openETCS

OETCS/WP2/D2.3 – 01/01

ITEA2 Project
Call 6 11025
2012 – 2015

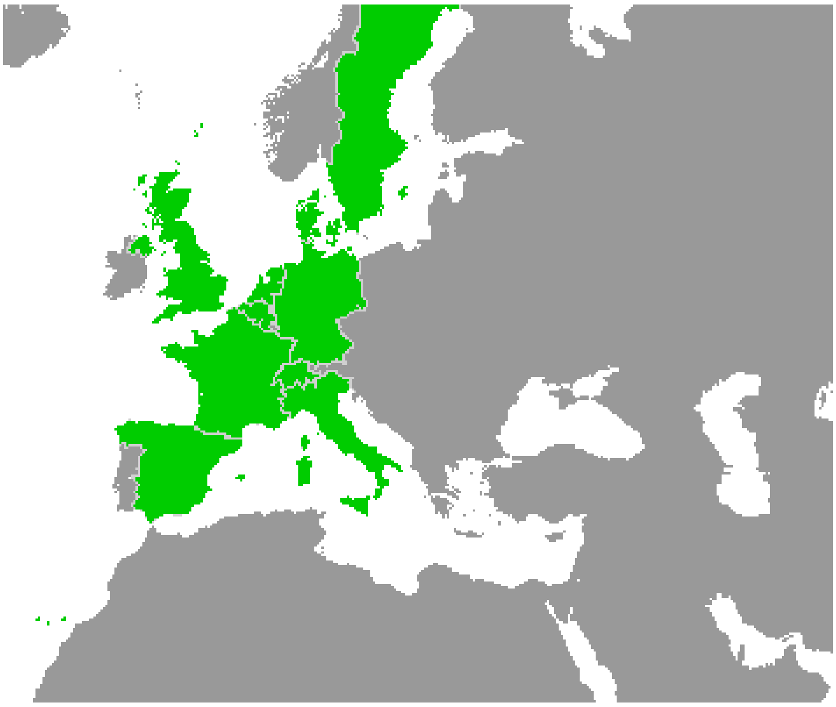
Work-Package 2: “Definition”

OpenETCS process

Definition of the overall process for the formal description of ETCS and the rail system it works in

Marielle Petit-Doche and Matthias Güdemann

April 2013



This page is intentionally left blank

Work-Package 2: “Definition”

OETCS/WP2/D2.3 – 01/01
April 2013

OpenETCS process

Definition of the overall process for the formal description of ETCS and the rail system it works in

Marielle Petit-Doche

Systerel

Matthias Güdemann

Systerel

Definition

Prepared for ITEA2 openETCS consortium
Europa

Abstract: This document gives a description of the process to be applied in the OpenETCS project. In the first part, the document gives a description of the specification and design activities for a critical system. The second part presents an abstract description of the case study issued from SUBSET-026.

Table of Contents

Figures and Tables

Figures

Tables

Document information	
Work Package	WP2
Deliverable ID or doc. ref.	D2.3
Document title	Definition of the overall process for the formal description of ETCS and the rail system it works in
Document version	01.01
Document authors (org.)	Marielle Petit-Doche (Systerel) Matthias Güdemann (Systerel)

Review information	
Last version reviewed	00.01.00
Main reviewers	S. Baro (SNCF)

Approbation			
	Name	Role	Date
Written by	Marielle Petit-Doche	WP2-D2.3 Sub-Task Leader	
Approved by	Gilles Dalmas	WP2 leader	

Document evolution			
00.01.00	01/03/2013	M. Petit-Doche	Document creation
Version	Date	Author(s)	Justification
01.01.00	30/04/2013	M. Petit-Doche	Description of the process according Paris and Charleroi meeting Review comments on 00.01

1 Introduction

The purpose of this document is to describe the specification and design activities for the OpenETCS project. The activities for safety, verification and validation are not in the scope of this document and will be described in WP4's documents.

To deal with a safety process, the specification and design activities shall follow the requirements of EN 50126, EN 50128 and EN 50129 and reflect usual activities for the development of railway critical systems (see D2.1 and D2.2). This description is linked to the set of requirements defined for the OpenETCS project in D2.6.

1.1 Motivation

This document describes the process to be applied during the OpenETCS project to achieve the main goals of the OpenETCS project:

A semi-formal reference specification for the ETCS requirements and architecture, completed by strictly formal models of sub-parts

The first goal of the project is to propose a semi-formal specification of the ETCS on-board functionalities according to UNISIG SUBSET-026, baseline 3.

The purpose of this model is:

- to enhance the understanding of the subset;
- to be able to animate the model for testing and analysing purpose at system level;
- to provide information on the completeness and soundness of the SUBSET-026;
- to be used as a reference semi-formal specification for the implementation of an on-board unit (by the OpenETCS project team and by industrial actors);

The output is a model, at least semi-formal, understandable by many formal approaches (SCADE, Simulink, B tools, OpenETCS tool chain. . .) that can be given to all railway actors, and if possible associated to SRS documents in the ERA database.

Thus, strictly formal models can be designed from this semi-formal model which allows for formal proofs of sub-parts of SUBSET-026. This will allow improving the understanding of the system, and will provide elements for verification and validation using formal proof.

The final goal is that industrial actors work with this model instead of the natural language specification. The objective is to cover as much as possible of the functionality of the on-board unit described in SUBSET-026 and to show the capabilities of analyses of a complex system using formal approaches.

Define the safety case concept for the full model and apply it on a subset of the on-board unit

The safety strategy and the safety case concept required for the full validation of the product, compliant to the CENELEC standards shall be taken into account in all steps of the specification and design process. This will allow industrial actors to reuse the models and processes to develop certifiable products.

In particular the definition of the process shall take into account specification as well as verification and validation of the safety properties on the models. The outputs of WP4 (safety plan, safety case concept, verification plan and validation plan) will complete the description of the safety process.

Provide a tool chain and process/methodologies for developing an on-board software that can fulfil the CENELEC requirements for SIL4 software

The design process of the system and the associated tools of the tool chain, shall be suitable to provide a certifiable product. For this purpose all steps of the process and the choice of the methods and tools shall be justified to ensure a safe approach to build an ETCS system.

The full safety process required to make OpenETCS *certifiable* according to CENELEC 50126, 50128 and 50129 shall be described in detail. The safety process will detail precisely which activities are required, why they are required, and the choices that are made to claim that a safe design process is guaranteed.

The use of formal methods, supported by tools, is highly recommended in this safety process for specification, design, verification and validation of the certifiable product.

The tool chain should include model editors, code generators, verification tools (including formal provers), validation tools (including test generators, simulators,..), document generation, version management, maintenance facilities, . . .

Provide an executable software package generated from the specification of on-board ETCS

An executable software of the specification shall be provided, as well as a non vital implementation of the on-board unit for laboratory test, simulation and as reference.

The output is the result of a functional implementation for the ETCS requirements and architecture which can be used by the industrial as reference.

%% Does this executable take into account real-time constraints ? API ? from which model it is derived ? %%
%%at which level of the V-cycle ? system ? software ?%%

Besides this executable software, the tool chain can be used to generate certifiable executable software from the formal model of sub-parts of the on-board unit.

1.2 Contents of this Document

As the Quality Plan D1.3 focuses on means to apply during the OpenETCS project (for example open source approaches or Scrum organization) the aim of this document is to define the main steps which are necessary within the OpenETCS project to produce a certifiable system according to the CENELEC standards. Safety, verification and validation activities are described in the outputs of WP4.

%%Give the references of WP4 deliverables%%

The first part of this document focuses on the description of the mandatory steps of a life-cycle to design a critical system according to the CENELEC standards, as described in figure 1.

The proposed process for the OpenETCS project shall describe:

- how to design a semi-formal model of the on-board unit system from the SRS SUBSET-026 ;
- how to design some subsets of the SRS SUBSET-026 within a safety process;
- how to produce a running model of the application software of the on-board unit.

For the 2 first objectives, the semi-formal model shall take into account the safety constraints to apply to the design of a critical railway system.

The second part of this document describes the system to design during the OpenETCS project, as well as the scope of the safety activities on this system.

2 Reference Documents

- CENELEC EN 50126-1 — 01/2000 — *Railways applications — The specification and demonstration of Reliability, Availability, Maintainability and Safety (RAMS) — Part 1: Basic requirements and generic process*
- CENELEC EN 50128 — 10/2011 — *Railway applications – Communication, signalling and processing systems – Software for railway control and protection systems*

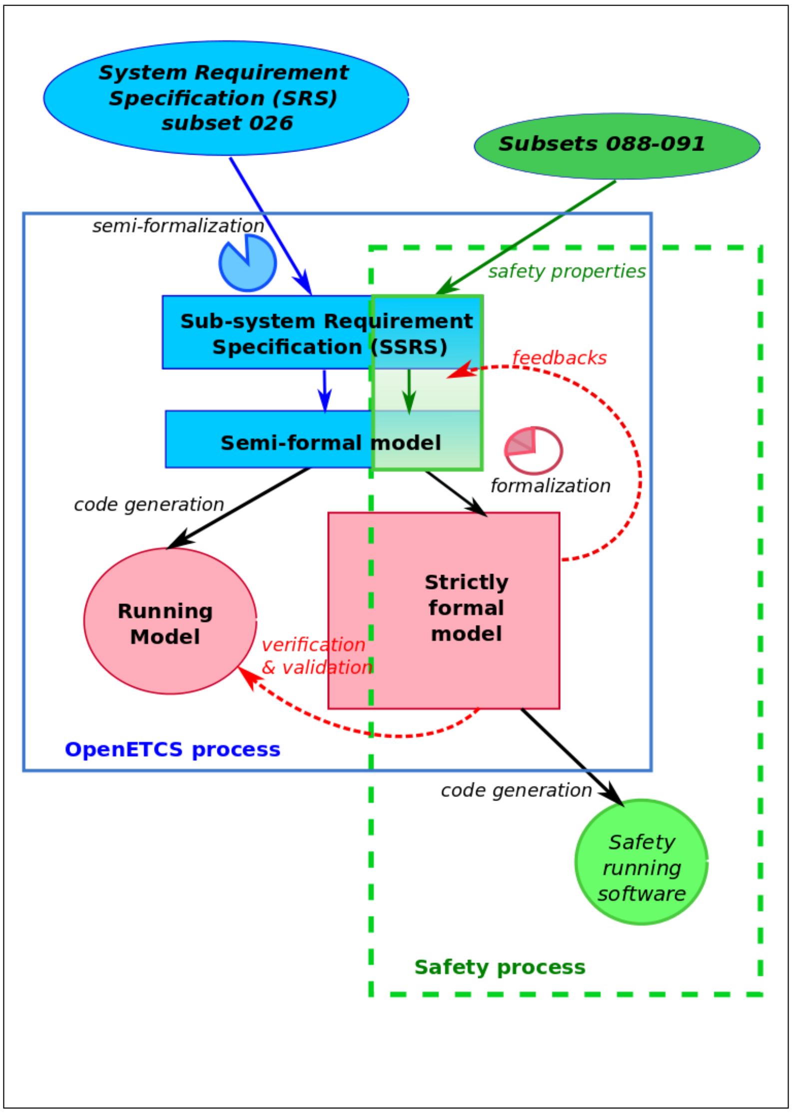


Figure 1. OpenETCS process

- CENELEC EN 50129 — 05/2003 — *Railway applications — Communication, signalling and processing systems — Safety related electronic systems for signalling*
- FPP — *Project Outline Full Project Proposal Annex OpenETCS* – v2.2
- SUBSET-026 3.3.0 — *System Requirement Specification*
- SUBSET-076-x 2.3.y — Test related ERTMS documentation
- SUBSET-088 2.3.0 — *ETCS Application Levels 1 & 2 - Safety Analysis*
- SUBSET-091 3.2.0 — *Safety Requirements for the Technical Interoperability of ETCS in Levels 1 & 2*
- CCS TSI — *CCS TSI for HS and CR transeuropean rail has been adopted by a Commission Decision 2012/88/EU on the 25th January 2012*
- D1.3 – Project Quality Assurance Plan
- D2.1 – Report on existing methodologies
- D2.2 – Report on CENELEC standards
- D2.6 – Requirements for OpenETCS

3 Conventions

The requirements are prefixed by "R-zz-x-y", and are written in a roman typeface, where "R" stands for "Requirement", "zz" identifies the source document,"x" is the version number and"y" is the identifier of the requirement. All the text written in italics is not a requirement: it may be a note, an open issue, an explanation of the requirements, or an example.

The placeholder "%%xxx%%" is used to indicate an unfinished paragraph or section which is to be defined or confirmed.

4 Glossary

API Application Programming Interface

FME(C)A Failure Mode Effect (and Criticity) Analysis

FIS Functional Interface Specification

HW Hardware

I/O Input/Output

OBU On-Board Unit

PHA Preliminary Hazard Analysis

QA Quality Analysis

RBC Radio Block Center

RTM RunTime Model

SIL Safety Integrity Level

SRS System Requirement Specification

SSHA Sub-System Hazard Analysis

SSRS Sub-System Requirement Specification

SW Software

THR Tolerable Hazard Rate

V&V Verification & Validation

5 OpenETCS Process

5.1 Overall Description

To pursue the goals given in the introduction, the development cycle for the project is presented in this document.

In order to minimise the number of different models and unused steps, the proposed process shall take into account the safety concepts from the early steps, i.e., the sub-system requirement specification (SSRS) and the semi-formal definition of the model instead of the SRS. Thus, these elements can be used in the safety process to deduce formal models as shown in figure 1.

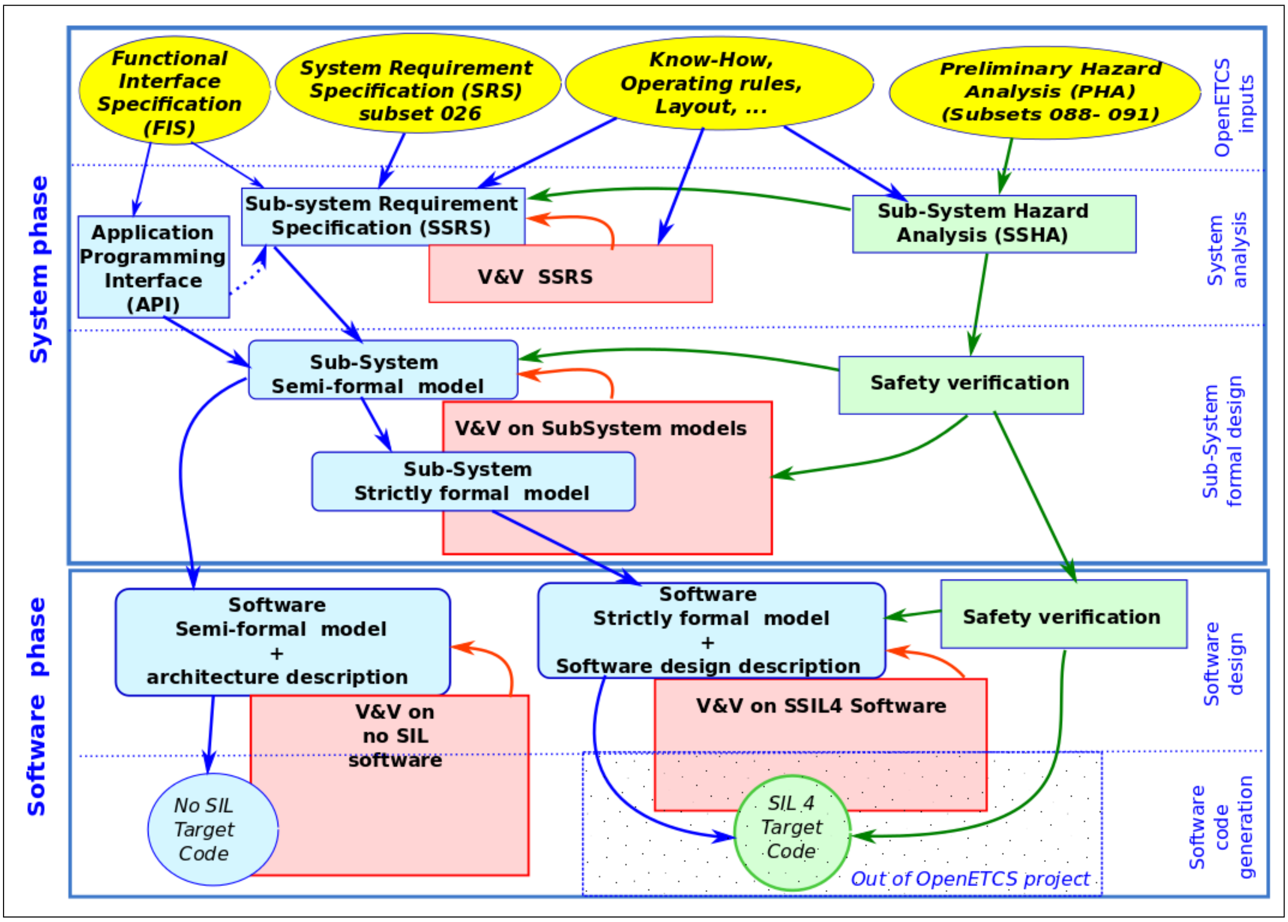


Figure 2. Whole process

The two most important elements of the System life-cycle of EN 50129 and the Software development life-cycle model of EN 50128 are the separation of the life-cycle into well-defined phases and the focus on the production and recording of extensive documentation of the development process. This allows facilitation of safety, verification, validation and assessment activities and confidence in the use of good practises to develop a critical system. To achieve this, an appropriate life-cycle must be defined for OpenETCS, following the constraints provided by the CENELEC standard, and appropriate roles and responsibilities must be assigned to the participants.

Figure 2 described the main phases and main activities of the OpenETCS process. Input elements of the project are in yellow, specification and design activities in blue, verification and validation activities in red, safety activities in green.

Two main phases are defined :

System phase to analyse the input documents and provide a model of the on board unit according SUBSET-026 and safety strategy:

- First system analysis shall provide a *Sub-System Requirement Specification* (SSRS) to define the scope of the system to design and its structure, completed with an abstract *Application Programming Interface* (API) to give the main interfaces of the system and interaction between software and hardware items. *Sub-System Hazard Analyses* (SSHA) allow the definition of safety properties.
- Secondly, a model, at least semi-formal, is designed to describe sub-system architecture, main functions and to allocate sub-system requirements. This model can be completed with a formal model to focus on some functions or properties.

Software phase to design the software and then generate applicative code of the sub-system. Two approaches are developed together from the same sub-system model:

- on one part to complete the semi-formal model to obtain a functional code covering as much as possible of the SSRS
- on the other part to provide methods and tools to obtain a SIL4 code and to apply this approach on a subset of the SSRS.

In the sequel, the main lines of this figure are going to be detailed. However, we are going to focus on specification and design activities only.

The Software Planning phase is defined by WP1 in the Quality Assurance Plan. The Software Test / Validation phase is defined by WP4 in the Validation Plan. The Software Verification activities are defined by WP4 in the Verification Plan.

Safety activities are described in EN50126. The proof that the process satisfies the requirements of the standard is out of the scope of this document and shall be managed by the Safety Case (WP4).

5.2 OpenETCS inputs

%%To Be Defined%%

The main input of the OpenETCS project is the SUBSET-026 3.3.0, which can be considered as the *System Requirement Specification* (see R-WP2/D2.6-01-011).

However, this document is not sufficient to produce a model of the on-board unit software. Thus a *Sub System Requirement Specification* shall be produced during this phase, in particular to define the structure of the system and to manage the requirement allocation (see R-WP2/D2.6-X-12).

External interfaces are partially provided by the UNISIG documents. However interfaces between ETCS units and application programming interfaces (API) have to be defined for the OpenETCS project.

5.3 System Analysis

5.3.1 Objectives

The aim of this phase is to have a clear definition of the sub-system to design, which is not given by the input documents, and to define the scope of the model to design. Thus during this step, we shall identify:

- a set of requirements which describe the functionalities of the sub-system and the expected results concerning performance, maintainability, safety, reliability,...
- the description of the architecture of the sub-system
- the description of the external and internal interfaces of the sub-system.

Safety activities are necessary to define the safety requirements of the system and to define which functions are considered vital or non-vital respectively.

5.3.2 Outputs

According to design activities, two documents shall be produced during this phase :

Sub-System Requirement Specification (SSRS) shall define the scope and the structure of the sub-system and manage the requirement allocation (see R-WP2/D2.6-X-10).

Application Programming Interface (API) shall describe the interfaces of the sub-system.

According the CENELEC standards, these documents complete the inputs document (SRS and FIS) to produce :

- the *System Requirements Specification* which describes all requirements of the system
- the *System Architecture Description and Software / HW interface definition* which specify how the Software and the HW interact as well as the location of the boundary between the two

5.3.3 Detailed Description

The first part is the definition of the *Sub System Requirement Specification* (SSRS) which shall allow:

- to clearly define the scope of SUBSET-026 to take into account for the design (only on-board functionalities are designed, track-side functionalities are out of the scope of the project) (see R-WP2/D2.6-X-10.2.4),
- to define the interfaces of the system: external interfaces and software/hardware interfaces (see R-WP2/D2.6-X-10.2.5, R-WP2/D2.6-X-10.2.6),
- to provide a functional architecture of the system with inputs and outputs of each function identified (see R-WP2/D2.6-X-10.2),
- to allocate SRS requirements to each function (see R-WP2/D2.6-X-10.3),
- to classify Safety versus Non-Safety items (functions, input/output, requirements, ...) from the safety analyses results (see R-WP2/D2.6-X-11),

However, the SSRS shall be compliant with the input documents of TSI (subset 26, FIS, ...) (see R-WP2/D2.6-X-10, R-WP2/D2.6-X-10.2.7 and R-WP2/D2.6-X-10.4)

It shall also facilitate safety, design, verification, validation and maintenance activities: full traceability between SRS and SSRS shall be provided (see R-WP2/D2.6-X-10.5).

Detection of inconsistencies or ambiguities in the input documents shall be discussed and tracked (see R-WP2/D2.6-X-10.6).

These tasks need some interactions with safety activities, for example to define safety tags on functions, requirements,...

The second part is the definition of the *Application Programming Interface* (API)....

%%To Be Defined%%

5.3.4 Means and tools

The SSRS and API shall be described as documents. However these documents shall be completed by a semi-formal model to describe the functional architecture of the on-board unit (see R-WP2/D2.6-X-10.2.2):

- to define the scope of the application to design (see R-WP2/D2.6-X-10.2.4),
- to split the main function of the system into independent functions (see R-WP2/D2.6-X-10.2.1),
- to describe the data flow between functions (see R-WP2/D2.6-X-10.2.3),
- to describe the abstract interfaces of the sub-system and its environment, with respect to the existing input documents (see R-WP2/D2.6-X-10.2.5 and R-WP2/D2.6-X-10.2.6),
- to support allocation of the requirement to the function and data (see R-WP2/D2.6-X-10.3).

The requirements from the SRS are allocated toward the functions of the SSRS (the architecture), possibly split and rewritten in order to restrict their scope to these functions. They are also rewritten in order to match the objects named in the architecture (in particular internal and external I/O). The requirements are provided in natural language (even if the objects are unambiguously named).

In view of verification activities, traceability between SSRS and SRS shall be provided (see R-WP2/D2.6-X-10.5). In practice, interpretations, additions and omissions of requirements shall be tracked and justified (see R-WP2/D2.6-X-10.5.1 and R-WP2/D2.6-X-10.5.3), as well as exported requirements (see R-WP2/D2.6-X-10.5.2).

According to CENELEC standards, no specific constraints are given on the tools used during this step: textual and graphical editors, with syntax checker, are needed to produce documents and model. Tools classified as T1 according EN50128 can be used.

5.4 Sub-System formal design

5.4.1 Objectives

The aim of this phase is to provide a model of the sub-system from the SSRS:

- to provide a semi-formal reference specification of the sub-system requirements
- to lift ambiguities
- to detect errors and inconsistencies.

5.4.2 Outputs

The main output of this step is a *semi-formal model of the sub-system* covering the architecture, interface description and requirement allocation of the SSRS.

This semi-formal model can be completed with *strictly formal models* to improve the understanding of the sub-system and to provide elements for verification and validation activities.

5.4.3 Detailed Description

To cover the OpenETCS project objective of formal models, a semi-formal model of the system specification is defined from the SSRS (see R-WP2/D2.6-X-12). This model shall reflect the architecture defined in SSRS (see R-WP2/D2.6-X-10.2.2). Requirements of the sub-system can be refined in a semi-formal means but the semi-formal model shall be as consistent as possible with the SSRS level of abstraction (see R-WP2/D2.6-X-12.2), in particular choices concerning software architecture and design have not to be described at this level. In practice, all the requirements of SSRS (see R-WP2/D2.6-X-12.2.1) and of the sub-system Hazard analysis (see R-WP2/D2.6-X-12.2.2) shall be covered by the semi-formal model.

Traceability between semi-formal model and SSRS shall be provided (see R-WP2/D2.6-X-12.2.5): interpretations, additions and omissions of requirements shall be tracked and justified (see R-WP2/D2.6-X-12.2.3), as well as exported requirements (see R-WP2/D2.6-X-12.2.6).

This semi-formal model can be completed with strictly formal models to improve the understanding of some part of the sub-system (see R-WP2/D2.6-X-14) and to provide elements for verification and validation activities especially concerning safety properties.

To facilitate safety activities, the safety relevant function should be as much as possible insulated from non safety relevant functions (see R-WP2/D2.6-X-17).

5.4.4 Means and tools

The means of description of the semi-formal model shall be understandable by domain experts (see R-WP2/D2.6-X-27), providing graphical description (see R-WP2/D2.6-X-27.1).

The semi-formal model shall reflect the functional architecture defined in SSRS. In particular the language used to design the semi-formal model shall allow it to be modular and extensible (see R-WP2/D2.6-X-15).

In view of validation activities, the means of description of the semi-formal model shall allow to execute or simulate it (see R-WP2/D2.6-X-33).

Parts of the subsystem shall be modelled strictly formally (see R-WP2/D2.6-X-14). This formal model shall be derived from the semi-formal one (see R-WP2/D2.6-X-14.2), as straightforward and automated as possible (see R-WP2/D2.6-X-14.4). Thus, the semi-formal model shall be designed (language and structure) in order to allow the design and validation of the strictly formal model (see R-WP2/D2.6-X-14.3); and to be easily translatable to other languages (see R-WP2/D2.6-X-30). As for semi-formal model, strictly formal model shall be modular and extensible (see R-WP2/D2.6-X-16) and shall refine the modular design of the semi-formal model (see R-WP2/D2.6-X-16.1).

The expressiveness of the language used to design the semi-formal and formal models shall allow formalisation of the classical objects used in the description of a critical system (see R-WP2/D2.6-X-31 and R-WP2/D2.6-X-32):

- state machines
- time-outs
- truth tables
- arithmetics

- braking curves
- logical statements
- messages and fields

In view of safety activities, the languages used for the models shall allow a declarative and formal expression of the safety properties (see R-WP2/D2.6-X-28), understandable by domain expert (see R-WP2/D2.6-X-28.1). The modelled safety properties shall be validated on the semi-formal model by test and on the strictly formal model by proof (see R-WP2/D2.6-X-23). Logical properties can be added to simplify the models but shall be validated just as the properties (see R-WP2/D2.6-X-34).

All means of description used shall be standardized or at least documented in detailed (see R-WP2/D2.6-X-29).

According to CENELEC standards, no specific constraints are given on the tools used during this modelling step: textual and graphical editors, with syntax checker, are needed to produce documents and model. Tools classified as T1 according EN50128 can be used.

5.5 Software design

5.5.1 Objectives:

In this phase, the system requirements shall be refined to take into account software constraints.

Two branches appear:

- The aim of the first branch is to design all the functionalities of the SSRS to produce a functional code, without SIL. As much as possible of the requirement of SSRS shall be covered at this phase.
- the aim of the second branch is to provide a method and a toolchain to produce a SIL 4 code. This approach has to be evaluated on a subset of the SSRS requirements.

5.5.2 Functional branch

For this branch, the semi-formal model defined during the system phase, shall be completed and detailed with software constraints, in such a way it is possible to produce an executable code for a given target.

Outputs:

The main output of this step is a semi-formal model which allow to produce an executable code. This model shall be completed by a *Software Architecture and Design Specification*, which describes the software architecture and the design choice.

Detailed Description:

Taking into account the SSRS requirements allocated to the software, a software architecture shall be defined, with a description of all the input/output of the software, as well as a description of the operational modes and behaviour.

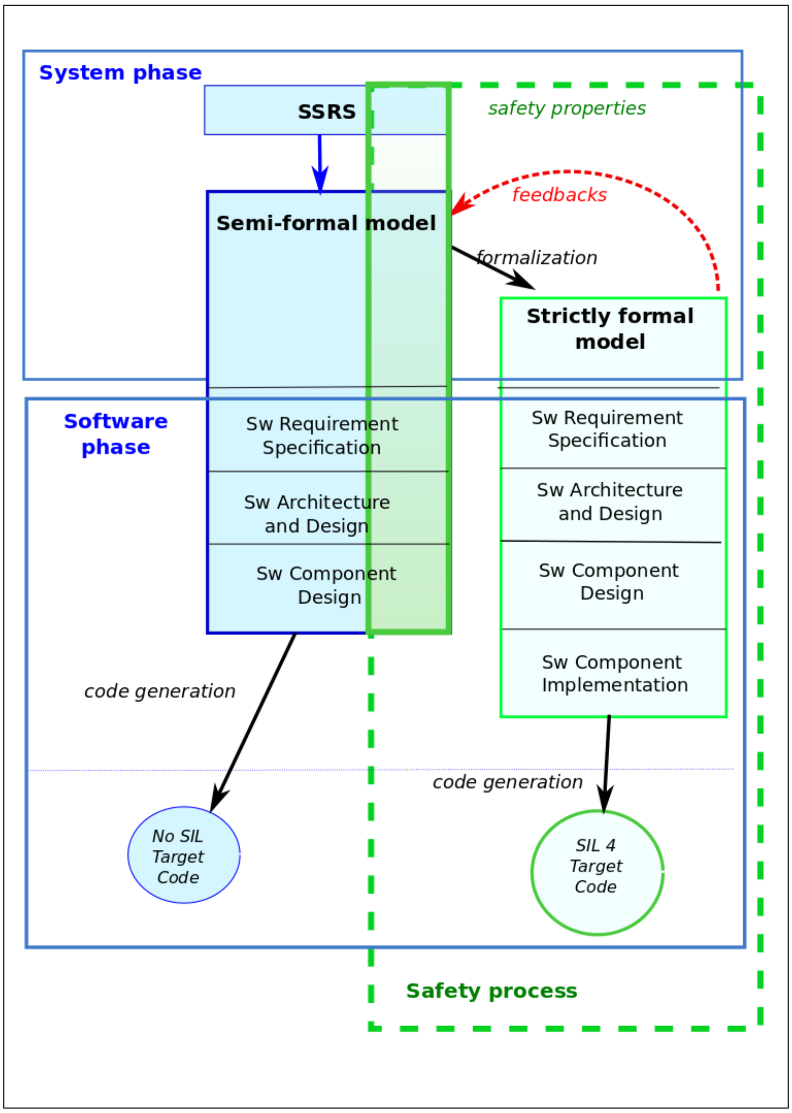


Figure 3. Software phase description

Then, each software component is designed according to the sub-system requirements: All sub-system requirements, allocated to software shall be referenced in the model or a justification shall be given.

It shall provide all the elements to have testable requirements on a target platform. All functions to perform shall be clearly identified. All existing constraints between hardware and software will be taken into account (cf. §7.2.1.1 of EN 50128).

In this phase, there shall also be an evaluation of the Hardware / Software interaction, its influence on the safety aspects of the system and the evaluation of the usage of already existing Software. It shall also ensure the testability and the appropriateness for formal proofs of the resulting Software, in particular by minimising the complexity and the size of the safety relevant parts (cf. §7.3.1.1 to 7.3.1.5 of EN 50128).

Interfaces between software components and the software environment are described in the *Software Interface Specification*.

Means and tools

Comment. To be completed according requirement of EN50128 and D2.6 Use of formal methods to describe. To each level shall be develop the semi-formal model ?

5.5.3 Functional and safety branch

The model shall cover the three software development phases according EN 50128 :

- Software Requirements Phase
- Software Architecture and Design Phase
- Software Component Design Phase

Documents:

From a design point of view, the outputs to produce in this phase are:

- the Software Requirements Specification
- the Software Architecture Specification
- the Software Design Specification
- the Software Interface Specification
- the Software Component Design Specification

Detailed Description:

The first step of this phase is to give an explicit description of software requirements according to system requirements and safety properties. The system architecture model from the preceding phase shall be extended with a model of the software. All software requirements shall be referenced in that model. In particular the *Software Requirement Specification* shall give a description of all the input/output of the software, as well as a description of the operational modes and behaviour. It shall provide all the elements to have testable requirements on a target platform. All functions to perform shall be clearly identified. All existing constraints between hardware and software will be taken into account (cf. §7.2.1.1 of EN 50128).

Then, a *Software Architecture Specification* shall be developed which allows meeting the software requirements and the necessary safety requirements without introducing unnecessary complexity. In this phase, there shall also be an evaluation of the Hardware / Software interaction, its influence on the safety aspects of the system and the evaluation of the usage of already existing Software. It shall also ensure the testability and the appropriateness for formal proofs of the resulting Software, in particular by minimising the complexity and the size of the safety relevant parts (cf. §7.3.1.1 to 7.3.1.5 of EN 50128).

The *Software Design Specification* shall give a description of the design choices, in particular software component decomposition, data description and requirement allocations. Interfaces between software components and the software environment are described in the *Software Interface Specification*.

Finally, the low level specification of each software components is defined in the *Software Component Design Specification* (cf. §7.4.1.1, §7.4.1.2 of EN 50128).

Means and tools

Comment. To be completed according requirement of EN50128 and D2.6 Use of formal methods to describe. To each level shall be develop the semi-formal model ?

5.6 Software code generation

5.6.1 Objectives:

In this phase, the system requirements shall be refined to take into account software constraints.

Two branches appear:

- The aim of the first branch is to design all the functionalities of the SSRS to produce a functional code, without SIL. As much as possible of the requirement of SSRS shall be covered at this phase.
- the aim of the second branch is to provide a method and a toolchain to produce a SIL 4 code. This approach has to be evaluated on a subset of the SSRS requirements.

5.7 Demonstrator

Objectives

Comment. What is the aim of the demonstrator ? to provide an executable model of the on-board unit What does that mean, Which functionalities have to be taken into account ? Which are the interfaces and API ? Does it take into account real-time, performance,... elements or just functional aspects ? What about safety ? What is the target platform ? Non safety code ?

Documents

%%input of the demonstrator : semi-formal model , at which level ?%%

Detailed Description

Open Issue. Manual versus automatic code generation ?

Means and tools

%%To detail according next meeting results%%

5.8 Concrete Code

Objectives

This code cannot be provided by the OpenETCS project : Safety activities are not conducted in the whole scope of the on-board unit subsystem, and elements of the target platform are not provided.

However, the description of how to produce such a code in a safe way is part of the OpenETCS project. This corresponds to the lower phases of the process according to E50128 :

- Software Component Implementation Phase
- Integration

Documents

According to EN50128, outputs of design for this phase are :

- Software Source Code and supporting documentation

Detailed Description

Open Issue. Manual versus automatic code generation ?

%%To detail according next meeting results%%

Means and tools

Comment. Shall be replaced by automatic code generation from formal models using refinement techniques.

6 OpenETCS Case Study

The EVC (European Vital Computer) is the heart of the ERTMS on-board system. This safety relevant computer implements the functions of the SRS subset 026 of UNISIG (for SRS versions beginning with baseline 3, published by ERA) in order to guarantee the safety of the train movements. The OpenETCS scope of application is related only to the EVC part of whole ERTMS system. The track-side part of the ETCS (the Radio Based Control) is excluded from the project activities, it is only considered through its interfaces with the On-Board part of ETCS.

A detailed specification of the sub-system will be given during the system analysis phase: the Sub-System Requirement Specification shall contain

- the high level description of the openETCS case study
- the environment and the architecture of the sub-system to design, with interfaces main functions and data-flow

Safety properties shall be given in the sub-system hazard analysis.