

# Scenario-based modeling and verification of system requirement specification for the European Train Control System

W. Tang, B. Ning, T. Xu & L. Zhao

*State Key Laboratory of Rail Traffic Control and Safety,  
Beijing Jiaotong University, China*

## Abstract

The high quality System Requirement Specification (SRS) is at the heart of the design and development of the European Train Control System level 2 (ETCS L2) with high safety and efficiency. However, the SRS, written in natural language with a shortage of rigorous mathematic foundation, makes it difficult to meet the high quality attributes of SRS, such as correctness, completeness and consistency. In order to tackle the above problems, the integration of a scenario-based model with a formal method, which is recommended to model and verify safety critical system (e.g., train control system), is proposed to improve the quality of the SRS for ETCS L2. First, the relevant operational scenarios are extracted from the SRS, then the corresponding UML sequence diagrams are constructed and finally the sequence diagrams are verified by the formal analysis tool (i.e., NuSMV) through a series transformation rules from UML sequences to NuSMV. The output analysis results facilitate improvement of the SRS qualities. Within the above modeling and verification process, the key mapping relationship is presented to ensure the consistency and traceability between the UML sequence model and the NuSMV specification.

*Keywords: European train control system, modeling and verification, scenario, sequence diagram, model checking.*

## 1 Introduction

The European Train Control System (ETCS) is a signaling, control and train protection system currently used by European railways. The ETCS can be configured to operate in a certain level. In this paper we take level 2 (ETCS L2)



as a research background, which is a radio based train control system used as an overlay on an underlying signaling system. It is not easy to analyze the ETCS because the system is very complex and is required to satisfy different kinds of critical properties. The System Requirements Specifications (SRS) of the ETCS L2 are written in natural language with a shortage of rigorous mathematic foundation, which makes it difficult to meet the high quality attributes of SRS, such as correctness, completeness and consistency. In order to tackle the above problems, the integration of a scenario-based model with a formal method, which is recommended to model and verify safety critical system (e.g., a train control system), is proposed to improve the quality of the specifications.

At present, scenario-based approaches have become a popular practice in some areas (such as business, industry, biology, etc. (De Backer *et al.* [1], Theißen *et al.* [2], Fisher *et al.* [3], Kahyaoğlu-Koračina *et al.* [4])), especially in safety critical areas. It is a simple and intuitive way to describe a system by listing various examples or scenarios of its intended behavior. The scenario-based approaches mitigate the processes of development, testing, validation and verification of the safety critical systems. Many notations are used to describe the scenarios; Liang *et al.* [5] and Amyot and Eberlein [6] listed and compared dozens of scenario notations in their works. Generally, the most popular notations are Message Sequence Charts (MSCs), Use Cases, Statechart Diagrams, Life Sequence Charts (LSCs) and Petri nets. As for the application of the scenario-based approaches, the predecessors have done a lot of work. Palshikar and Bhaduri [7] presented HMSC templates to specify interaction scenarios and discussed efficient algorithms for verifying them. Lee *et al.* [8] used time Petri-nets to verify the acquired scenarios. Fahland [9] presented a novel, operational, formal model for scenario-based modeling with Petri nets. Autili *et al.* [10] proposed the Property Sequence Chart (PSC) notation to specify temporal properties, and defined the precise semantics of PSC. Mäkinen and Systä [11] discussed the MAS algorithm, which is an interactive algorithm that synthesizes UML statechart diagrams from sequence diagrams, and shows its practical implementation in modeling. However, the consistency between models and scenarios still cannot be guaranteed, and the properties abstraction form specifications are not specified. Our approach is to tackle the two problems.

Having surveyed all of these scenario notations, considering the efficiency and expressiveness, we adopted the UML sequence diagram to model the operational scenarios of the ETCS L2 SRS. It is a semi-formal modeling method expressed by graphic notation. Compared to MSCs, the sequence diagram is less expressive because of the lack of explication as MSCs with Z.120 ITU-T [12], but it is powerful enough to model the operational scenarios. We do not need the complex algorithms of MSCs, and just reserve the benefits of describing the interactions, the time orders and single traces. As for LSC and PSC, they are extensions of MSC for specified usages. Additionally, all of these extended graphical notations are subsets of UML 2.0 interaction sequence diagrams, but they are not as general as the sequence diagram. So we use the sequence diagram to model for its universality and its powerful expressiveness in describing

interactions. Moreover, the correctness and consistency are guaranteed during the modeling process.

There are different attempts for verifying the systems rigorously and completely, such as model checking or theorem proving. Theorem proving requires a solid mathematical background for the verification teams to reason and proof, so it is impractical for most practitioners. Conversely, model checking can be fully automated and has been successfully used in the verification of practical applications. The model checker SPIN (Holzmann [13]) is well known in verifying the properties expressed in Linear Temporal Logic (LTL) formulas. Another model checker is NuSMV (Cimatti *et al.* [14]) (a modified version of the original SMV), which has useful features like branching time logic (CTL) for expressing properties. In this paper, we adopted CTL to describe the properties, so the NuSMV model checker is employed. If the properties are not satisfied, NuSMV may give counterexamples to demonstrate and trace the errors. The analysis of the verification is on the basis of the model checking results. The traceability and completeness can be assured in this stage.

The remainder of the paper is structured as follows. Section 2 introduces the conception of scenario and summarizes the operation scenarios of the ETCS L2 SRS. We provide a typical operational scenario to demonstrate the process of scenario extraction. Section 3 expatiates on the steps of scenario-based modeling and illustrates the graphical notation of the sequence diagram. The mapping relationships between scenarios and models are interpreted by a case study of the shunting scenario in ETCS L2. Section 4 is dedicated to verifying the properties abstracted from the ETCS L2 SRS using the NuSMV model checker, and briefly analyzes the results of the verification. Finally, we present the conclusion and future work in Section 5.

## 2 Scenario extraction

### 2.1 Scenario definition

Since the 1980s, scenarios have been used in system development, and many definitions are given by practitioners. For example, in Weidenhaupt *et al.* [15], where the authors make an investigation to the current practice of scenarios in system development, it states that scenarios present a possible way to use a system to accomplish some desired function. Rolland *et al.* [16] propose a scenario classification framework to consider scenarios along four different views. Others focus on the behaviors, functions or interactions expressiveness, and divide scenarios into global and local, positive and negative, etc. (Uchitel *et al.* [17] Bai *et al.* [18]). According to their works, as well as relating to the characteristic of the ETCS L2, we define scenarios as follows.

**Definition (Scenario):** the scenario is a sum of the interactions and behaviors of the system and its intra components, which are exhibited in operations.

The mediums of scenario description are numerous, including natural language text, graphics, images, videos or designed prototypes. Generally, the methods that describe scenarios in specifications are text and graphics, while

those that describe software testing are software prototypes. Sometimes, photos and videos may be used to interpret the scenarios of requirements for better communication between developers and users.

Furthermore, the modeling language for scenarios may be provided in semi-formal or formal notations. However, descriptions can also be informal as they are expressed using concrete terms of the reality.

## 2.2 ETCS L2 operational scenarios overview

The ETCS L2 SRS define procedures, entities and states that compose the operational scenarios. The procedures focus on the required changes in status and mode of the described ETCS entities. There are 12 procedures that are necessary for interoperability within the scope of ETCS Level 2. According to these procedures, we elicit typical operational scenarios as follows.

- Start of Mission
- End of Mission
- Shunting Initiated by Driver
- Entry in Shunting with Order from Trackside
- Override EoA
- On-Sight
- Level Transitions
- Train Trip
- Change of Train Orientation
- Train Reversing
- Joining/Splitting
- RBC/RBC Handover

All of the operational scenarios are typical and important. Considering complexity and entity inclusiveness, we take the “Shunting Initiated by Driver” scenario as case study, which is one of the basic operations of train movement.

## 2.3 Typical operational scenario extraction

Informal descriptions of the procedures are presented in natural language text and graphics. We remove the specialized terminologies and narrate the procedures in common terms, so that nonexperts can understand. For example, the “Shunting Initiated by Driver” scenario is narrated as follows.

The driver selects “Shunting” while the train is at standstill and the ETCS on-board equipment is in one of the following modes: FS, OS, SR, UN or SB. The on-board equipment checks the current ETCS level of operation. If it is level 1, the transition to SH mode shall be made. Then the “End of Mission” procedure is executed on entering the SH mode. While the ETCS level is 2, the ETCS on-board equipment sends the “Request for Shunting” message to the RBC, together with a position report (with special value “position unknown” if the position is not known). The on-board equipment indicates to the driver, through the DMI, that a SH permission request to the RBC is pending, and awaits the SH permission. When the “Request for Shunting” message is received, the RBC will send SH permission to the on-board equipment if the schedule permits, otherwise the “SH refused” message will be sent. If the on-board equipment receives the SH permission from the RBC, transition to the SH mode shall be made and the mode change shall be reported to the RBC. Then the “End of Mission” procedure is executed on entering the SH mode. While the on-board equipment receives the

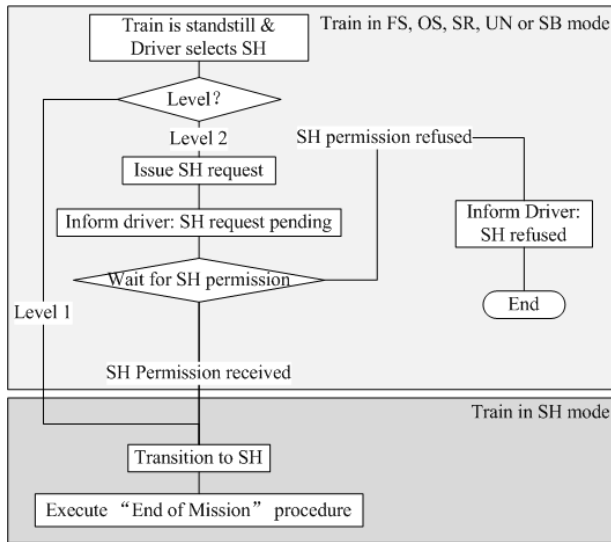


Figure 1: Graphic description for the shunting scenario.

information from the RBC “SH refused”, an indication shall be given to the driver through the DMI, showing that SH permission was refused by the RBC.

The narrative text description is detailed, but very redundant and not intuitive enough, so have given the graphic of the scenario in Fig. 1 as supplementary information.

### 3 Scenario-based modeling

#### 3.1 Mapping rules

It is very important to guarantee the correct models and ensure consistency between scenarios and models. This is the key point of our work. Therefore, we construct a one-to-one mapping between scenarios and UML sequence diagrams. The rules of mapping are enumerated as follows.

- Rule 1: An entity of the scenarios is a participant of the sequence diagram.
- Rule 2: An event occurrence of the scenarios is a message (event) of the sequence diagram.
- Rule 3: Operations of the scenarios are messages (function) of the sequence diagram.
- Rule 4: The time order of cascading events in the scenarios is interpreted by the life line of the sequence diagram.
- Rule 5: Selections in the scenarios are translated by the ‘alt’ operator of the sequence diagram.

- Rule 6: Conditions in the scenarios are presented by the ‘opt’ operator of the sequence diagram.
- Rule 7: Concurrent scenarios events are depicted in parallel combined fragments with operator ‘par’ of the sequence diagram.

### 3.2 Modeling

In the modeling stage, we are concerned with the behaviors and interactions of the system. UML sequence diagrams are behavioral diagrams with the simple, expressive, intuitive, graphical and standardized notations used to specify interactions among system entities in many different situations (Lima *et al.* [19]). Our scenario-based model is presented in semi-formal notation. According to the semantics of UML 2.0 interaction diagrams, efficient algorithms may be applied as well. However, it is beyond the scope of this paper, so we do not discuss this here.

The scenario-based modeling process contains all the steps from scenario extraction to model construction. We will explain the steps as clearly as we can by modeling the “Shunting Initiated by Driver” operational scenario. To ensure the consistency between the model and the system, we employ the mapping rules to translate the operational scenario into a standard UML sequence diagram. The concrete steps are listed as follows:

- a) *Scenario extraction.* Extract operational scenarios from the specifications, and depict them in an understandable way using the medium mentioned in section 2.1;
- b) *Modeling elements collection.* Sort out the entities and their behaviors, the events, time order and other elements we are concerned with from the operational scenarios;
- c) *Mapping.* Map the basic elements of the scenarios to the sequence diagrams using the mapping rules presented in section 3.1. Construct a relationship table for modeling needs.
- d) *Model construction.* Construct the sequence diagram model according to the mapping relationship. Check the consistency between the scenarios and the sequence diagrams.

Following the listed steps, we construct the sequence diagram model of the “Shunting Initiated by Driver” scenario. The first step has already been done in section 2.3. In the rest of the steps, we sort out the modeling elements as follows:

- Entities: On-board, RBC, Driver
- Event Occurrences: (i). Select “Shunting”; (ii). Send/receive the “Request or Shunting” message; (iii). Indicate that a SH permission request to RBC is pending, and await SH permission; (iv). Send/receive the SH permission; (v). Report the mode change; (vi). Send/receive/indicate the “SH refused” message
- Operations: Transition to SH mode
- Selections: Level 1 or Level 2, received the SH permission or not
- Conditions: Train is at standstill and the Driver selects SH

The Conditions are considered as default satisfied. The mapping relationship between the scenario and sequence diagrams is listed in Table 1.

Table 1: The mapping relationship between the scenario and sequence diagrams.

Scenario/Sequence Diagram		
Entities/Participants	Event Occurrences or operations	Messages
On-board RBC Driver	select “Shunting”	Shunting
	Transition to SH mode	EnterSHMode
	Send/receive the “Request or Shunting” message	RequestForShunting
	Indicate that SH permission request to RBC in pending, and await SH permission	WaitForSHPermission
	Send/receive the SH permission	SHPermission
	Report the mode change	ModeChange
	Send/receive/indicate the “SH refused” message	SHRefused

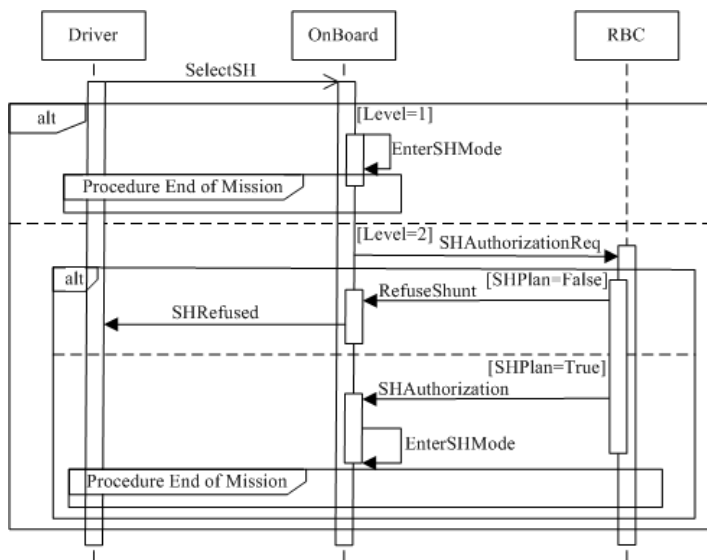


Figure 2: The sequence diagram of the “Shunting Initiated by Driver” scenario.

Table 1 lists the mapping relationships of basic elements, but selections and conditions are not enumerated. According to the mapping rules, selections are corresponding to the ‘alt’ operator in the sequence diagram. Since the conditions are default satisfied, we omit them. Finally, the sequence diagram is constructed as Fig. 2.

In Fig. 2, in the alternative combined fragments, the ‘alt’ operator is represented as an if condition; only the true condition branch will be executed. The fragment named “*End of Mission*” Procedure quotes other sequence diagrams that may model the “End of mission” operational scenario. It does not affect our approach elaboration, so its detail will not be discussed here.

## 4 Verification

The formal verification of the specifications aims at improving the quality of the specifications, checking whether the SRS meets the design intent or not. For instance, verifying that desired behaviors have not been ruled out by over-constraining the requirements, while the undesired behaviors have been ruled out by under-constraining the requirements.

### 4.1 Properties

As mentioned in section 1, ETCS SRS is requested to be high quality, which embodies the properties of the 3Cs (Correctness, Completeness and Consistency). We specify these properties concretely and classify them as *Domain-independent features* and *Domain-dependent features*. The former covers all the basic characteristics of safety-critical system requirements, while the latter presents the special properties of railway domain system requirements.

#### 4.1.1 Domain-independent features

Domain-independent features do not require any domain knowledge, such as logic consistency and realizability. They are listed as follows:

- 1) *Reachability*, to check the completeness of states. It can be used to deduce the sequential relationship of system operation.
- 2) *Transition*, to check the transitivity between states.
- 3) *Deadlock*, describes a state wherein a process is waiting for some event that will never happen.
- 4) *Mutual exclusion*, multiple situations should not occur simultaneously in a system.
- 5) *Definitiveness*, to check the synchronization of the states.

#### 4.1.2 Domain-dependent features

The domain-dependent features are the safety and non-safety functions or properties related to the specific areas. The railway domain-dependent features can be divided into safety properties and liveness properties. Meanwhile, scenario compatibility is added in for the purpose of enhancing the consistency of the ETCS SRS. The definition and description of these properties are listed as follows:

- 6) *Safety*. This means that nothing bad can occur. The system should be safe and reliable.
- 7) *Liveness*. This means that the required events will occur eventually. The functions of the ETCS should be satisfied.



- 8) *Scenario compatibility*. This check aims at verifying whether a scenario is possible, or if the scenarios are consistent or not.

## 4.2 Translations

From modeling to verification, we translate the sequence diagram into a program that may import to the model checker NuSMV. The translation rules can be found by referring to Tanuan [20]. The specified properties are stated by CTL notations. Generally, CTL notations contain path quantifiers (A, E) and temporal operators (X, F, G, U). For the path quantifiers, 'A' means 'For all paths' and 'E' means 'A path exists'. For temporal operators, 'X' means 'Next time', 'F' means 'Eventually (or in the future)', 'G' means 'Globally (or Always)', and 'U' means 'Until'.

According to the classification of the properties in section 4.1, we abstract the verifying properties from the specified scenario item by item. Table 2 lists the properties with their corresponding CTL notations.

The representative properties abstract from the "Shunting Initiated by Driver" scenario are presented as follows.

- 1) *Reachability*. The onboard equipment can enter the SH mode.

CTL: SPEC EF(onboard.in\_SH)

- 2) *Transition*. The driver selects "Shunting" while the train is at a standstill and the ETCS is operating in level 2. When the onboard equipment received the SH permission from the RBC, the transition to the SH mode shall be made. Considering the driver as the environment of the ETCS system, we change the name of driver actions into external\_Event.

CTL: SPEC EF((train.in\_Stop & external\_Event.env\_SelectSH & (level = 2) & RBC.ev\_SHPermission) & EX(onboard.in\_SH))

- 3) *Definitiveness*. The train should not be controlled by ETCS L1 and ETCS L2 simultaneously.

CTL: SPEC !AG(in\_Level2 <-> in\_Level1)

- 4) *Mutual exclusion*. The train should not move when the onboard equipment is waiting for SH permission.

CTL: SPEC !EF(train.in\_Moving & onboard.in\_WaitForSHPermission)

- 5) *Liveness*. The onboard equipment can transfer to SH mode eventually.

CTL: SPEC AG(EF(onboard.in\_SH))

Table 2: The properties and their CTL notations.

Properties	CTL formulas	Properties	CTL formulas
Reachability	EF(state=s1 state=s2)	Definitiveness	!AG(state=s1 <-> state=s2)
Transition	EF(state=s1 & EX(state=s2))	Safety	AG(!C)
Deadlock	AG(EX(state = s1))	Liveness	AG(EF(state=G))
Mutual exclusion	!EF(state=s1 & state=s2)		

```

(EF onboard.in_SH).....true
(EF (((train.in_Stop&external_Event.env_SelectSH)&(level=1))&radioblo.....true
(~(AG (in_Level1<->in_Level2))).....true
(~(EF (train.in_Moving&onboard.in_WaitForSHPermission))).....false
(AG (EF onboard.in_SH)).....false
(AG (~(radioblockcentre.ev_SHRefused&onboard.in_SH))).....false

```

Figure 3: The properties' verification results.

6) *Safety*. If received the information “SH refused” from RBC, the onboard equipment should not switch to SH mode.

CTL: SPEC AG(!(radioblockcentre.ev\_SHRefused & onboard.in\_SH))

### 4.3 Check and analyze

Using NuSMV to verify the properties presented in section 4.2, we obtained the verification results as shown in Fig. 3.

In Fig. 3, only the first three properties are satisfied. However, the model checker is able to provide a counterexample for each of the remaining properties. We present the failing trace related to the verification of each property as follows.

- The forth property counterexample: When the onboard equipment switches to OS mode before the acceptance of shunting, the train is moving. This error may be caused by the interruption of radio communication with RBC or the message delay.
- The fifth property counterexample: The onboard equipment is locked in the state “WaitForSHPermission”, because the model does not consider the situation when no SH permission message is received from the RBC.
- The sixth property counterexample: If the ETCS L1 is controlling the train, the “SH Refused” message from RBC is not considered. So the onboard equipment does not switch to SH mode when receiving the “SH Refused” message.

The counterexamples show the traces of errors. The errors may arise from the translations, the models or the specifications. We should check whether the semantics of the CTL formulas comply with the properties described by natural language or not, and modify the models that are built incorrectly. Moreover, the realizability of the specifications should be considered. If the specifications are proved to be defective by the verification, we should provide the proof of errors and propose an amendment to the domain experts.

## 5 Conclusion and future work

In this paper, we propose an integration of scenario-based approach with a formal method to model and verify the SRS for ETCS L2. The significant advantage of the proposed method is that (a) the scenarios modeled by sequence diagram facilitates the communication between the domain experts and the designer, because UML is a *de facto* in the software industry; (b) differently from the testing and simulation, the formal method allows the analysis of all the

possible scenarios that ETCS L2 can possible generate, which assures the correctness, completeness and consistency of SRS for ETCS L2; (3) the mapping rules from the scenario to sequence diagrams and the formal method ensure the consistency between the SRS and the formal model. In the verification phase, the classification of properties and precise semantics of the temporal formulas specify the extensions and steps of model checking. The methods and steps stated in the paper give instructions regarding modeling and verification for specifications, and they can be used to other systems besides ETCS.

However, two problems are still unsolved. One is the simulation of a concurrent event when verified in the model checker NuSMV, and the other is the modeling of the continuous variable. Our future work is solving these two problems. Additionally, modeling and verifying the entire ETCS system are also the task of the future.

## Acknowledgements

This work is supported by National High-Technology Research and Development Program ("863" Program) of China No. 2009AA11Z221, National Science & Technology Pillar Program of China No. 2009BAG12A08, State Key Laboratory of Rail Traffic Control and Safety Autonomic Research Project Under grant No. RCS2008ZZ005, and "AnXing" Project of Beijing Jiaotong University.

## References

- [1] De Backer, M., Snoeck, M., Monsieur, G., Lemahieu, W. & Dedene, G., A scenario-based verification technique to assess the compatibility of collaborative business processes. *Data & Knowledge Engineering*, **68(6)**, pp. 531-551, 2009.
- [2] Theißen, M., Hai, R., Morbach, J., Schneider, R & Marquardt, W., Scenario-based analysis of industrial work processes. *Collaborative and Distributed Chemical Engineering. From Understanding to Substantial Design Process Support: Results of the IMPROVE Project*, ed. M. Nagl and W. Marquardt, Lecture Notes in Computer Science, vol. 4970. Springer-Verlag: Berlin and Heidelberg, pp. 433-450. 2008.
- [3] Fisher, J., Harel, D., Hubbard, E.J.A., Piterman, N., Stern, M.J. & Swerdlin, N., Combining state-based and scenario-based approaches in modeling biological systems. *Computational Methods in Systems Biology*, Springer-Verlag: Berlin and Heidelberg, pp. 236-241. 2005.
- [4] Kahyaoglu-Koraćina, J., Bassetta, S.D., Mouata, D.A. & Gertler, A.W., Application of a scenario-based modeling system to evaluate the air quality impacts of future growth. *Atmospheric Environment*, **43(5)**, p. 1021-1028, 2009.
- [5] Liang, H., Dingel, J. & Diskin, Z., A comparative survey of scenario-based to state-based model synthesis approaches. *Proc. of the 2006 Int. Workshop. On Scenarios and state machines: models, algorithms, and tools*, ACM: Shanghai, China, pp. 5-12. 2006



- [6] Amyot, D. & Eberlein, A., An evaluation of scenario notations and construction approaches for telecommunication systems development. *Telecommunication Systems*, **24**(1), pp. 61-94, 2003.
- [7] Palshikar, G.K. & Bhaduri, P., Verification of scenario-based specifications using templates. *Electronic Notes in Theoretical Computer Science*, **118**, pp. 37-55, 2005.
- [8] Lee, J., Pan, J.-I. & Kuo, J.-Y., Verifying scenarios with time Petri-nets. *Information Software Technology*, **43**(13), pp. 769-781, 2001.
- [9] Fahland, D., Oclets – scenario-based modeling with Petri nets. *Proc. of the 30<sup>th</sup> Int. Conf. On Applications and Theory of Petri Nets*, Springer-Verlag: Berlin and Heidelberg, pp. 223-242, 2009.
- [10] Autili, M., Inverardi, P. & Pelliccione, P., A scenario based notation for specifying temporal properties. *Proc. of the 2006 international workshop on Scenarios and state machines: models, algorithms, and tools*, ACM: Shanghai, China, pp. 21-28, 2006.
- [11] Mäkinen, E. & Systä, T., MAS – an interactive synthesizer to support behavioral modeling in UML. *Proc. of the 23<sup>rd</sup> Int. Conf. On Software Engineering*, IEEE Computer Society: Toronto, Ontario, Canada, pp. 15-24, 2001.
- [12] ITU-T, Recommendation Z. 120: Message Sequence Chart (MSC). Geneva, 2000.
- [13] Holzmann, G.J., (eds). *The spin model checker: primer and reference manual*, Addison-Wesley, 2004.
- [14] Cimatti, A., Clarke, E., Giunchiglia, F. & Roveri, M., NuSMV: a new symbolic model verifier. 1999, [http://www.cert.fr/esacs/doc/a\\_villafiorita\\_1.doc](http://www.cert.fr/esacs/doc/a_villafiorita_1.doc)
- [15] Weidenhaupt, K., Pohl, K., Jarke, M. & Haumer, P., Scenarios in system development: current practice. *IEEE Software*, **15**(2), pp. 34-45, 1998.
- [16] Rolland, C., Achour, C.B., Cauvet, C., Ralyté, J., Sutcliffe, A., Maiden, N., Jarke, M., Haumer, P., Pohl, K., Dubois, E. & Heymans, P., A proposal for a scenario classification framework. *Requirements Engineering*, **3**(1), pp. 23-47, 1998.
- [17] Uchitel, S., Kramer, J. & Magee, J., Negative scenarios for implied scenario elicitation. *ACM SIGSOFT Software Engineering Notes*, **27**(6), pp. 109-118, 2002.
- [18] Bai, X. Tsai, W.-T., Feng, K., Yu, L. & Paul, R., Scenario-based modeling and its applications. *Proc. of the 7<sup>th</sup> IEEE Int. Workshop. On Object-Oriented Real-Time Dependable Systems*, IEEE Computer Society, Los Alamitos, pp. 0253, 2002
- [19] Lima, V., Talhi, C., Mouheb, D., Debbabi, M., Wang, L. & Pourzandi, M., Formal verification and validation of UML 2.0 sequence diagrams using source and destination of messages. *Electronic Notes in Theoretical Computer Science*, **254**, pp. 143-160, 2009.
- [20] Tanuan, M.C., Automated Analysis of Unified Modeling Language (UML) Specifications, in Master's thesis, University of Waterloo, Canada. 2001.