the communication session is established. Otherwise, On-board sends a message to inform RBC that "No compatible version supported". When session is validated, On-board sends a request to get authorization to move and RBC verifies the itinerary and validate or not the request in the form of movement authorized.

UML models allow one to detect the errors in the procedure through class diagrams and sequence diagrams. As shown in this case study, if we make an error in the identifier of on-board. A failure in the connection procedure will occur because On-board cannot initiate communication and each connection is defined by a date and association identifier RBC Radio and on-board. In the case, when RBC initiates communication, On-board searches in its memory the number of RBC and if it cannot find it or gets a wrong Identifier number, it cannot contact RBC and returns a report to the Control Center. Then, if a message cannot be sent, the connection will be not established. Different kinds of verifications were possible on the proposed UML diagrams of the case study: for instance, the verification that a functionality specified in the sequence diagram of "Establishing a communication session" is present in class diagrams, or on the contrary that no functionality not required by all the specification is present in the class diagrams. In practice, we discovered some methods related to never activated functions or useless statements, which were inherited from an early specification in the class diagram. Moreover, at a glance verifications on UML models were straightforward: a sequence diagram should contain only the processes that are involved in that function, as specified by high

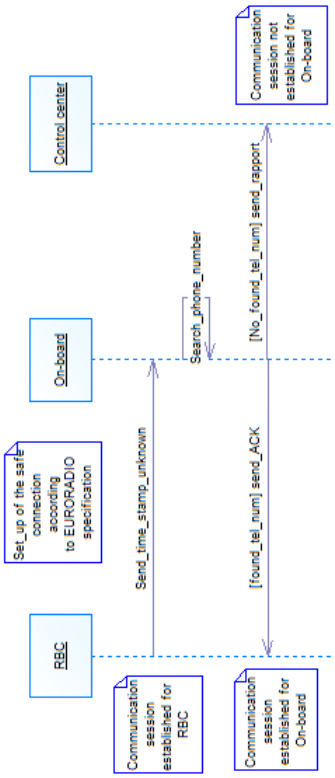


Figure 10: Sequence diagram of "Establishing a communication session" initiated by RBC

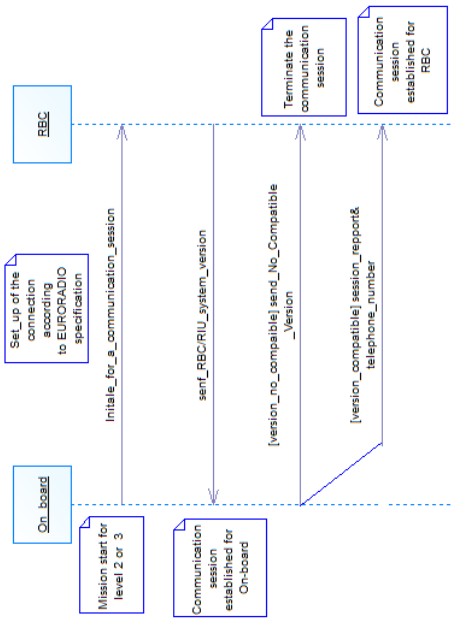


Figure 11: Sequence diagram of "Establishing a communication session" initiated by the On-borad

level requirements. Such kind of analyses was performed informally, exploiting the know-how and skill of railway experts.

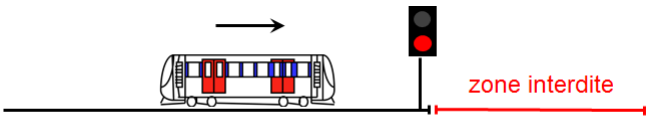


Figure 12: Illustration of example 1

6 Formal verification of an ERTMS/ETCS level 1

In this section, we consider two examples of specification of an ERTMS/ETCS level 1 to explain how to use the set theory in order to convert some safety informal railway specifications into formal ones. Then, we construct the abstract machine, the refinement and the implementation of an example of a safety railway specification using B method. Finally, we generate the POs and prove them using the Atelier B tool. Note that the formal specifications are based on the notation defined in (Abrial 1996).

Example 1

Let us consider the following informal railway safety specification for an ERTMS/ETCS level 1 (cf. Figure 12):

If the train enters the forbidden zone called "zone interdite", then emergency braking called "Frein urgence" must be triggered.

The formal specification using the set theory is:

$$\begin{array}{l} Position\_train \subseteq POSITIONS \wedge \\ Zone\_interdite \subseteq POSITIONS \wedge \\ Position\_train \cap Zone\_interdite \neq \emptyset \Longrightarrow \\ Frein\_urgence = TRUE \end{array}$$

Example 2

Let us consider another following informal railway safety specification for an ERTMS/ETCS level 1 (cf. Figure 13):

The railway line is divided into elementary fixed areas called block sections. Each block section can be occupied by at most one train.

The formal specification using the set theory is:

$$Est\_occupe\_par \in CANTONS \ +\!\!-\!\!> \ TRAINS$$

The notation $+\!\!-\!\!>$ means that the application Est_occupe_par is a partial function from the set

Figure 13: Illustration of example 2

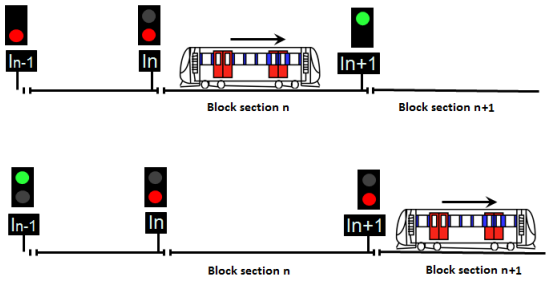


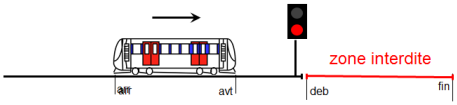
Figure 14: Detailed illustration of example 2

"CANTONS" to the set "TRAINS". In other words, an element of "CANTONS" may not have an image, and all the elements of "CANTONS" have at most one image.

The safety properties defined in examples 1 and 2, which must be always respected are called "INVARIANTS". Indeed, safety properties can always be translated in the form of "INVARIANTS" expressed as logical statements called predicates. The first step is the formal translation of the requirements from informal specifications. This step is crucial because any error or omission at this stage is clearly not covered by the following formal process. However any property expressed correctly enters a process to demonstrate mathematically that the whole development respect it, making the initial specification all the more crucial. The state of the system is described by "VARIABLES" which are subject to change. The "VARIABLES" involve also optionally constants (CONSTANT) and sets (SETS) which may remain at the abstract formal specification. The changes are related to the services provided by the system, formalized by "OPERATIONS". "OPERATIONS" are also described in a formal language called "SUBSTITUTIONS". In the example 2, in which we have added block sections n and $n+1$ (cf. the detailed Figure 14), we consider the following operation:

$$
\begin{array}{|l|}
\hline
OPERATIONS \\
Avance\_train = \\
n := n+1 || In+1 := ROUGE || In-1 := VERT \\
\hline
\end{array}
$$

Where || design the parallel "SUBSTITUTION". The process of proof is to show that any "OPERATION" respects the "INVARIANT". That is what we call proof obligations (POs). We have to demonstrate that each PO is true like the mathematical proof




```
MACHINE ZI
SETS POS
VARIABLES position_train, zone_interdite, freinage_urgence
INVARIANT position_train <: POS & zone_interdite <: POS & freinage_urgence
: BOOL &
(zone_interdite /\ position_train /= {} => freinage_urgence = TRUE)
INITIALISATION position_train, zone_interdite, freinage_urgence
:(position_train <: POS & zone_interdite <: POS & freinage_urgence : BOOL &
(zone_interdite /\ position_train /={} => freinage_urgence=TRUE))
END
```

```
REFINEMENT ZI_r
REFINES ZI
ABSTRACT_VARIABLES position_train , zone_interdite , freinage_urgence
INITIALISATION
position_train, zone_interdite :(position_train<:POS & zone_interdite<:POS)
|| freinage_urgence := bool(position_train /\ zone_interdite /= {})
END
```

```
IMPLEMENTATION ZI_i
REFINES ZI_r1
VALUES POS=0..100
CONCRETE_VARIABLES arr, avt, deb, fin, freinage_urgence
INVARIANT arr : NAT & avt : NAT & deb : NAT & fin : NAT &
          position_train=arr..avt & zone_interdite=deb..fin
INITIALISATION arr:=0; avt:=10; deb:=50; fin:=80;
        IF deb>fin THEN freinage_urgence:=FALSE
        ELSIF avt<deb THEN freinage_urgence:=FALSE
        ELSIF arr>fin THEN freinage_urgence:=FALSE
        ELSE freinage_urgence:=TRUE
        END
END
```

Figure 15: Abstract machine, refinement and implementation of example 1

of a theorem. A PO mathematically translates the following statement:

"If the INVARIANT is TRUE before the OPERATION, it will be TRUE after the OPERATION".

We show in Figure 15, the B abstract machine, the refinement and the implementation of the example 1 written using the ASCII notation of B (AtelierB 2014). Note that the refinement machine adds the explicit definition of the variable *Frein urgence*. A graphical visualisation of the fourth generated POs is shown in Figures 16 and 17. As we can see, all the four POs of the implementation machine ZI_i were proven. Thus, we have proven that the final code, which will be generated automatically from the implementation machine ZI_i using the Atelier B tool, implements the formal specification of example 1.

7 Conclusion

In this paper, we have shown how to construct UML models of ERTMS/ETCS in order to formalize and verify functional and systems requirements. We have also explain how to make formal verification using B method in order to proof safety properties of rail-

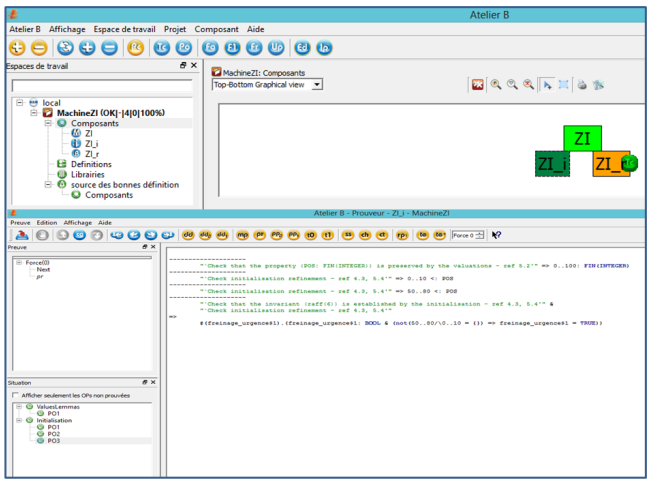


Figure 16: AtelierB user interface showing the generated and proven POs of example 1

```
----------
"`Check that the property (POS: FIN(INTEGER)) is preserved by the valuations - ref
5.2'" => 0..100: FIN(INTEGER)
----------
"`Check initialisation refinement - ref 4.3, 5.4'" => 0..10 <: POS
----------
"`Check initialisation refinement - ref 4.3, 5.4'" => 50..80 <: POS
----------
"`Check that the invariant (raff(6)) is established by the initialisation - ref 4.3, 5.4'" &
"`Check initialisation refinement - ref 4.3, 5.4'"
=>
#(freinage_urgence$1).(freinage_urgence$1: BOOL & (not(50..80/\0..10 = {}) =>
freinage_urgence$1 = TRUE))
```

Figure 17: Detailed descriptions of POs of example 1

way signalling systems and to complete the verification procedure performed using UML. The case study presented here is part of a real specification of ERTMS/ETCS where the approach was successfully applied. The proposed method enables techniques that can be used by manufacturers to formalize and check automatically the conformance of their equipment (on-board, track-side) to their functional and systems requirements.
Our futures work will focus on directly converting UML models to formal languages that can ensure formal verification of safety specifications.

Acknowledgment

This work was carried out and funded by the French National Research Agency, through the project ANR-13-JS03-0007 RECIF.

References

Abrial, J.-R. (1996). *The B-Book: Assigning programs to meanings*, Cambridge University Press.

AtelierB (2014). Ascii notation for b. http://www.atelierb.eu/ressources/symboles1.8.6.uk.pdf, *Technical report*.

Barger, P., Schön, W. & Bouali, M. (2009). A study of railway ERTMS safety with Colored Petri Nets, *European Safety and Reliability Conference, ESREL'09, Prague, Czech Republic*.

Bernardi, S., Flammini, F., Marrone, S., Mazzocca, N., Merseguer, J., Nardone, R. & Vittorini, V. (2013). Enabling the usage of UML in the verification of railway systems: The DAM-rail approach, *Reliability Engineering and System Safety* **120**: 112 – 126.

EN50126 (2000). Railway Applications - The Specification and Demonstration of Reliability, Availability, maintainability and Safety (RAMS), *Technical report*, CENELEC.

ERTMS/ETCS (1998). RAMS Requirements Specification, Reference EEIG : 96S126.

ERTMS/ETCS (2010a). ERA, Functional Requirements Specification, Ref: ERA/ERTMS/003204, *Technical report*.

ERTMS/ETCS (2010b). ERA, System Requirements Specification, UNISIG SUBSET - 026, Ref: Index004-SUBSET-026, *Technical report*.

Fantechi, A., Fokkink, W. & Morzenti, A. (2013). *Formal Methods for Industrial Critical Systems: A Survey of Applications*, Wiley-IEEE, chapter Some trends in formal methods applications to railway signaling, pp. 63–84.

Ghazel, M. (2014). Formalizing a subset of ERTMS/ETCS specifications for verification purposes, *Transportation Research Part C: Emerging Technologies* **42**: 60 – 75.

Jabri, S., El koursi, E., Lemaire, E. & Bourdeaud'huy, T. (2009). Modelling of the European Rail Traffic Management System (ERTMS) for Checking Objectives, *12th IFAC Symposium on Control in Transportation Systems, US*.

Mecitoglu, F. & Soylemez, M. T. (2013). A UML Modelling Approach for a Railway Signalization System Simulator and SCADA System, *1st IFAC Workshop on Advances in Control and Automation Theory for Transportation Applications*.

Zimmermann, A. & Trowitzsch, J. (2009). Reliability Evaluation of Distributed Embedded Systems With UML State Charts and Rare Event Simulation, *Dagstuhl-Workshop MBEES: Modellbasierte Entwicklung eingebetteter Systeme V, Schloss Dagstuhl, Germany, pp.128-139*.