IMPROVING SAFETY BY INTEGRATING DYSFUNCTIONAL ANALYSIS INTO THE DESIGN OF RAILWAY SYSTEMS

SANA DEBBECH, PHILIPPE BON & SIMON COLLART-DUTILLEUL
Université de Lille/Nord de France, IFSTTAR/COSYS/ESTAS, France

ABSTRACT

In order to cope with the increasing design complexity of safety-critical systems, safety assurance should be considered as early as possible in the design process. Using Model-Based System Engineering (MBSE) approaches however lead to new challenges regarding the cohesive integration of both safety engineering and system design along the system development process. Moreover, it helps to anticipate safety problems and detect errors as soon as possible. This is the case of railway systems, which are complex socio-technical systems. From this point of view, the purpose of the present study is to formalize a safety reasoning based on the definition of critical scenarios. The objective is to propose a proactive approach that takes these requirements into account early in the system architecture design. By identifying the impact on the design of the architecture, we will ensure safety by integrating technical devices and human interventions. Based on the related literature, the Preliminary Risk Analysis (PRA) is attested to define safety conditions. These safety requirements are expressed with a high level of abstraction according to the level of knowledge engineering. Qualitative risk analysis methods, such as Fault Tree Analysis (FTA) will be used to analyze the propagation of failures. The second challenge is to trace the high level requirements during the design steps. In order to help the designer to consider safety aspect in the system architecture synthesis, we integrate safety concerns from early design stages, within the MBSE approach. In this paper, we propose a methodology to effectively identify safety conditions, thus to anticipate risks. We also focus our work on the European Railway Traffic Management System (ERTMS). Finally, we applied specific transformation rules on our ERTMS ontology in order to build a Unified Modeling Language (UML) model.

Keywords: dysfunctional analysis, safety requirements, model-based safety engineering, ontology, ERTMS.

1 INTRODUCTION

Due to the rising complexity and basically ubiquitous application, more and more intensive systems become safety-critical in several domains like aerospace, nuclear and transportation. In such interactively complex systems, there are many branching paths among components making the interactions unpredictable to system designers and users. Therefore, complex systems are error prone and safety critical because errors lead to accidents with potentially catastrophic effects. Consequently, the design of such systems is challenging. Indeed, the increasing complexity of manufactured systems makes their development and safety analysis more difficult. Thus, big efforts are required to manage the complexity, maintaining coherence and consistency through the development and deal with numerous requirements relevant to multiple domains. In order to prevent as many accidents as possible, efforts are being focused on safety in many domains. Requirements for the development and life cycle of safety-critical systems present guidelines to make systems more and more fault-tolerant. However, a potential source of safety critical problems can only be anticipated if we integrate safety requirements as early as possible in the system architecture design. The identification of necessary safety conditions makes it possible to reduce the occurrence of dangerous situations, so as to guarantee a required Safety Integrity Level (SIL) [1]. This parameter impacts directly the system architecture design and determines the probability rates

WIT Transactions on The Built Environment, Vol 181, © 2019 WIT Press
www.witpress.com, ISSN 1743-3509 (on-line)
doi:10.2495/CR180361

of the deviation from fulfilling the system functions. Its possible values range between "0" (less critical) and "4" (most critical). The design of a specific system and its subsystems depends on the value of the SIL associated with each functionality of the system. Thus a system architecture including SIL4-functionalities must guarantee the maximum level of safety integrity, which would for example imply adding redundant hardware nodes. In railway systems, most safety functions related to the infrastructure or the movement of rolling stock are SIL4.

Because more and more functionality is transferred from hardware to software components, it becomes continuously harder to verify safety aspects. Model-based safety analysis can help solve this problem by finding causal connections between component malfunctioning and overall system hazards. Besides, taking advantage of these notions requires to build models combining both functional and dysfunctional aspects. Most of the current practices on the system safety assurance rely mainly on manual processes. Currently, the railway safety community refers to qualitative methods such as the PRA or the FTA before the design process. Nevertheless, it does not exist a tool-based methodology that makes possible the integration of the safety assessment earlier into the design stages.

In this paper, we will present our approach that improves safety in railway systems as soon as possible in the system architecture design. Firstly, we will define an ontology of the ERTMS system to have a structured and non ambiguous presentation of the system. Then, we will transform it into a UML class diagram to remedy the communication problem between safety engineers and design engineers with the semi-formal model. The aim of this study is to define the necessary safety requirements that will be taken into account along the design phases. They are expressed with a high level of abstraction as a first challenge. The second one is to trace their evolution and their impact on system's other requirements. The paper starts with the description of our proposed methodology. The following section present the case study, its modeling and its safety analysis. Section 4 discusses some related work and compares it with ours. Finally, we conclude the paper and give some perspectives.

2 SAFETY ANALYSIS INTEGRATION IN AN MBSE APPROACH WITH UML

System Engineering (SE) approaches offer relevant solutions for formalizing and comprehending complex systems. Therefore, safety assessment is, most of the time, implemented manually and separately from the design process. Both are very dependent of the skill and the experience of engineers. This problem is amplified by the fact that safety and system design engineering have developed their own techniques and methodologies. For the railway domain, safety analysis is executed through standard methods such as FTA [2], PRA [3] or Failure mode, effects and criticality analysis (FMECA) [4] and formal verification methods like the B method [5]. In order to avoid error-prone processes and to involve both safety engineering and system design, we employ *a model-based safety engineering*. These approaches allow to share the same model of the system by safety engineers and system designers, while using different views of it. Moreover, it enables to save time and, more importantly, it makes safety assurance as explicit part of an iterative design process. Indeed, Model-Driven Engineering (MDE) [6] is being successfully adopted in many domains and industrial research projects [7]. The direct benefit of MDE is automating a part of the process of safety assurance. For example in the proposed case study, by calculating certain information related on the position of the train or its speed when the component involved will be faulty.

As a contribution within this paper, we propose a methodology to integrate safety analysis into the design process from the first stages for railway systems. The aim of this study is to anticipate safety problems and decrease risks related to the movement of rolling stock,

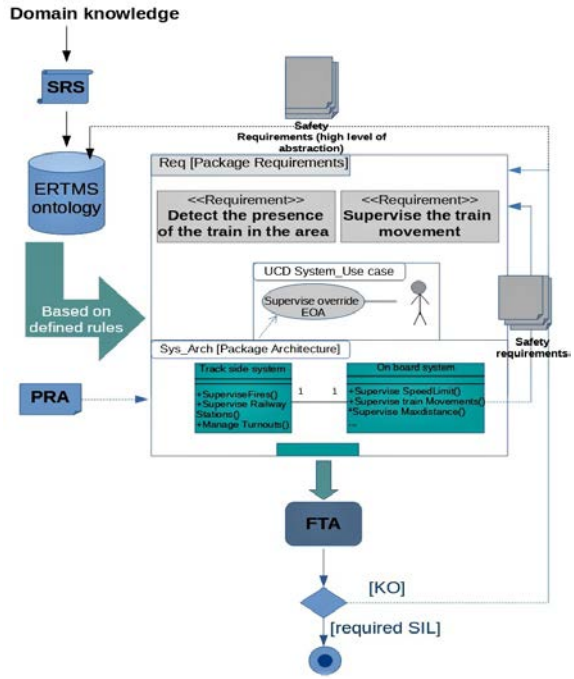

Figure 1: The overall methodology.

the infrastructure and to the humans errors. We focus our work on the ERTMS system. In order to regroup and create a formal structure of the concepts of this domain into a web of knowledge, the knowledge of ERTMS is formalized using several notations. In this study, we will use the ontology to model and formalize the ERTMS System Requirements Specifications (SRS). Ontologies are structured representations of knowledge of a certain domain. Several definitions of the term "ontology" have been provided. Ontologies present their own methodological and architectural peculiarities. On the methodological side, the main peculiarity is the adoption of a highly interdisciplinary approach, where philosophy and linguistics play a fundamental role in analyzing the structure of a given reality at a high level of generality and in formulating a clear and rigorous vocabulary [8]. On the architectural side, the most interesting aspect is the centrality of the role that an ontology can play in a complex system, leading to the perspective of *ontology-driven complex systems modeling*. The railway domain is an environment where numerous heterogeneous information sources exist. The ERTMS system basically relies on information exchange. Ontologies provide a number of useful features for intelligent systems, as well as for knowledge representation generally [9]. In the railway domain, documents describing the SRS [10], provided by the European Railway Agency (ERA) were issued with the specific aim of explaining and clarifying the usage of a part of the terms/concepts used in this domain and of the system itself. The overall proposed methodology is summarized in Fig. 1.

In this study, we choose an ontology creation tool using the Web Ontology Language (OWL), called the Protégé tool. Protégé 5.2.0 was developed by Mark Musen's group at Stanford Medical Informatics. Protégé's plug-in architecture can be adapted to build both simple and complex ontology-based applications. In this environment, concepts are

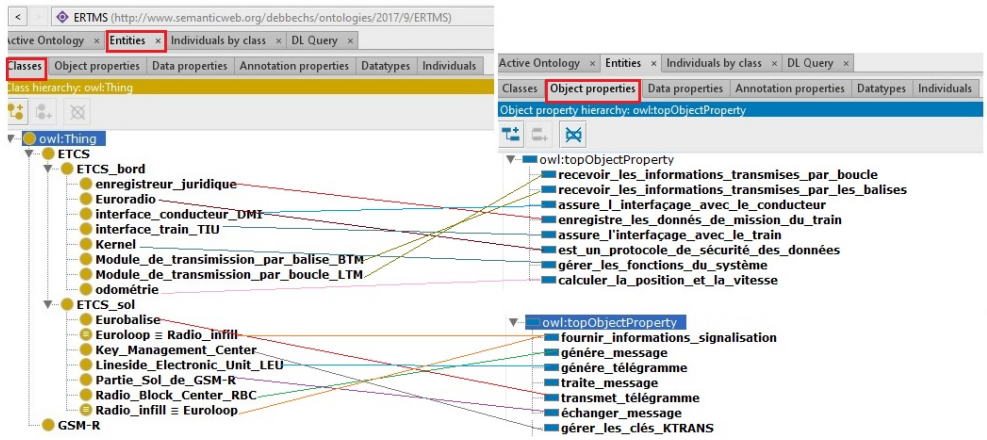

Figure 2: The ERTMS ontology.

Table 1: Rules of analogy Ontology/UML class diagram.

Ontology	UML class diagram
Entities	Classes
Properties of entities	Attributes of classes
Functions	Operations
Relations between entities	Associations between classes

formalized as classes together with their types of properties and relations between them. Our ERTMS ontology is composed of 3 layers, based on the ERTMS architecture: ETCS on-bord, ETCS Trackside and Global System for Mobiles-Railway (GSM-R). These sub-systems are formalized as classes. Every sub-system (class) is composed of several components, which each one has its own functions, properties and communication interfaces (relations) with the other components. The ERTMS ontology, modeled with a high level of abstraction, is presented in Fig. 2.

As mentioned before, we use the SRS documents, written in natural language (English) and we built our ontology in French. The asserted class hierarchy view is one of the primary navigation devices in Protégé. It is presented as a tree where nodes represent *classes*. A child-parent relationship in the tree represents a sub/super class relationship in the class hierarchy. Moreover, a class will be shown as a child of another class in the tree if it is asserted to be: (1) a *SubClassOf* that other class, or (2) if it is asserted to be *EquivalentTo* a class expression that is an intersection containing that other class as an operand. For example, in our ERTMS ontology, the *radio infill* is equivalent to *the euroloop* in the ETCS track side. In order to have an understandable representation of the ERTMS system model, we translate this ontology to a UML class diagram based on some defined rules [11]. These rules are applied by analogy and we summarize them in Table 1.

For the purpose of supporting the SE approach, we must use a tool to model system requirements and the ERTMS system architecture and ensure the coherence between these different views. Indeed, we choose the UML thanks to its several advantages, semi-formal capabilities of modelisation and formal semantics [12]. Simultaneously, we use the Object

Constraint Language (OCL) that is a formal language used to express side-effect free constraints in a UML model. Semantics for OCL includes necessarily semantics for class diagrams [13]. Consequently, the ERTMS ontology is modeled in UML class diagrams and OCL constraints. The OCL offers many advantages in our study, for instance, the expression of constraints that cannot be described using the description logic. It is a powerful language in terms of avoiding ambiguity (formal language) and ensuring consistency of the OCL expression with the rest of the model by the complete checking. Moreover, UML is more and more applied in different academic and industrial projects because it makes easier communication and manipulation of the same system model by design engineers and safety engineers [14]. For example, UML class diagrams are used for representing ontologies and UML object diagrams for representing instance knowledge [11]. To sum up the first step of our methodology, we define the ontology of ERTMS and we model it on UML class diagram based on defined rules of transformation presented in Table 1.

Then, once the functional model is available, a functional hazard assessment (Preliminary Risk Analysis) can be achieved and a list of failure modes of functions/components of the system and their effects can be defined. Indeed, functions of the system are classified according to their criticality, so their SIL. Consequently, safety requirements derived from the PRA will decrease dangerous situations and reach the required SIL. Furthermore, railway risks related to the movement of the train, the infrastructure or human errors are well known. In order to anticipate safety problems and ensure the overall reliability of the system, we take into account safety conditions early in the design phases. In this level, we can built the Fault Tree Analysis (FTA) related to a specific dangerous situation.

Requirement engineering is a primordial activity in the architecture system design. A requirement specifies the capability or condition that must (or should) be satisfied. A requirement may specify a function that a system must perform or a performance condition a system must achieve [15]. As indicated below, the aim of our study is to integrate new safety requirements as soon as possible in the design stages. So, we have to formulate safety requirements with a high level of abstraction in order to have a flexible integration of requirements and have a good visibility of the requirements evolution. Safety requirements can be grouped into two categories [16]: (1) requirements related to compliance and good practice; (2) specific system performance related requirements.

In this paper, we focus on the second category related to the system performance. Safety requirements can be expressed according to the desired actor who is involved to accomplish the failed service. We have two methods to anticipate safety problems: *component redundancy* and *human interventions*. Component redundancy is the most applied protection measure in the hazard log. This mitigation measure is related to novel equipment, or new processes, or any novel environment states within which the regular *equipment* or *processes* is to operate. This add-on intervention provokes spending in terms of cost and maintenance. Undoubtedly, high priority is given to the overall safety of the system and the users. Once the safety requirement deduced from the hazard assessment and expressed with the high level of abstraction, it will be integrated into the system architecture. The second challenge of our study is to trace the requirement's evolution. Indeed, we aim to model the relationships between requirements and integrated safety requirements. Consequently, we will keep track of the requirement's interaction. Once the functional model is reliable, the required SIL is achieved. The inherent complexity of complex systems imposes the use of powerful tools for the implementation of the requirements traceability. In this study, we choose SysML (Systems Modeling Language) thanks to its advantages to model requirements explicitly [15]:

- The requirements model allows to centralize the verification of the affectation of requirements at least to one component of the model;

- SysML provides a way to express traceability links formally;
- SysML distinguish two categories of links: (a) between requirements (*containment*, *derive*, *copy*, *trace*); (b) between requirements and the implementation (*satisfy*, *verify*, *refine*);
- SysML supports three representations: graphic, tabular or tree;
- Annotation mechanisms capture design choice's arguments.

SysML is a general-purpose modeling language for SE applications. It supports the specification, analysis, design, verification and validation of a broad range of systems and systems-of-systems. SysML is defined as an extension of a subset of the UML using UML's profile mechanism [17]. In this study, we are particularly interested by its ability to represent text-based requirements and relate them to other modeling elements. Indeed, a requirement is defined as a stereotype of UML Class subject to a set of constraints. A standard requirement includes properties to specify its unique identifier and text requirement [15]. Additional properties such as verification status, can be specified by the user. Several requirements relationships are specified to relate requirements to others as well as to other model elements. It defines a requirements hierarchy, deriving requirements, satisfying requirements, verifying requirements and refining requirements. Thus, a generic trace requirement relationship provides a general-purpose relationship between a requirement and any other model element. Finally, we illustrate our methodology on a case study from the ERTMS level 2.

3 A CASE STUDY

In this paper, the work focus on the ERTMS system level 2. Particularly, one of the most critical safety functions: *the occupancy of the area by the train* is chosen. The case explained in this paper revolves around this safety function. The hazard assessment is implemented in order to decrease railway risks, particularly related to the infrastructure and the rolling stock traffic such as:

- The rear-end collision: a train hits another one in front of him in the same way. The Pomponne disaster on 23 December 1933 is one of the most tragic examples [18].
- The frontal collision between trains: it is most often due to human errors. The collision of Zoufftgen on October 2006 is an example [19].
- The side collision can happen when two tracks converge at a point and two trains are in the same way. The accident of Provins on 6 February 2005 is an example [20].

In order to explain the functioning of the component involved in this task, the role of relays of the track circuits have to be described. As a matter of fact, lines are equipped with railroad communications system, called Automatic Block Signaling (ABS). These ABS are used for regulating train traffic and for dividing railroad light signals that change automatically through the action of moving trains. Under this system the runs between stations are divided by automatic communication light signals into block sections from 1 to 3 km long. The distance between the light signals is the minimum distance for the safe simultaneous traffic of trains. The automatic action of the intermediate light signals within each block section is brought about by electric *track circuits* separated by insulating joints. One end of each track circuit is joined to the source of power and the other to a rail relay switch that controls the light signals through a contact system. If the block section is free, the current is fed from the battery by the rails through the winding of the track relay, thus turning the light signal to "proceed" (green). When the first wheels of a train make contact with the track circuit, the track relay turns the light signal to "stop" (red). The track circuits

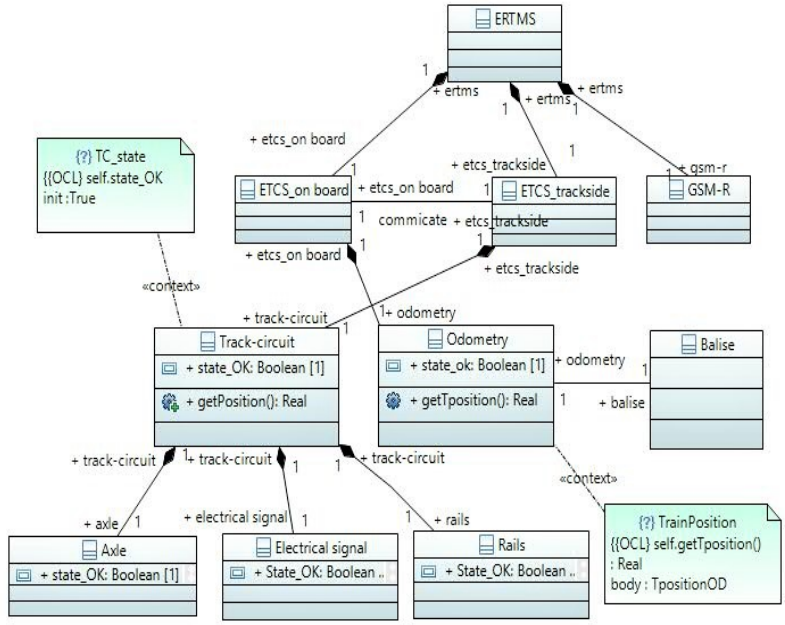

Figure 3: UML class diagram for a part of the ERTMS system.

also monitor the conditions of the rails: if any rail is broken or missing, the track relay turns on and turns the light signal to red.

As mentioned below, in this study the ERTMS/ETCS Level 2 is considered. This level is a digital radio-based signal and train protection system (ATP). It involves continuous supervision of train movement with continuous communication, which is provided by GSM-R, between both the train and the track-side. Line-side signals are optional in this case. The train detection is performed by the track-side equipment which is out of the scope of ERTMS.

The first step of the proposed methodology is the elaboration of the ERTMS ontology. As shown on Fig. 2, the ERTMS ontology is composed of 3 classes: ETCS track-side, ETCS on-board and GSM-R. We choose to model the ontology as a static model consisting of a class diagram to specify the classes in the domain and their relationships. In order to implement ontologies, all attributes are specified in public visibility because an ontology is a shared public view of a domain [11]. As previously argued in Section 2, UML is not used because it is a graphical syntax but rather because it represents a knowledge semi-formal notation. Moreover, the ontology representation defined in this paper is the application of UML and OCL. The Fig. 3 presents the UML class diagram defining the part of the ERTMS system involved in the critical function described below.

UML diagrams are modeled using *Eclipse Papyrus 5.2.0*. It provides an integrated, user-consumable environment for editing any kind of Eclipse Modeling Framework model and particularly supporting UML and related modeling languages such as SysML and MARTE. Constraints expressed in OCL are related to the *train position* obtained from the odometry and/or the track circuit. It is the principal attribute in the class diagram. This attribute allows the comparison between the values obtained from these two components and the check of the corresponding area's occupancy by the function *get TrainPosition*. The purpose of the study

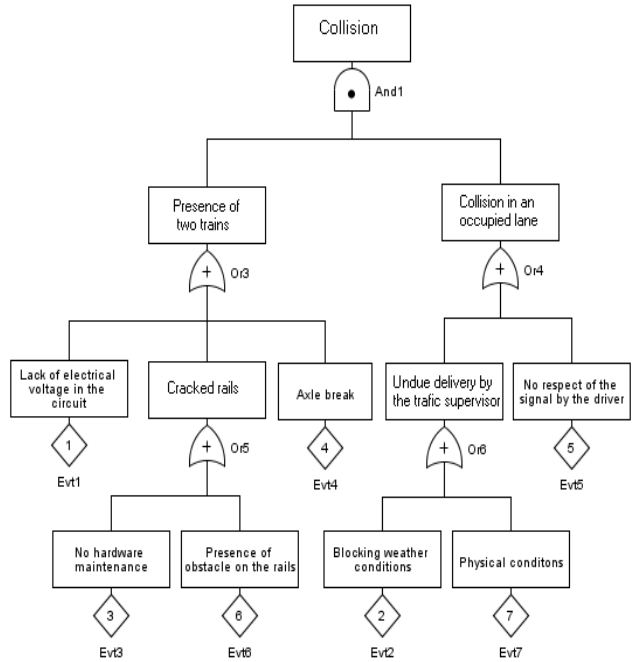

Figure 4: The Fault Tree for "collision between two trains".

is to decrease railway risks related to the train traffic. To anticipate these safety problems, we propose to integrate the dysfunctional analysis as early as possible in the architecture system design. In the beginning of the case study, the functional analysis of the track circuit has been presented. Indeed, it is essential to understand its functioning in order to have a precise analysis of its component's failures. Then, the track circuit's potential failure modes are determined. The failure mode is related to its principal components: the axle, the contact wheel-rail and human errors. Parsing the interactions among track circuit components, the failure propagation can be inferred and the Fault Tree is built. For the safety critical function, the undesirable event or the dangerous situation is *the collision between two trains*. The FTA defining this Top Event and its causality relations is given in Fig. 4.

Once the FTA of the specified dangerous event is obtained, the mitigation measures to decrease this risk are considered. These protection measures are expressed with a high level of abstraction in the form of safety requirements. As a matter of fact, when a failure of one component of the track circuit is detected, the odometry sub-system will accomplish the task: communicating the train position to the on-board equipment according to the balise information. This is the counter measure against this failure to ensure the reliability of the system. Then, we integrate this safety requirement into the UML class diagram as an association class between the track circuit and the odometry and OCL constraints. In the case study, the odometry is assumed to be always well functioning. It is well known that the on-board odometry may have a lake of precision. However this drawback is not always critical: the corresponding system analysis is not detailed in the present study. The *get position* operation takes into account the failure state of the track circuit and devolve the ETCS on-

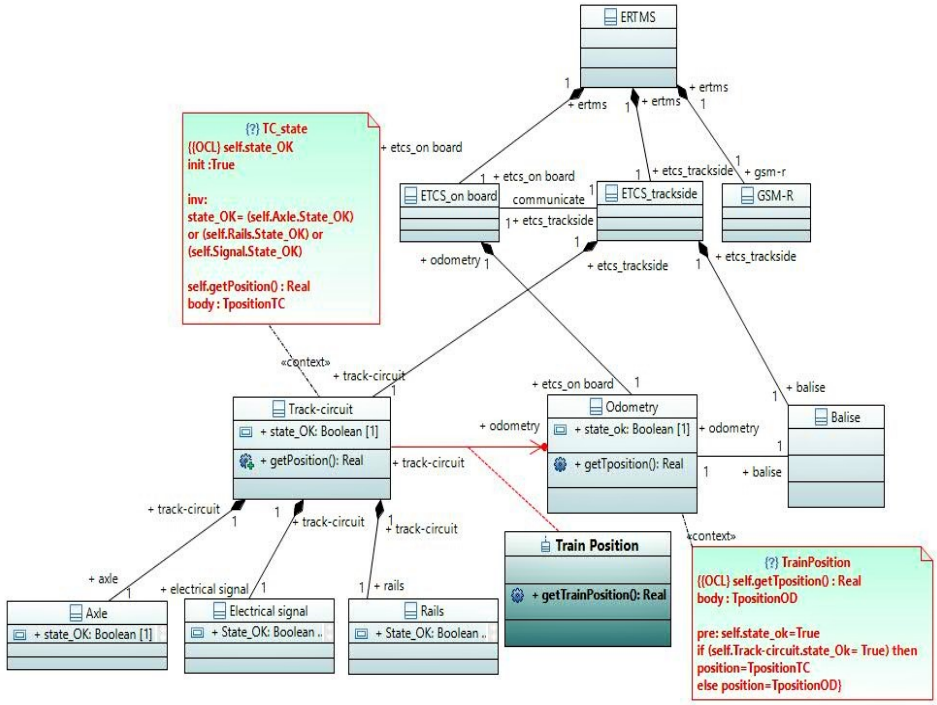

Figure 5: UML class diagram with new safety requirements.

board (odometry) to obtain the train position. Fig. 5 presents the UML class diagram with the integrated safety requirement.

The second challenge of the present study is the requirement's traceability to ensure the coherence in terms of the requirement's interactions and consequently the reliability of the system. For this purpose, the SysML-based *requirement diagram* is defined. Moreover, several requirements relationships can be specified and a trace of requirement's evolution can be stored. A particular focus is put on the mechanism of interactions between existing requirements and integrated requirements. Fig. 6 shows the several relationships between requirements expressed with a high level of abstraction. The new safety requirement integrated in the requirement's package model is derived from the "the detection of the train passing" realized by the track circuit. This link is justified by its execution if and only if the track circuit is defective. Consequently, the on-bord system delegates the odometry to obtain the train position: This safety requirement has a containment relationship with "the management of the on-board system". Finally, the requirement model is validated and the system performance is ensured.

The sequence diagram is presented in Fig. 7 in order to explain its application into the overall system and its behavior compared to that of the other components. This dynamic diagram depicts the sequencing of events when the safety requirement is integrated in the system model. Finally, as shown in Fig. 5, Fig. 6 and Fig. 7, the integration of the safety requirement into the system architecture design allows to anticipate safety problems and consequently the overall reliability of the system.

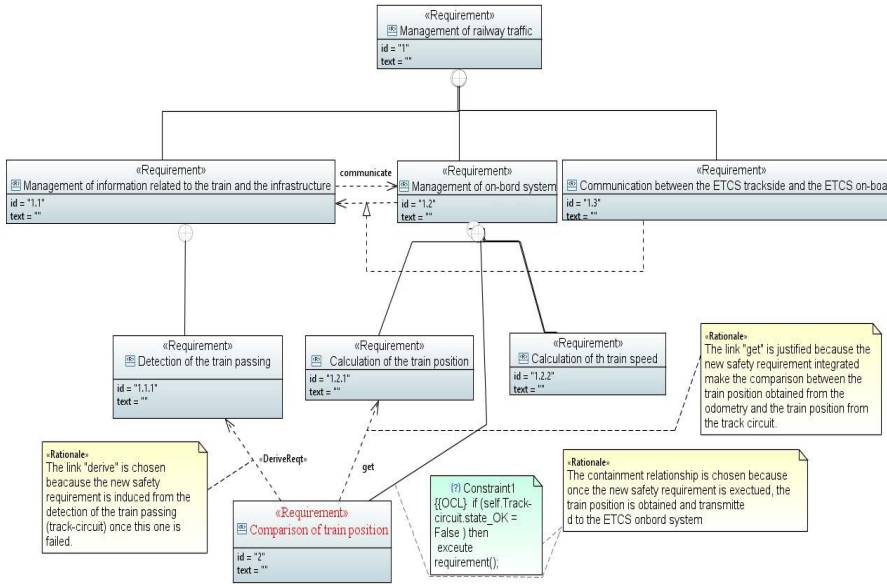

Figure 6: SysML requirement diagram with integrated safety requirement.

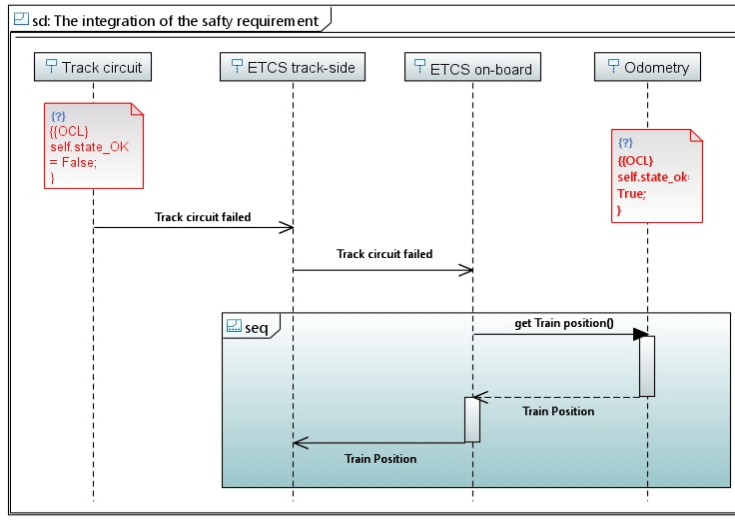

Figure 7: The UML sequence diagram of the application of the integrated safety requirement.

4 CONCLUSION AND OUTLOOKS

The aim of this study is the integration of the dysfunctional analysis into the railway system design and specially the ERTMS system. This paper has presented an MBSE methodology to take into account safety analysis as early as possible during the design phases. It is illustrated with a case study from the ERTMS/ETCS level 2 and a critical function is chosen. In fact, an ERTMS ontology is constructed in order to have a structured representation of the domain knowledge. Then, we apply some defined rules to transform it into a UML class diagram. After determining the Track circuit's components failure modes, the FTA of the dangerous event related to the safety function is built. Therefore, the analysis of the fault propagation allows the proposition of protection measures to anticipate safety problems. These mitigation measures are expressed with a high level of abstraction as safety requirements. Their integration into the requirements package model leads to interactions with the other requirements. Finally, a requirement diagram is set to explain the overall component's behavior.

In the field of System Engineering processes, the best practices are supported by a wide theoretical and technical documentation. The integration of the dysfunctional analysis or safety concerns in general-purpose modeling process is a big challenge that has been explored in many directions. In this paper, we focus on previous works which are receiving specific attention in the system engineering community. Feiler and Rugina [21] proposed a framework to model the error state propagation in a hierarchical architecture. They demonstrate that error propagation can occur at the components level, at the hardware level and between the hardware and components. In order to limit or avoid the error propagation, the authors define adapted filters (guards), for example between the interconnection of components. For the same purpose, some industries and academics defined an architecture description language called EAST-ADL [22] in order to specify component-based software infrastructures in automotive applications. It allows the modeling of the system failure behavior and its analysis using safety analysis tools. In [23], the authors proposed a meta model using UML class diagrams and OCL constraints to integrate safety concerns into SE processes. In addition, they defined redundancy policies for updating the dynamical allocation of functions caused by dysfunctional events. In their paper [24], Guillerm et al. describe a method for declining safety requirements of complex systems. The refinement of the requirement notion for treating the safety ones is a necessary step or achieving the safety integration. In [14], Cancila et al. use a UML profile to integrate some safety concerns in SE processes. This language allows the risk analysis and define automatically some safety attributes such as the SIL. However, these works didn't consider the requirements traceability after the integration of safety requirements in the system architecture design. In addition, none of them take into account the transition from the system ontology to a semi-formal model (UML class diagram). In this paper we proposed a methodology based on a high level abstraction of safety requirements to fill these gaps.

Moreover, we intend to propose tooling contributions to assess the failure propagation related to a specified component of any critical safety function. In a later phase, some evaluation parameters will be identified in order to investigate the system reliability and the operational security.

REFERENCES

[1] Summers, A.E., Techniques for assigning a target safety integrity level. *ISA Transactions*, **37**(2), pp. 95–104, 1998.
[2] Limnios, N., *Fault Trees*, John Wiley & Sons, USA, 2013.

[3] Mortureux, Y., *Preliminary risk analysis*, Techniques de l'ingénieur. Sécurité et gestion des risques, SE2 (SE4010): SE4010, 2002.
[4] Bouti, A. & Kadi, D.A., A state-of-the-art review of FMEA/FMECA. *International Journal of Reliability, Quality and Safety Engineering*, **1**(4), pp. 515–543, 1994.
[5] Abrial, J.R., *The B-Book: Assigning Programs to Meanings*, Cambridge University Press, UK, 1996.
[6] Schmidt, D., Model-driven engineering. *IEEE Computer*, **39**(2), pp. 25–31, 2006.
[7] Ougier, F. & Terrier, F., ADONA: an open integration platform for automotive systems development tools, *From Model-Driven Design to Resource Management for Distributed Embedded Systems, IFIP TC 10 Working Conference on Distributed and Parallel Embedded Systems (DIPES)*, 2006.
[8] Guarino, N., Formal ontology and information systems. *Proceedings of FOIS*, **98**, pp. 81–97, 1998.
[9] Hoinaru, O., Mariano, G. & Gransart, C., Ontology for complex railway systems application to ERTMS/ETCS system. *FM-RAIL-BOK Workshop SEFM'2013 11th International Conference on Software Engineering and Formal Methods*, 2013.
[10] E. U. G. UNISIG, System Requirements Specification (SRS) version 3.4.0, E. R. Agency, 2016, http://www.era.europa.eu/Document-Register/Pages/Set-2-System-Requirements-Specification.aspx. Accessed on: 2 May 2017.
[11] Cranefield, S. & Purvis, M., UML as an ontology modeling language. *Proceedings of the Workshop on Intelligent Information Integration, 16th Int. Joint Conference on AI (IJCAI-99)*, Germany, 1999.
[12] Manfred, B. & Cengarle, M.V., UML formal semantics: lessons learned. *Software and Systems Modeling*, **10**(4), pp. 441–446, 2011.
[13] Richters, M. and Gogolla, M., On formalizing the UML Object Constraint Language OCL, eds T.W. Ling, S. Ram & M.L. Lee, 17th *Int. Conf. Conceptual Modeling, Lecture Notes in Computer Science*, Volume 1507, Springer-Verlag, 1998.
[14] Cancila, D., Terrier, F., Belmonte, F., Dubois, H., Espinoza, H., Gérard, S. & Cuccuru, A., Sophia: a modeling language for model-based safety engineering, *MoDELS ACE-MB*, Denver, CO, pp. 11–25, 2009.
[15] Object Management Group: SysML v 1.5 Online. www.omg.org:spec/SysML/, p. 161. Accessed on: 6 Dec. 2017.
[16] Lucic, I., *Risk and Safety in Engineering Processes*, Cambridge Scholars Publishing, UK, 2015.
[17] Systems Modeling Language, Online. https://en.wikipedia.org/wiki/Systems_Modeling_Language. Accessed on: 6 Jan. 2018.
[18] The Pomponne accident, Online. https://fr.wikipedia.org/wiki/Accident_ferroviaire_de_Lagny-Pomponne. Accessed on: 9 Nov. 2016.
[19] The Zoufftgen collision beatt report, Online. http://www.bea-tt.equipement.gouv.fr/resume-du-rapport-final-a164.html. Accessed on: 9 Nov. 2016.
[20] The Provins accident beatt report, Online. http://www.bea-tt.developpement-durable.gouv.fr/resume-du-rapport-final-a51.html. Accessed on: 9 Jan. 2018.

[21] Feiler, P. & Rugina, A., Dependability Modeling with the Architecture Analysis & Design Language (AADL). Technical report, Software Engineering Institute, Carnegie Mellon, 2007.
[22] ATESST Project. Advancing Traffic Efficiency and Safety through Software Technology. ATESST STREP – FP6 project, Online. http://www.atesst.org. Accessed on: 9 Mar. 2017.
[23] Piriou, P.Y., Faure, J.M. & Deleuze, G., A meta-model for integrating safety concerns into systems engineering processes. *7th Annual IEEE International Systems Conference (SysCon) 2013*, Orlando, FL, pp. 298–304, 2013.
[24] Guillerm, R., Demmou, H. & Sadou, N., Combining FMECA and Fault Trees for declining safety requirements of complex systems. *Advances in Safety, Reliability and Risk Management: ESREL 2011*, pp. 207, 2011.