OETCS/WP7/O7.3.2

openETCS

ITEA2 Project
Call 6 11025
2012 – 2015

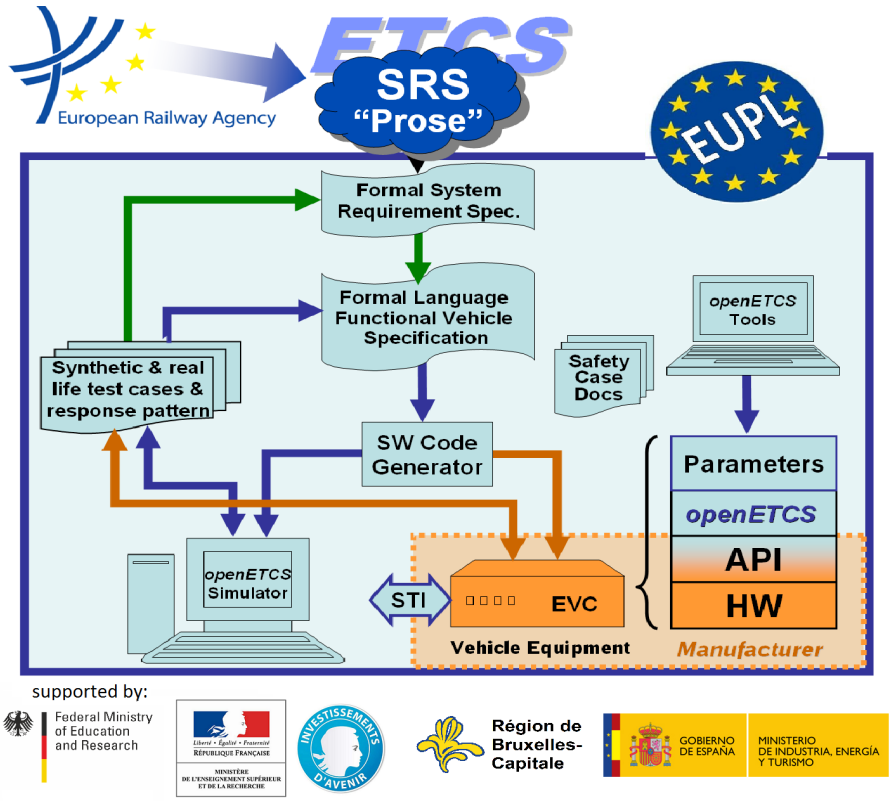
Work-Package 3: “Tool chain”

Tool chain Design Specification

Description of the OpenETCS tool chain implementation

Cecile Braunstein and Jan Peleska

August 2013



This page is intentionally left blank

Work-Package 3: “Tool chain”

OETCS/WP7/O7.3.2
August 2013

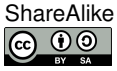
Tool chain Design Specification

Description of the OpenETCS tool chain implementation

Cecile Braunstein and Jan Peleska

University Bremen

Tool chain Architecture



Prepared for ITEA2 openETCS consortium
Europa

Abstract: This document defines the tool chain architecture : the links between the tools and the interoperability mechanism implemented.

The present tool chain will propose some alternative path to give more flexibility with the chosen tools.

Table of Contents

Figures and Tables

Figures

Tables

Document Information

Document information	
Work Package	WP7
Deliverable ID or doc. ref.	O7.3.2
Document title	Tool chain architecture
Document version	00.00
Document authors (org.)	Cécile Braunstein (Uni.Bremen)
	Jan Peleska (Uni. Bremen)

Review information	
Last version reviewed	
Main reviewers	

Approbation			
	Name	Role	Date
Written by	Cécile Braunstein Jan Peleska	WP7-T7.3 Sub-Task Leaders	
Approved by			

Document evolution			
Version	Date	Author(s)	Justification
00.00	25.07.2013	C. Braunstein	Document creation

List of Terms

Notation	Description
API	Application programming inerface.
EMF	Eclipse Modeling Framework.
ERTMS	European rail traffic management system .
EVC	European Vital Computer.
OBU	On Board Unit.
SRS	System Requirement Specification.

1 Introduction and Motivation

Chap resp. CB/JP

1.1 Motivation

1.2 Scope of the document

1.3 Reference documents

[1] Sylvain Baro and Jan Welte. Requirements for openETCS. Requirements D2.6, openETCS, April 2013.

[2] Marielle Petit-Doche and Matthias Güdemann. openETCS process. Definition D2.3, openETCS, February 2013.

[3] Marielle Petit-Doche and WP7 Participants. D7.1: Report on the final choice of the primary toolchain. Primary Toolchain OETCS/WP7/D7.1, openETCS, July 2013.

[4] Oscar Slotosch, Martin Wildmoser, Jan Philipps, Reinhard Jeschull, and Rafael Zalman. ISO 26262-tool chain analysis reduces tool qualification costs. *Automotive 2012*, 2012.

[5] UNISIG. SUBSET-026 – system requirements specification. SRS 3.3.0, ERA, March 2012.

[6] UNISIG. SUBSET-034- train interface FIS. Technical Report 3.0.0, ERA, March 2012.

[7] Anthony I. Wasserman. Tool integration in software engineering environments. In Fred Long, editor, *Software Engineering Environments*, volume 467 of *Lecture Notes in Computer Science*, page 137–149. Springer Berlin Heidelberg, 1990.

2 OpenETCS Tool chain Presentation

Chap resp. CB/JP

2.1 The OpenETCS tool chain

2.1.1 Definition

The tool chain provides the tool support and the development process to provide a formalized specification of SRS and an executable code of the OBU.

The tool chain is composed by two kind of tools :

1. *Development tools*: those used along the phases of the software development process (Requirement engineering, modeling ...).
2. *Management tools*: those used transversely during the complete development process (version management, requirements traceability ...).

These tools are called vertical and horizontal tools in [7]. Figure 1 shows the idea of the complete tool chain integration.

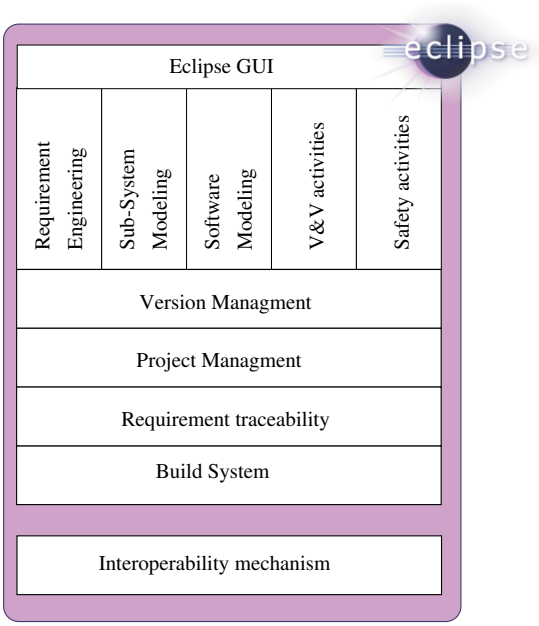


Figure 1. The OpenETCS tool chain

In the next chapters we will give more details on the data integration between the development tools by defining the inputs and outputs artifact of each tools. This document describes the interface between the tool chain activities. The interoperability mechanism will be defined in a separate document. However, the first version of the tool chain will implement a file-based data exchange.

It has been decided (see [3]) that the tool platform hosting the tool chain is Eclipse with the eclipse modeling framework (EMF). This implies that the tool chain will be a set of eclipse plug-ins.

It also implies that we can rely on already available plug-in and features for the versioning, the project management or the build system. The use of EMF will also the software development, by providing a meta model and an API for manipulating EMF components.

2.1.2 The SysML Model

A SysML model of the tool chain will be added to this document. It will allow us to have a formal representation of the tool chain and help to model precisely the different interaction between the development tools as well as the management tools. The SysML model may be also seen as a guideline for integrated new tool: each new tool should be fully described and comply to the defined interfaces Moreover following the idea of Slotosch [4] the model will be the basis to the qualification analysis.

The requirements of WP2 [1] as well as our intern requirement (Appendixes A and B) will be included in the SysML model. Each tools and their connection should then also comply to the requirements list.

2.2 OpenETCS EVC lifecycle

The openETCS lifecycle has been defined in [2] as presented figure 2.

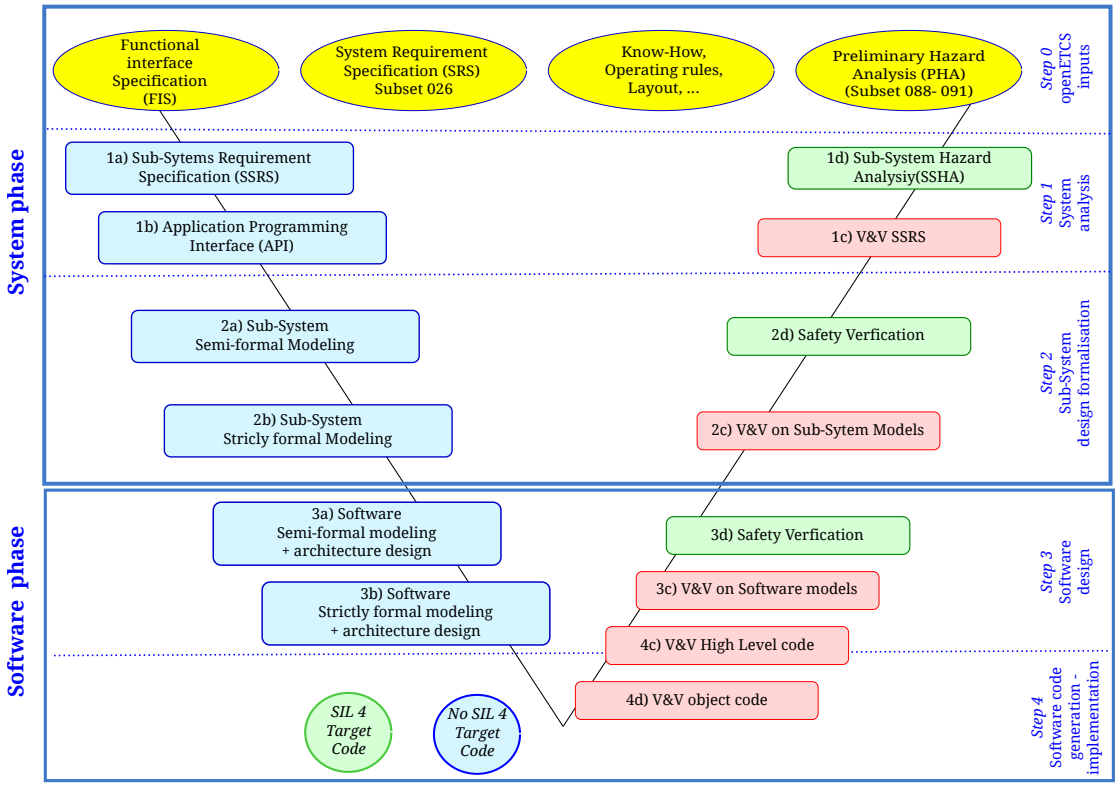


Figure 2. openETCS Process (rough view)

The lifecycle defined all the different steps needed to produce a EVC certifiable SIL4. However in order to define a tool chain we need to define the lifecycle by means of activities (Fig. 3). Each activities may be achieved by one or more tools in the tool chain.

The next chapters will define the limits of each activities, we will show what should be consumed and produced by each activities. Note that lifecycle activities may be realized by more than one tool, thus some activities will be presented with alternative tools.

This vision of the tool chain allowed us to think in term of interface and not about tool only. This will give more flexibility for the implementation of tool chain alternatives.

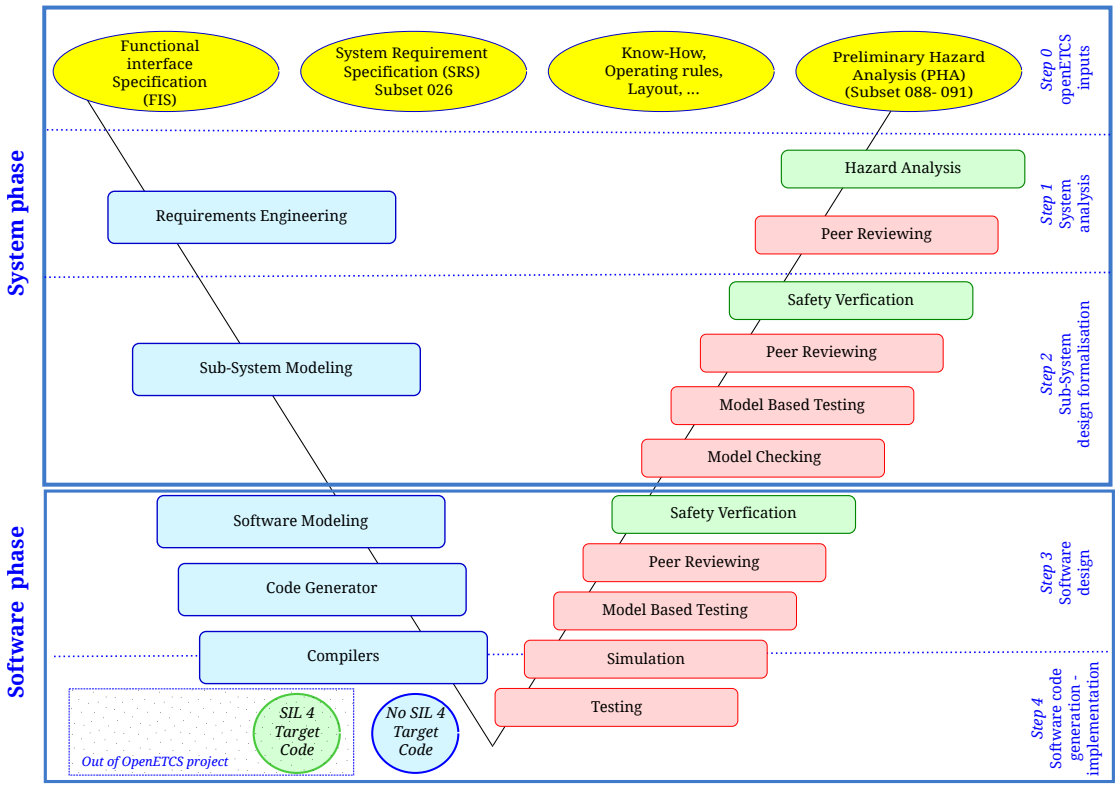


Figure 3. openETCS Process Tool chain activities

add description of each activities

2.3 Management tools

add the list of the management tools and a description

- Version Management
  Version management of the artifacts and the openETCS glue code
- Tool version Management
  Keep track of the compatible and status version of each tools.
- Collaborative work
- Project Management
- Build system
- Non regression test

3 OpenETCS Requirement Engineering

Chap resp. ??

3.1 Activities purpose

Requirement engineering is the starting point of the modeling activities. It is the basis for describing how all the actors and tools may interact to produce the desired system.

In OpenETCS the specification describing the requirement comes from three document, two of them related to the ERTMS [5, 6], the third one to the know-how of the operators and the different actors of the design process. The translation into a requirement database in computer readable format remains a manual task.

Figure 4 presents different tools alternative to realize the requirement management activities within our tool chain. The activity may be achieved by combining some of these tools to get the best of each. For example one can start by collecting the requirements in a csv format and then add the dependency with ProR or Papyrus.

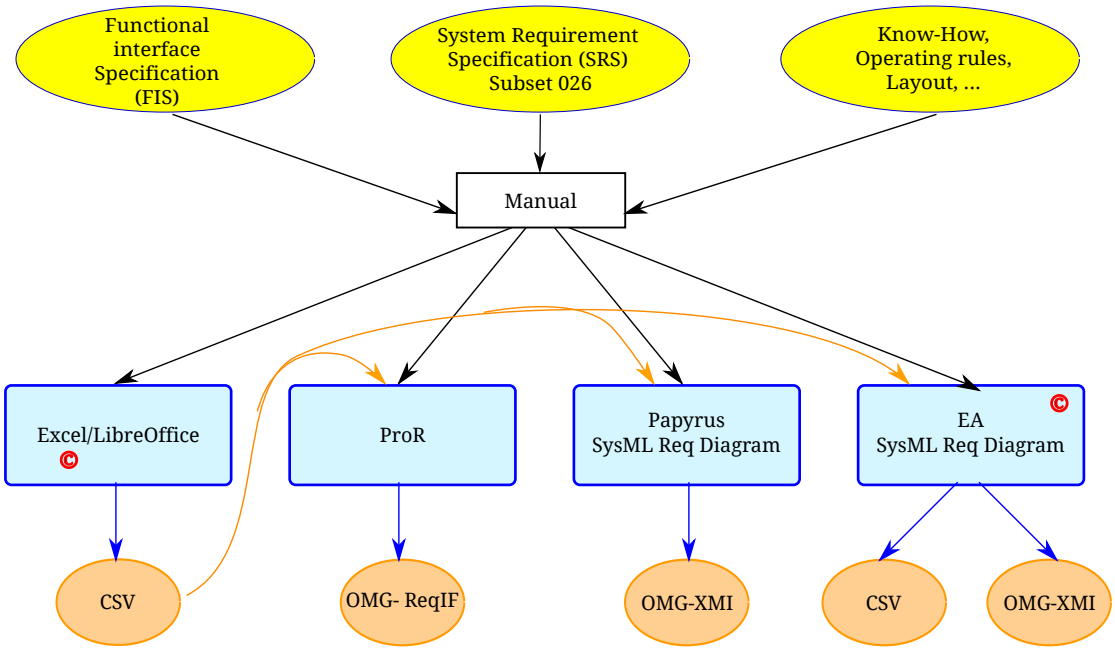


Figure 4. Requirement Engineering

Tools description or reference to their description

3.2 Input/Output artifacts

Take a decision to the interface or exhibits the compatibility between the format and how it may be achieved

4 OpenETCS Sub-System Modeling

Chap resp. ??

add references It has been decided [] to use SysML as a modeling language and papyrus as modeling tool.

The use of SysML is restricted to the following :

- Requirements:
  - Requirement Diagram
  - Requirement relationships : `verify/satisfy/trace`
- Structures:
  - Block Diagram
  - Internal Block Diagram
  - Ports
- Behaviors:
  - State machines
  - Activity diagram (optional)

All tool should be able to understand and to produce the same SysML format.

4.0.1 Input/Output artifacts

5 OpenETCS Software Modeling

Chap resp. ??

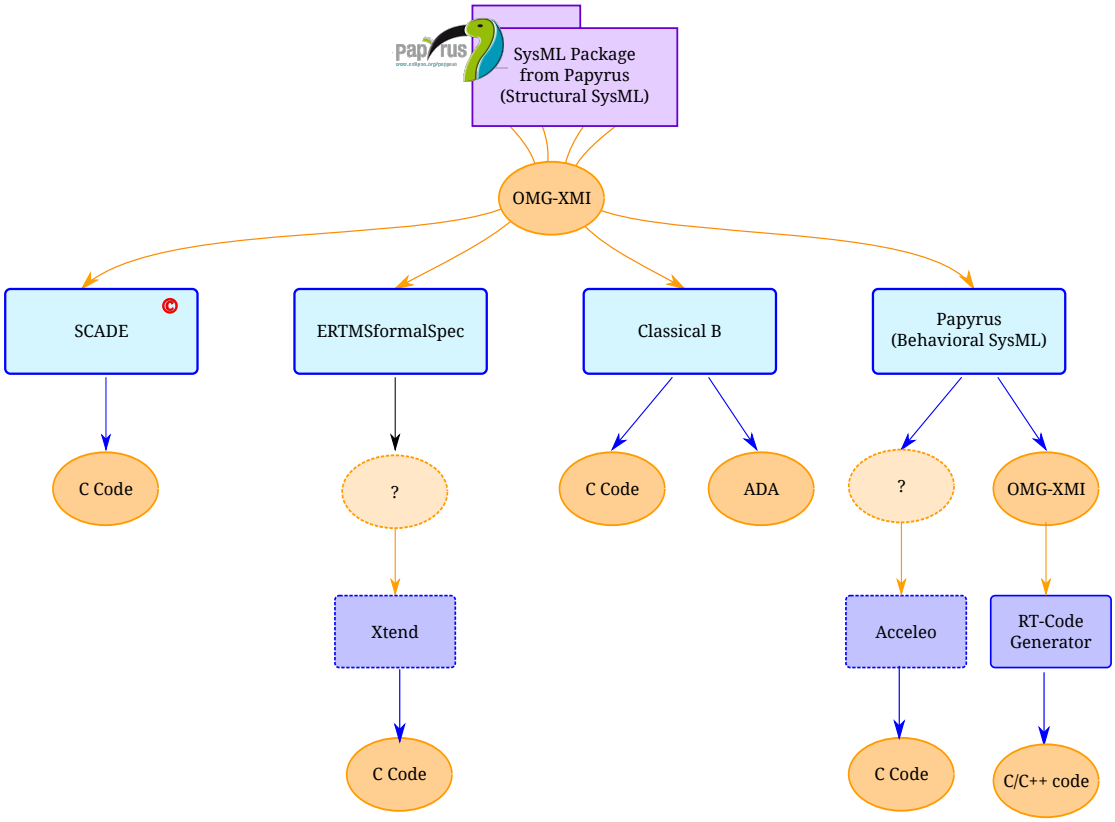


Figure 5. Software phase description

5.0.1 Input/Output artifacts

6 V&V Activities

Chap resp. ??

7 Interoperability specification

Chap resp. ??

To make the tool chain flexible, easy to maintained etc ... We need to define the interface between the tools in a loosely way. The interoperability mechanism should assist the flexibility as well as give some guidance to integrate new tools along the tool chain development.

We propose that for each tool chain activities the WP7 will agree on a common input/output format. This will help to link the activities altogether and ease the tools update or change. It keeps the tools independent.

That implies to have an open interoperability specification. The interoperability mechanism can be decomposed in different concept.

- Architecture: what are the principles of the interoperability mechanism all tools agree on (e.g. RESTful architecture)
- Communication : the communication protocol and data exchange format (e.g. XML)
- Syntax: the structure of information represented in a common data format. What tools need to parse and interpret information (e.g. RDF)
- Semantic layer : Basic semantic to make the tools understand each others. (e.g. Meta model EMF)

The OSLC project proposes a set of standard interface to link the different lifecycle activities together. The idea is to have a common core protocol and architecture and then for each activities a domain specific interface. All activities do not necessarily share the same interest for the same data. Some part of the interfaces are already covered, others still need to be specify. The amount of needed work to comply to this specification should be quantified and a decision of whether or not OpenETCS join the OSLC project should be made.

During the tool chain development we need to maybe go step by step and start with exchange by file and then add more restful architecture. At each development step each tool should conform to the defined interface.

8 Integration in the tool platform

Chap resp. ??

8.1 Eclipse plug-ins

7.7 Tools chain	
7.7.1 Usage	
R-WP2/D2.6-X-36	The tools chain shall be composed as far as possible of Open Source components licensed under a license compatible with the EUPL license.
R-WP2/D2.6-X-36.1	Closed source components may be used, but only if their use is not mandatory in the process, or if an open source counterpart is provided.
R-WP2/D2.6-X-36.2	If a closed source component is used, it has to be displayed how an open source component has to be designed to replace the closed component later.
R-WP2/D2.6-X-37	The tools chain shall be portable to common operating systems.
R-WP2/D2.6-X-37.1	The tool chain shall run stable on all main operating systems.
R-WP2/D2.6-X-37.2	The tool chain shall run with a good performance on all main operating systems.
R-WP2/D2.6-X-38	The tools used in the tool chain shall be able to cooperate, i.e. the outputs of one tool will be suitable to be used as the inputs of another tool.
R-WP2/D2.6-X-38.1	All possible input and output formats of a tool have to be documented.
R-WP2/D2.6-X-38.2	Open data formats shall be used for the tool cooperation.
R-WP2/D2.6-01-032	If tools are required for configuration management, they will be considered as part of the tool chain.
R-WP2/D2.6-X-40	The tools chain shall allow to generate executable code from the model(s).
7.7.2 Information management	
R-WP2/D2.6-X-41	The tools chain shall be sufficiently robust to allow large software management (at least covering the onboard part of the SUBSET-026).
R-WP2/D2.6-X-41.1	It shall allow modularity at any level (proof, models, software).
R-WP2/D2.6-X-41.2	It shall allow the management of documentation.
R-WP2/D2.6-X-41.3	It shall allow distributed software development.
R-WP2/D2.6-X-41.4	It shall allow simultaneous multi user usage.
R-WP2/D2.6-X-41.5	It shall include an issue-tracking system, in order to allow change management and errors/bugs management.
R-WP2/D2.6-X-41.6	It shall allow to document/track the differences between the models and the ERTMS reference.
R-WP2/D2.6-X-41.7	It shall support management of subsequent Subset-026 versions, as well as differences tracking between Subset-026 versions.
R-WP2/D2.6-X-41.8	It shall allow concurrent version development, or be compatible with tools allowing concurrent version development.
R-WP2/D2.6-X-41.9	The version management tools shall use model-based version control instead of text-based version control, when appropriate.
R-WP2/D2.6-X-41.10	In particular it shall allow to track the roles and responsibilities of each participant on a configuration item, at each step of the project lifecycle.
R-WP2/D2.6-X-41.11	In particular, version management shall allow to track version of the safety properties together with the models.
R-WP2/D2.6-01-035	The tool chain shall allow traceability between:
R-WP2/D2.6-01-035.01	the documentation/requirements and the models,

R-WP2/D2.6-01-035.02	the documentation/requirements and the tests,
R-WP2/D2.6-01-035.03	the models and the tests,
R-WP2/D2.6-01-035.04	the documentation/requirements and the models,
R-WP2/D2.6-01-035.05	the documentation/requirements and the safety properties/requirements,
R-WP2/D2.6-01-035.06	the models and the safety properties/requirements,
R-WP2/D2.6-01-035.07	the tests and the safety properties/requirements.
R-WP2/D2.6-X-43	The tools chain shall be compliant to EN 50128 for the corresponding tool
7.7.3 Testing	
R-WP2/D2.6-01-036	The SFM shall be executable in debug mode (step-by-step), allowing inspection of states, variables and I/O.
R-WP2/D2.6-01-037	The environment shall be emulated by high level construction of the inputs. Justification. “High level” means that it will not be necessary to define bitwise the inputs at each cycle. On the contrary, some automation will be available to define the behavior of the inputs.
R-WP2/D2.6-01-038	The tool chain shall allow to write, execute and store test cases and use cases for the SFM.
R-WP2/D2.6-X-47	Version management will allow to map test cases version to the SFM, the FFM and source code versions.
R-WP2/D2.6-X-48	The tool chain shall allow to generate test cases for the SFM, the FFM and source code from a test model.
R-WP2/D2.6-X-49	The tool chain shall allow to write, execute and store test sequences combining multiple test cases for the SFM, the FFM and source code.
7.7.4 Conformance to standards	
R-WP2/D2.6-X-50	Each tool in the tool chain shall be classified among T1, T2 and T3 depending on its usage in the process.
R-WP2/D2.6-01-042	The tool chain shall conform to EN 50128 requirements, for the corresponding SIL and tool class 6 .
R-WP2/D2.6-01-042.01	For T2 and T3 tools 7 , the choice of tools shall be justified, and the justification shall include how the tools failures are covered, avoided or taken into account (ref. to EN 50128 6.7.4.2).
R-WP2/D2.6-01-042.02	All T2 and T3 tools must be provided with their user manuals.
R-WP2/D2.6-01-042.03	For all T3 tool, the proof of correctness or the measure taken to guarantee the correctness of the output w.r.t. their specification and the inputs shall be provided
R-WP2/D2.6-01-042.03.01	. . . for data transformation,
R-WP2/D2.6-01-042.03.02	. . . for software transformation (e.g. translation, compilation. . .).

Appendix B: WP7-specific requirements

R-WP7/T7.3-X-1 The tool chain shall include support for the following graphical user interface facilities:

R-WP7/T7.3-X-1.1 model syntax highlighting and auto-completion, for the parts of the model that are text-based

R-WP7/T7.3-X-1.2 graphical representation of Subset-026 braking curves

R-WP7/T7.3-X-1.3 tree-based display of model elements

R-WP7/T7.3-X-1.4 model animation and debugging

R-WP7/T7.3-X-1.5 perspective management: offer in a single graphical user interface several interconnected views for SRS, model and tests.

R-WP7/T7.3-X-2 The tool chain shall provide the following reporting facilities:

R-WP7/T7.3-X-2.1 detailed implementation metrics

R-WP7/T7.3-X-2.2 detailed traceability reports

R-WP7/T7.3-X-2.3 detailed model errors and warnings list

to be continued

