OETCS/WP4/D4.2.3V02

openETCS

ITEA2 Project
Call 6 11025
2012 – 2015

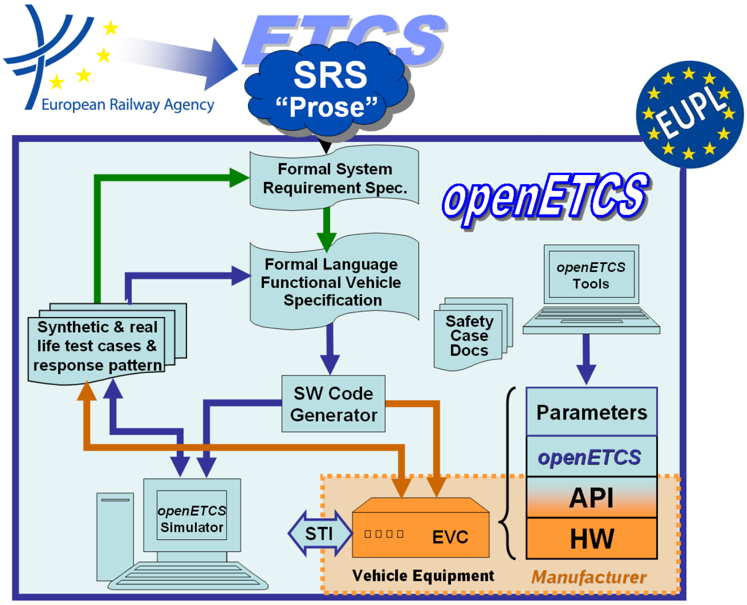
Work-Package 4: “Validation & Verification Strategy”

openETCS Hazard and Risk Analysis Methodology

Guideline for the safety related activities in an openETCS Onboard Unit development

Jan Welte

March 2014



Funded by:

Federal Ministry of Education and Research

République Française
Ministère de l'Enseignement Supérieur et de la Recherche

Investissements d'Avenir

Région de Bruxelles-Capitale

Gobierno de España
Ministerio de Industria, Energía y Turismo

This page is intentionally left blank

Work-Package 4: “Validation & Verification Strategy”

OETCS/WP4/D4.2.3V02
March 2014

openETCS Hazard and Risk Analysis Methodology

Guideline for the safety related activities in an openETCS Onboard Unit development

Jan Welte

Technische Universität Braunschweig
Institute for Traffic Safety and Automation Engineering
Langer Kamp 8
38106 Braunschweig, Germany
eMail: openetcs@iva.ing.tu-bs.de
WebSite: www.iva.ing.tu-bs.de

Output Document

Prepared for openETCS@ITEA2 Project

Abstract: This document presents an overview about the main activities performed during the OpenETCS development process to identify hazards inside the openETCS software and to assess the risk resulting from these hazards. Thereby, a consistent methodology is derived, which connects the hazard and risk analysis for the system and subsystems from which potential hazards and corresponding risks will be derived with methods to identify and enforce to ensure the overall quality during the design of all safety related parts of the system. Overall, the safety plan will guarantee that the safety design activities required in D2.6 are covered and all safety requirements are fulfilled.

Table of Contents

Figures and Tables

Figures

Tables

Document Control

Document information	
Work Package	WP4
Deliverable ID or doc. ref.	D4.2
Document title	openETCS Safety Plan
Document version	00.20
Document authors (org.)	Jan Welte (TU-BS)

Review information	
Last version reviewed	–
Main reviewers	–

Approbation			
	Name	Role	Date
Written by	Jan Welte	WP4-T4.4 Task Leader	October 2013
Approved by	–	–	

Document evolution			
00.01	01/10/2013	Jan Welte	Document creation
Version	Date	Author(s)	Justification
00.1	28/01/2014	Jan Welte	Extended Introduction
00.2	14/03/2014	Jan Welte	Reword document structure to incorporate review comments from internal assessment

1 Introduction

During system development the system limits and components have to be established before the necessary steps can be taken to ensure that the system behavior is not unsafe. In this context safety is understood as protecting the environment from hazards coming from the system as distinguished from security which covers protecting the system itself from hazards coming from the outside. Respectively, safety and security are comparative terms which have to be specified in context of the system for which they shall be proven. The standards for system development like EN 50126, EN 50128 and EN 50129 provide only guidelines how safety for a system shall be determined and assured by providing certain management principles and methods for the respective system development. As the openETCS project is related to a number of different system definitions like the overall railway system, the on-board unit, the Kernel software and the development tool chain different safety aspects have to be taken in consideration. Only a small number of these can actually be determined in the openETCS context alone.

1.1 Purpose

The safety plan is part of the overall safety process which is "the series of procedures that are followed to enable all safety requirements of a product to be identified and met" [**?**]. To ensure that the safety process is implemented in a proper way the EN 50129 requires a safety management which is consistent process for RAMS presented in the EN 50126.

The purpose of the safety plan is to show in details how the safety requirements will be implemented over the development process. To do this the plan has to present the management structure, all related activities and documentations over the life-cycle, as well as the approval mile-stones and review requirements. As the product life-cycle is an ongoing process and, specifically in a project like openETCS, iterative changes are taking place, the safety plan has to be updated to assess the respective safety effects. Thereby, the safety plan will also include the plan for the safety case, which itself is the final assessment document to demonstrate the actual product compliance with all specified safety requirements which have been implemented according to the safety plan.

As the movement characteristics of a train set specific limits in which a driver alone is able to avoid derailment or any kind of collisions railway signaling and protection systems have been developed to ensure safe train movements. Respectively, the mayor parts of a train control system like ETCS include functionalities which shall guarantee that the overall railway system does not get in a hazardous situation.

Since the openETCS project does not produce a specific train borne on-board unit the openETCS safety plan, will not cover all specific software and hardware aspects. The focus is mainly on the EVC kernel functionality and only partially on further real time and hardware/software integration aspects. However, this safety plan shall at least cover the safety process needed to ensure that the openETCS software development can be extended to become the functional basis for a complete on-board unit development.

1.2 Document Structure

As the openETCS software development process and the respective tool chain a used in an iterative process this version of the safety plan is also only one iteration step. As the design selectivities and the use of tools proceed further more details concerning the hazard identification methods, safety verification and validation principals and their corresponding result documentation will be added to this document. So far the version 1.0 of the safety plan will only cover the overall safety process and it general activities as well as the safety management activities which will create the safety plan as the final documentation document. Details are only presented for the hazard identification and evaluation method, as well as for the argumentation structure management method. All detailed activities concerning verification and validation of different safety requirements are closely connected to the V&V plan and all activities described there.

As safety is a term which is used in different context and therefore interpreted differently depending on the used reference system, the following section 1.3 presents the principal concept and terminology on which the following safety process is founded [**?**]. These section shall present and clarify the underlying understanding of safety as it in standards EN 50126, EN 50128 and EN 50129 for railway system development. Thereby, it is important to state more precisely the relation between quality and safety for software as this is the core aspect for the openETCS development.

1.3 Background Information

In the context of railway system development the EN 5012X and the ISO 900X standards are closely connected. Respectively, the terms safety and quality and related terms as reliability, maintainability, availability, usability and security are often used together. For example the EN 50129 names evidence of quality and safety management as conditions for a safety acceptance. The following sections shall give a common definition for all this aspects, which are required in a CENELEC confirm development process.

1.3.1 Quality

The ISO 9000 defines quality as "degree to which a set of inherent characteristics fulfills requirements" [**?**], which clearly shows a more objective view as the EN 50129, which defines quality as "a user perception of the attributes of a product" [**?**]. As for system development the given requirements always present the objective against which the system has to be measured this definition is consistent with the definition of verification and validation as it is presented in the EN 50128 and the openETCS V&V plan.

Depending on the respective field of application specific standards further specify the characteristics which shall be taken in account to establish quality. The EN 50126 as primary standard for railway applications gives RAMS as the major contributor for the quality of service.

1.3.2 RAMS

1.3.3 Software Quality

1.3.4 Safety

1.3.5 Probabilistic Safety Concept

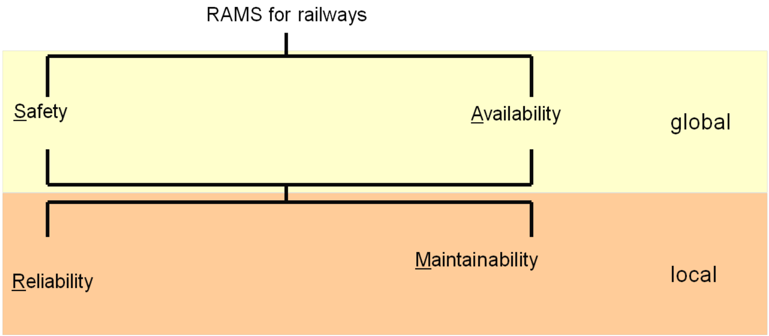


Figure 1. RAMS for Railway elements and their relations [?]

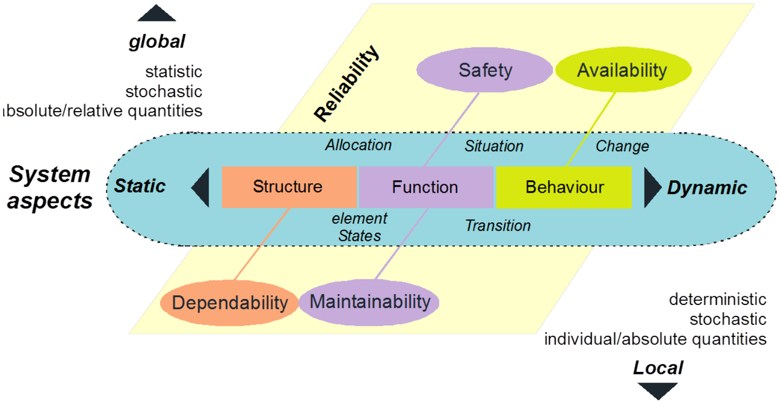


Figure 2. Systems Characteristics and their relations to RAMS in Railways [?]

Risk

The EN50128 standard defines safety as the “freedom from unacceptable levels of risk of harm to people" [?], which shows that the safety approach required by the CENELEC standards is risk-based. As the risk is defined as the “combination of the rate of occurrence of accidents and incidents resulting in harm (caused by a hazard) and the degree of severity of that harm" [?] this approach is based on a probabilistic understanding of event occurrence. The overall relations between all these safety-related terms used to define the safety properties, characteristics and quantities are outlined by the Risk-Genesis-Model of Schnieder, which is shown in the following figure 3.

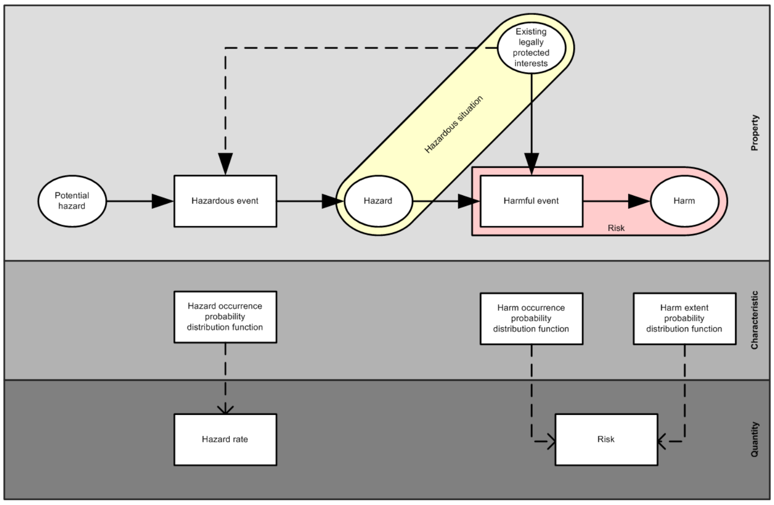

Existing legally protected interests
Hazardous situation
Property
Potential hazard
Hazardous event
Hazard
Harmful event
Harm
Risk
Hazard occurrence probability distribution function
Harm occurrence probability distribution function
Harm extent probability distribution function
Characteristic
Hazard rate
Risk
Quantity

Figure 3. Risk-Genesis-Model showing the relations between the safety-related terms [?]

This demonstrates that the first step is to define the system properties, specifically identifying the harms and their related hazardous situations. This has to be performed during a system hazard analysis. Afterwards the respective properties have to be determined by assessing the risk concerning the identified hazards. Based on this work safety integrity levels can be assigned to all system functionalities which are then allocated during the design to certain parts of the operational equipment. As this work is closely related to all design decisions, it has to be done iteratively for all abstraction levels during the system design.

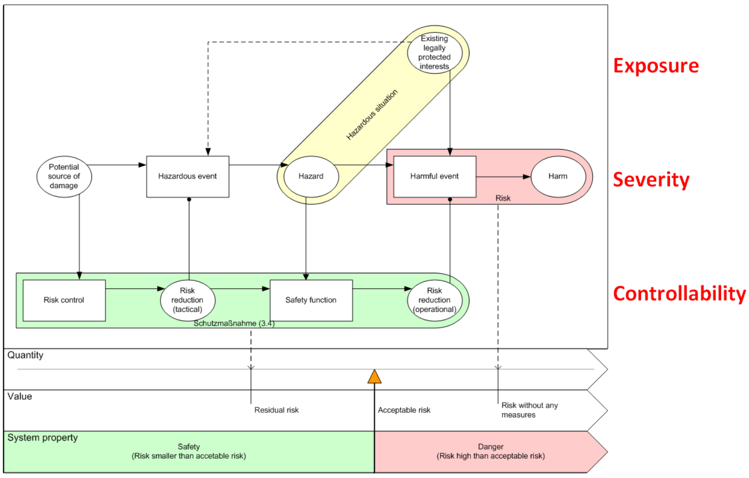

Existing legally protected interests
Hazardous situation
Potential source of damage
Hazardous event
Hazard
Harmful event
Harm
Risk
Risk control
Risk reduction (tactical)
Safety function
Risk reduction (operational)
Schutzmaßnahme (3.4)
Exposure
Severity
Controllability
Quantity
Value
Residual risk
Acceptable risk
Risk without any measures
System property
Safety (Risk smaller than accetable risk)
Danger (Risk high than acceptable risk)

Figure 4. Risk control process [?]

Safety Integrity Level

The safety integrity level represented the acceptable risk for every part of the system. The risk control process as it is presented in figure 4 is then performed to ensure that the safety integrity levels are reach by every part of the system.

Since software itself does not fails in the way technical equipment does the specific software safety integrity level represent a qualitative measure with respect to the required degree of correctness for the software functionality than a qualitative value for the likelihood of failing. To reach the needed degree of correctness for the software various design, verification and validation methods are required corresponding to the assigned software safety integrity level. This process leads to safety requirements which have to be implemented in the software design as well as verified and validated. Respectively the EN50126 describes the safety design process as a series of safety tasks for each life cycle phase. This task are related to a number of safety artifacts which are created, used and adapted over time through the different safety design activities.

1.3.6 Safety Case

The EN 50129 states that evidence of quality management, safety management, as well as functional and technical safety have to be provided for safety acceptance. The safety case shall be the structured safety justification document demonstrating that these conditions have been satisfied. Therefore, the British Ministry of Defence defines a safety case as “a structured argument, supported by a body of evidence that provides a compelling, comprehensible and valid case that a system is safe for a given application in a given environment.”[**?**] This definition emphasizes the distinction between the “argumentation” and the “evidences”, which corresponds to the distinction between rules coming from the legal requirements and facts resulting from the actual working process. This clear distinction between the safety argumentation and the evidences helps to structure the safety case and improvement of the discussions with the legal authorities.[**?**]

2 Document Evolution

This document will be further specified with the progress of the design process to update and detail the corresponding hazard and risk analysis methods. This specifically reference to the refinement of safety properties on different modeling levels and their corresponding verification and validation methods. All resulting changes and additional safety activities have to be maintained according to the EN 50128 and document according to the overall safety case process.

The openETCS development plan presented in chapter **??** is based on the available information in the Quality Assurance Plan and from WP 2 D2.3 and D2.4. As the development methods and processes are still evolving this document has to be adopted accordingly.

Concrete methods to verify and validate safety relevant properties derived from the hazard control methods descriped in this document, will be specified in the Verification and Validation plan. Also the tools used to support those activities will be detailed in this plan.

3 OpenETCS Development process

The main products of the openETCS project will be the OpenETCS specification model used to generate the OpenETCs on-board software and the OpenETCS tools chain development, which is used to formalizes the ERA Specifications for ETCS, generate the Software Code and perform verification and validation activities. Both parts have their own development process, but the focus focus hazard and risk analysis and the safety case shall be on the software development process. The openETCS tool chain shall only be considered with respect to its role in the software development process as the tool qualification is mainly out of this project. The safety related activities

3.1 Formalization and software development

The software development in the OpenETCS project shall be performed as an open, model-based, agile Software development. The respective combination of methods used during the development process shall comply with a SIL 4 development process according to EN 50128 for which the requirements are analyzed and shown in detail in D2.2. The overall openETCS Software development process presenting the development principals how the on-board unit requirements for ETCS shall be formalized first in a semi-formal SysML architecture model and afterwards in a detail formal SCADE behavior model, which is the basis for the automatic source code generation, are outlined in WP 2 D2.3 and D2.4.

The detailed principals and phases of this development process are presented in the Quality Assurance Plan. As this development process and the supporting tools are still evolves the detailed development description has to be updated as the project continues.

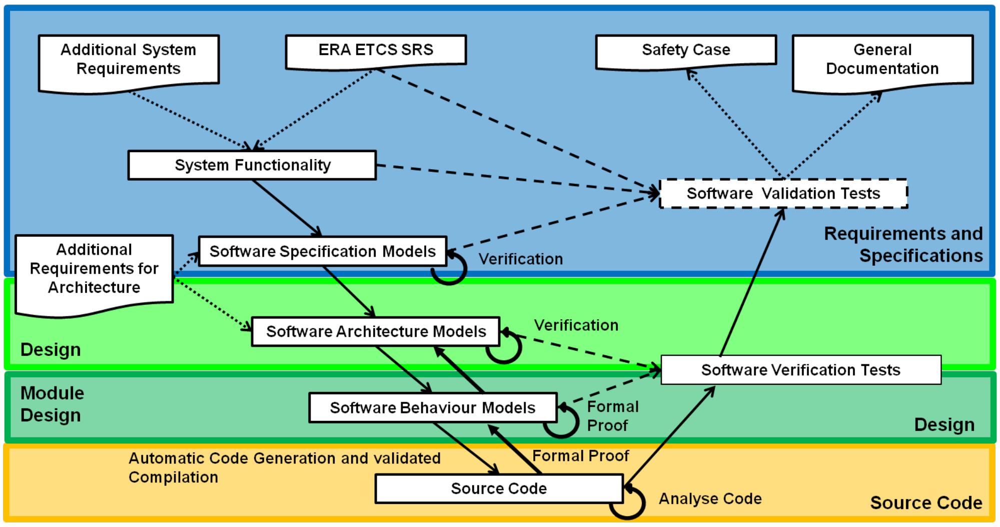

Additional System Requirements
ERA ETCS SRS
Safety Case
General Documentation
System Functionality
Software Validation Tests
Requirements and Specifications
Additional Requirements for Architecture
Software Specification Models
Verification
Software Architecture Models
Verification
Design
Software Verification Tests
Module Design
Software Behaviour Models
Formal Proof
Design
Automatic Code Generation and validated Compilation
Formal Proof
Source Code
Analyse Code
Source Code

Figure 5. Overall openETCS software development process

Figure 5 shows the general artifacts for the openETCS Software development mapped to the EN 50128 V model development life cycle. As the Verification and Validation activities are specific to the design artifacts further details are not available at this point. The Verification and Validation plan describes a number of methods which can be applied for different properties in the context of the openETCS development principals.

Corresponding to the development artifacts the openETCS development process has 5 main distinguishable phase show in table 1. Every phase besides the validation phase is mostly supported by one means of description and one corresponding tool, but the overall traceability of the development requires linking requirements and design elements between all phases.

Table 1. Phases openETCS software development process

Phase	Name	Description	Main Tool component
P1	Software Requirement Phase	Requirement input documents converted to a ReqIF format and informally analyses. The analysis specifies relationships between requirements and revises parts of the requirements to obtain a detail and atomic abstraction level usable for moralization. To support thus the requirements are categorized and grouped.	ProR
P2	Software Architecture Modeling Phase	Building a SysML based on the informal analysis using the categorization of requirements. The architecture model focuses on functional blocks and data flows.	SysML Papyrus
P3	Software Behavior Modeling Phase	Building the SCADE model for the separated basic functional blocks. The SCADE model describes the detailed behavior for the function using the data flows.	SCADE Suite
P4	Code Generation Phase	Based on the SCADE Behavior Model C code will be automatically generated. This C code is than compiled to executable code which runs on the EVC.	SCADE Suite + C Code compiler
P5	Formal Validation phase	Using test models and model checking techniques, to validate the correct model behavior.	Various tools

Although the phase represent different working steps in the life cycle according to EN 50128 the agile development process applied in openETCS induces that the phases are performed iterative during the development process. Respectively, the traceability linking in combination with the verification methods has to ensure that changes in one phase of the development are adopted for all linked artifacts up- and downwards in the development process.

3.2 Interfaces to Safety Activities

As safety is a system property the software development has to be performed according to the safety functionalities allocated to the software. Thereby, the resulting Software SIL defines the combination of quality methods to be performed to ensure that all requirements including the allocated system safety requirements are implemented correctly. To provide all needed justification that the required Software SIL is reached, the development has provide complete documentation for the safety case. As the openETCS software development is not part of a specific system development, but has the objective to specify and implement the overall ETCS functionality, it has to interact with a respective generic safety management process and take the overall safety requirements for a train control system into account. As show in figure 6 the over all development is determined by a number of interactions between design, verification and validation and the general quality and safety management.

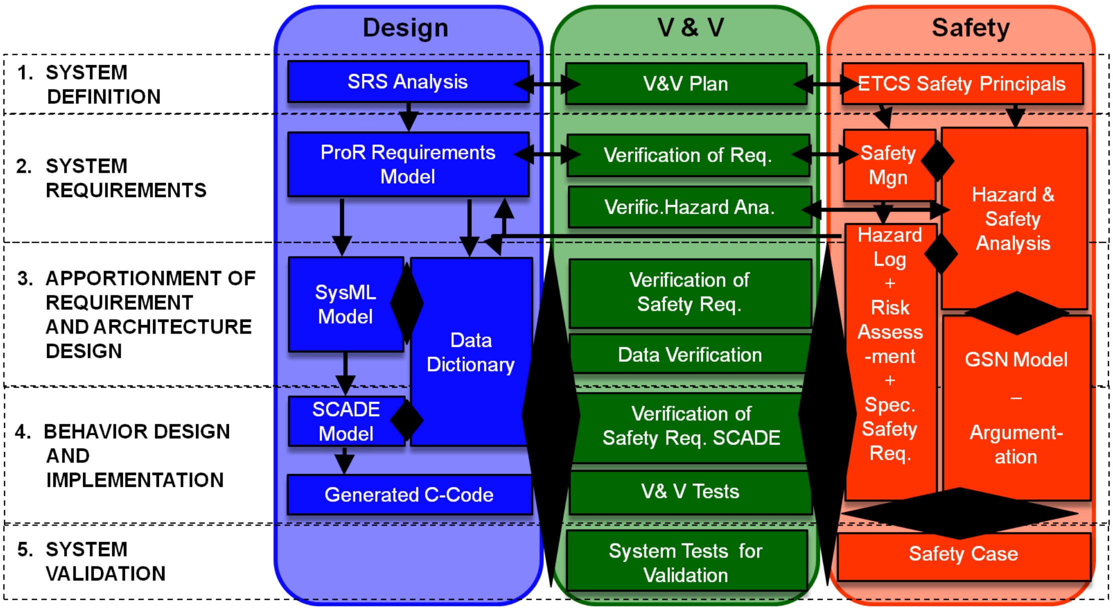


Figure 6. OpenETCS development relations between design, verification and validation and safety activities

3.2.1 Interface to Quality and Safety Management

In parallel to the design process the safety management process has to perform hazard and risk analysis methods for the software design to ensure that the design concept can reach the expected safety integrity level for the overall system. Thereby, potential hazards resulting from the software functionality and its design have to be identified and their risk has to be assess in the system context. To control the risk respective measures have to be taken, which be realized through additional safety requirements to which the design has to be conform. The hazard log represents the central element of this safety management as it collects the hazards and manages the respective risk control.

As the Software SIL determines the development methods which have to be applied to ensure a correct development, the quality management has to ensure that these methods are chosen according to the standard and are applied correctly. This has to be specified in the Quality Assurance Plan and to be documented in the overall safety case.

3.2.2 Specific Interfaces to the Design Process

To be able to respect the system safety requirement these have to be apportioned to the different software components. This has to be done for all abstraction levels during the during the system design.

3.2.3 Interface to Verification and Validation

The verification activities have to ensure that all system safety requirements have been apportioned and implemented correctly at the respective software component.

The validation activities have to ensure that the overall software implementation matches the actual requirements. Therefore, it has to be ensured specifically that the Software in it system context respects the overall system safety requirements.

The verification and validation plan do specify the appropriate methods used. The verification and validation reports as evidence for the qualified software development become part of the safety case.

4 Hazard and Risk Analysis

The safety design process and the resulting documentation constitute the main documents for the system approval, as it is required by European and national law to do everything reasonable expectable to prevent harm. Accordingly the CENELEC standards build the common technical rules for the development process. The Common Safety Methods present a concept based on the EN50126 how the risk evaluation and management has to be performed.

Therefore the main references concerning the safety design process are the CENELEC standards, mainly the EN50126 on how the safety aspects have to be handled as part of the RAMS management over the development process. The overall risk evaluation concept is also defined at this point. The specific concerning the safety case preparations are defined in the EN50129 including the Safety Integrity Level concept.

Since the ETCS system specification for the on-board unit shall be formalized and implemented in the software during the openETCS software development, the specific safety requirements for the on-board unit have to been determined according to the overall system requirements. These are mainly given by the following two parts of the CCS TSI:

- UNISIG SUBSET-026 System Requirements Specification (Version 3.3.0)
- UNISIG SUBSET-091 Safety Requirements for the Technical Interoperability of ETCS in Levels 1 and 2 (Version 3.2.0)

In relation to SUBSET-91 further documents should be take into account to derive specify safety aspect:

- Part of TSI Annex A
  - SUBSET-036
  - SUBSET-037
  - SUBSET-040
  - SUBSET-041
  - SUBSET-098
- Not part of TSI Annex A
  - SUBSET-039
  - SUBSET-078
  - SUBSET-079
  - SUBSET-080
  - SUBSET-081
  - SUBSET-088

Based on these overall safety goals as specific hazard and risk analysis for the software subsystem has to be performed to allocate the requirements and set-up the openETCS hazard log as shown in figure 6. Cased on the principal software architecture the risk can be assess, which then leads to the definition of risk control measures. These are the basis for specific safety requirements for the on-board unit software design and its verification and validation. During the development these requirements are adopted if necessary for the different abstraction levels from the high level model down to the source code. The safety case, which shall be supported by a GSN model as presented in chapter refsafetycase has to present all needed documentation to show that these leads to the required risk level.

4.1 General ETCS Safety Principals

Based on the ETCS reference architecture Subset 91 gives the role of ETCS as train protection as the following:

"To provide the Driver with information to allow him to drive the train safely and to enforce respect of this information, to the extent advised to ETCS."

Respectively, the Core Hazard for the ETCS reference architecture is defined as the following in Subset 91:

"Exceedance of the safe speed or distance as advised to ETCS."

Based on the role of ETCS and its respective SIL 4 quantification the maximum allowed rate of occurrence (Tolerable hazard rate) of the ETCS Core Hazard for ETCS on-board is

$$1.0 \times 10^{-9} hour^{-1} train^{-1}.$$

The same value is specified for the corresponding track-side.

Adapted form the ETCS system safety analysis presented in Subset 88 the Annex A of Subset 91 presents the List of Hazardous Events inside ETCS that might cause the ETCS Core Hazard to occur, either alone or in combination with other failures. These are the events not eliminated by the operational analysis. 34 of these hazardous events are allocated to the Kernel, which make them the basis for the on-board software hazard and risk analysis.

4.2 Proof of concept

To evaluate the hazard and risk analysis process in the openETCS development environment a proof of concept has been performed during the VnV Level 1 activities in WP 4 by project partners Systerel, All4Tec and AEbt. As the respective design artifact for the proof of concept a SysML modeled formalizing one part of Subset 26 has been chosen:

"3.5 Management of Radio Communication (MoRC)."

This model has been created by the project partner Siemens during the design process first in SCADE and then extended to a SysML model. The Model can be found under `https://github.`

com/openETCS/model-evaluation/tree/master/model/SCADE_Siemens/MoRC_System/MoRC_System/SysML_Model. For the Proof of Concept only the SysML model has been used as the scope was on the hazard and risk analysis for the SysML architecture allocation.

To do so the following hazardous event given in Subset 91 has been identified as the event mainly related to this part of the specification:

“KERNEL-6 Manage communication session failure” which results an “RADIO (INFILL) Transmission data consistency failure (safety related transmission function)”.

The following task have been performed during the Proof of Concept as examples for the activities during a hazard and risk analysis.

1. Conformity comparison for the MoRC
   - Conformity comparison regarding sub functions between Subset 26 and SysML model
   - Conformity comparison regarding Input Output data between Subset 26 and SysML model
2. Safety Analysis for the MoRC
   - Hazard & Risk analysis for each sub functions
   - Derivation of safety Target (SIL) for each sub functions
   - Derivation of safety requirements for each sub functions
3. Safety Assessment for the MoRC
   - Derivation of the safety requirements
   - Assessment of the model against these safety requirements

The first task conformity comparison has been performed to verify the model, as no formal verification process has already been performed on the chosen preliminary model. The conformity was passed, as all sub function of the subset 026 function (MoRC) and those of the model match together and all Input & Output data for each sub function of the subset 026 function and those of the model also match together.

The second task Safety Analysis was performed using methods for three different safety activities as defined in the EN 50129 standard: hazard identification, risk estimation and evaluation, derivation of the Safety requirements. In general hazard identification and risk analysis are performed using traditional approaches like FMECA or HAZOP. In the case of openETCS since models are available a model-based, computer supported approach (based on the Tool “Safety Architect” from the company ALL4TEC) for hazard identification has been used in addition. Both results have been compared at the end to validate the results.

For functions coming from the Subset 026 a hazard and rick analysis has been already performed, including a SIL allocation as presented in Subset 091. All these functions have a SIL 4 safety requirement; therefore sub function from these functions shall have also a SIL 4 safety requirement. For functions and corresponding sub function coming from others sources than the Subset 26 a SIL allocation on the basis of techniques described in the standards EN 50126, 50129 or IEC 61508-5 shall be performed.

The third Task of the process consists in performing a Safety Assessment, which implies derivation of safety requirements and the following assessment of the model against these safety requirements. The derivation of the Safety requirements required knowledge in the function being analyzed, the interfaced components, the ETCS in general and CENELEC standard.

The details results for the Proof of Concept can be found at `https://github.com/openETCS/validation/tree/master/VnVUserStories/VnVUserStoryAll4Tec-AEbt`.

Overall the Proof of Concept has confirmed the correctness of the model as far as this has been possible in the limited scope.

The FMEA has identified 23 hazardous events in the subsystem which could lead to the Kernel 6 event. As measure to prevent this occurrence of these events 18 safety criteria have been derived.

The model-based safety analysis showed 12 safety assumptions related to over 40 granular function events. These are related to the assumed unwanted behavior in different subfunction blocks. Overall both methods seam to create comparable results, but due to the status of the model and the complexity of the mathematical model-based analysis both results are difficult to match. Respectively, the model-based analysis has to be specified further to fit the modeling process. But due to the size of Subset 26 on-board function, the model-based analysis is in the focus to lower the amount of manual work.

As the proof of concept has clear shown that the process is able to derive Safety requirements, which can be allocated to different specification based on the software architecture, the results can be used to define the openETCS hazard and risk analysis process.

4.3 Safety design process supporting tools

Supporting software tools are needed to handle the safety artifacts and to some degree to more efficiently perform the safety design activities. As some safety artifacts like the safety requirement specifications and the safety backlogs are closely related to design artifacts the same tools can be used. Especially all requirements should be handled by one tool to ensure full traceability and provide one main interface for the verification and validation activities.

Depending on the methods used for hazard and risk analysis appropriate tools are needed to perform the analysis, collect the hazards and associated risks in the hazard log and to evaluated possible risk control measures. Thereby, traceability has to be guaranteed between all activities. As the main architecture will be designed in SysML, safety analysis tools like the Safety Architect Tool will be used to analyze the hazard propagation in the functional decomposition and to derive the resulting risk level. In this way short feedback iterations to the design process can be realized.

5 OpenETCS Safety Case

If the openETCS specification model and the generated on-board software shall be used in operation sufficient documentation is needed to demonstrate the used quality and safety management and their methods according to CENELEC standards, so that all potential implementers know which steps have to be performed by them to provide a safe product. One of the resulting objectives of the openETCS project is to identify an efficient way for the safety case process and to develop improvement strategies to reach certification at various National Safety Authorities thus reducing time and money in industry for the ETCS development by avoiding unnecessary or redundant procedures.

Since the safety plan and safety case provide the basis for the safety approval the tools used to generated these artifacts should help to generate a consistent argumentation and efficiently collect the data needed to provide evidence. Respectively, interfaces to manage documents and automatically generate reports would be helpful functionalities.

5.1 Structure

Starting from a set of requirements, the strategy to demonstrate the safety of a product is to be developed and graphically described (see figure 1). In general, the fulfilment of each requirement will be shown by a tree of argumentation. The leaves of these trees specify the corresponding evidences (e.g. test results or analysis results). These evidences have to be documented and the corresponding documents accrue during the corresponding phases of the CENELEC development process described in the EN 50126. It turned out that such graphical argumentation structures ease the discussions with legal authorities as they understand the essence of the argumentation strategy in a very short time. In addition, through referencing the corresponding documents in the leaves of these trees, information retrieval is strongly supported.

5.1.1 Quality Management Evidence

5.1.2 Safety Management Evidence

5.1.3 Functional and Technical Safety Evidence

5.2 Model-based Argumentation

5.2.1 Goal Structured Notation

5.2.2 ACedit

6 Conclusion

References