OETCS/WP7/O7.3.1

openETCS

ITEA2 Project
Call 6 11025
2012 – 2015

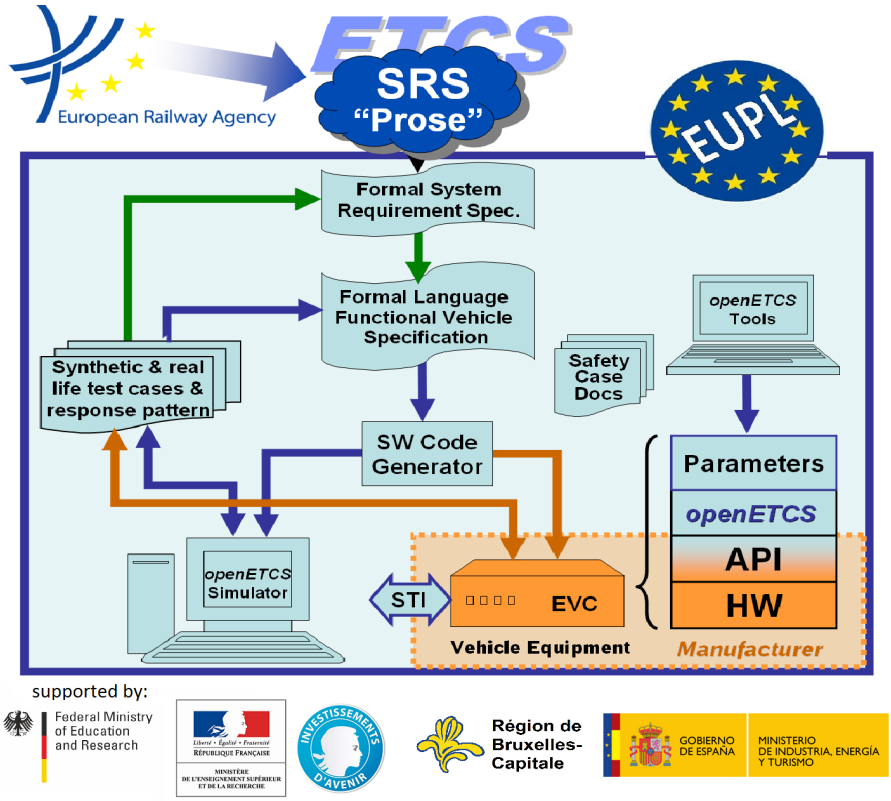
Work-Package 3: “Tool chain”

Tool chain Development Plan

Description of the tool chain development process

Cecile Braunstein and Jan Peleska

May 2012



This page is intentionally left blank

Work-Package 3: “Tool chain”

OETCS/WP7/O7.3.1
May 2012

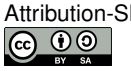
Tool chain Development Plan

Description of the tool chain development process

Cecile Braunstein and Jan Peleska

University Bremen

Software development Plan



Prepared for ITEA2 openETCS consortium
Europa

Abstract: This document defines the development process of the openETCS tool chain.

Table of Contents

Figures and Tables

Figures

Tables

Document Information

Document information	
Work Package	WP7
Deliverable ID or doc. ref.	O7.3.1
Document title	Tool chain development Plan
Document version	00.02
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
00.01	29.04.2013	C. Braunstein	Document creation
00.01	26.06.2013	C. Braunstein	Document correction

List of Terms

Notation	Description
COTS	Commercial off-the-shelf.
EMF	Eclipse Modeling Framework.
ETCS	European Train Control System.
IDE	integrated development environment.
OBU	On Board Unit.
SIL	System Integrity Level.
SRS	System Requirement Specification.
SSRS	Subsystem Requirement Specification.

1 Introduction and Motivation

The purpose of this document is to set out the development plan for the tool chain design, development and integration process. Following the goals of the openETCS project defined in [4], this document contributes to the **definition of a tool chain for developing on board software that can be certified by EN50128 requirements (see [7])**.

The tool chain provides the tools support and the development process as defined in [1] for the following openETCS activities :

- Formalization of the SSRS
- Design of the semi-formal model of ETCS on board.
- Design of the formal model of part of the ETCS on board.
- Code generation for non vital part of the OBU
- Documentation production
- Requirements traceability
- Testing on various levels of the development process
- Formal verification and Validation of the produced software

The target software of the tool chain is a the OBU based on the SRS Subset-026 [6]. The tool chain should help to assist the design process of *certifiable* SIL4 software (see. [3]) for the OBU.

This document provides the following information:

- Methods for specifying the OpenETCS tool chain - section 3
- Life cycle of the OpenETCS tool chain - section 4
- Development Environment of the OpenETCS tool chain - section 5

2 Reference documents

2.1 Project 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] Merlin Pokam and Norbert Schäfer. Report on CENELEC standards. Requirements D2.2, openETCS, April 2013.

[4] Project Outline Full Project Proposal Annex openETCS *open proofs methodology for the european train control system.* Requirements v2.2, openETCS, 2011.

[5] 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.

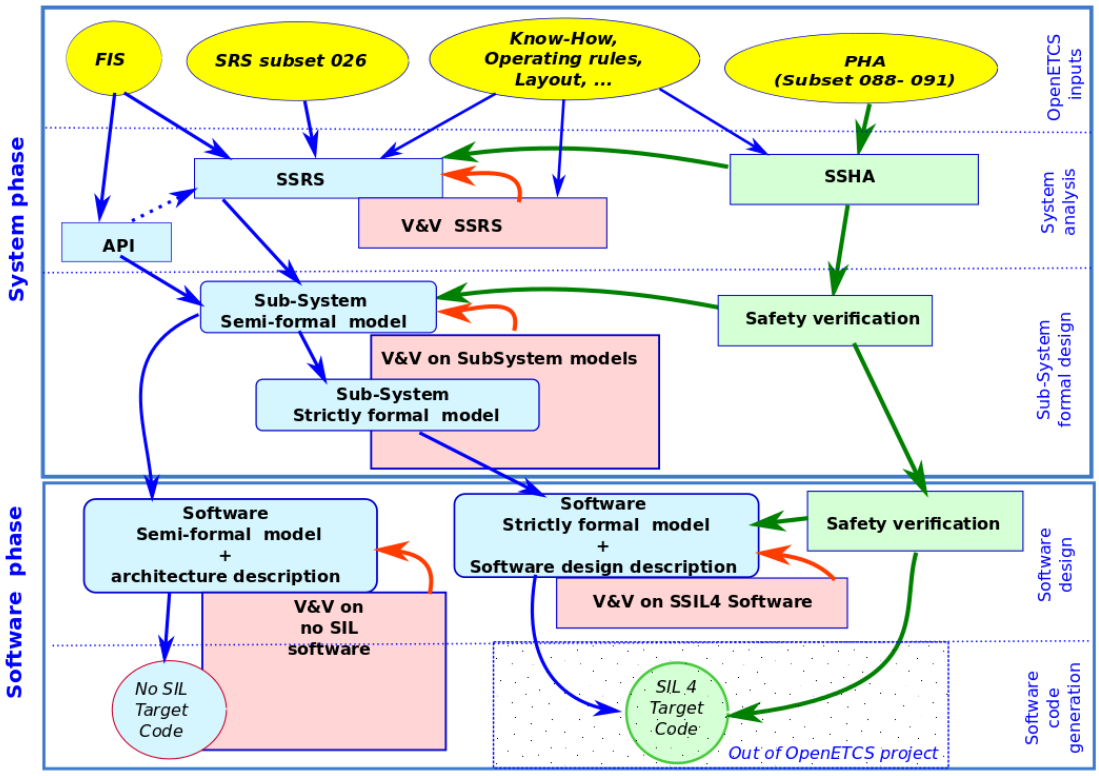


Figure 1. Life cycle of the on board software

2.2 Standards documents

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

[7] European Standard. *Railway applications-Communication, signalling and processing system- Software for railway control and protecton system*. CENELEC EN 50128. DIN, October 2011.

[8] Object Management Group. OMG Systems Modeling Language (OMG SysML^{TM}). `http://www.omgsysml.org`, June 2012.

3 OpenETCS Tool chain Definition Methods

The openETCS tool chain is the implementation of the design process of the on board unit (OBU) according to the CENELEC EN 50128. It coordinates the set of tools and methods needed to accomplish this task and it defines the procedure to follow to achieve a on board unit software *certifiable* SIL4.

Following the proposal of the [2] the tool chain implements the life cycle presented figure 1.

Moreover the tool chain should support the activities for producing certifiable software such as :

- Software planning
- Requirements tracing
- Tool confidences
- Documentation/report production

- Testing
- Verification and validation

The tool chain should also take care to provide the following functioning infrastructure to allow robust distributed development within the defined life cycle.

- a continuous automated build system,
- mechanisms to upgrade tools in the platform,
- mechanisms to add tools to the chain at a later stage (without breaking compatibility),
- tool chain documentation system.

Finally the OpenETCS tool chain should satisfy the requirements of [1]. Annex A provides a check list of all requirements the tool chain must fulfill.

4 OpenETCS Tool chain Life Cycle

This part defines the development of the tool chain itself. It defines the steps to achieve for the implementation of the tool chain.

4.1 OpenETCS Tool chain use of COTS

To design the life cycle defined in the previous section, the tool chain will use the components selected by the WP7. The tool chain will intensively use COTS application to reduce the cost for the development of the OBU.

The decision on the selection of means of description, tools and tool platform are defined in the document [5]. The selection of the secondary tools, e.g. the choice of verification and validation tools are described in [].

4.2 OpenETCS Tool chain development

The tool chain development will follow the SCRUM process. The following activities will be covered by the tool chain development.

- openETCS tool chain architecture specification
  Definition of the tool chain composition
- openETCS tool chain design specification
  Definition of how the tool chain is implemented including the definition of the interoperability mechanism
- Software development
  Implementation of the tool chain in particular for the need of tool interoperability
- Test and verification plan
  Definition of how to test the tool chain.

The tool chain specification is out of the scope of the document. It has been defined by the WP2, the requirements are listed in Appendix 1.

5 OpenETCS Tool chain Development Environment

5.1 Development environment

Any GNU system running on a PC may be used. IDE such as Eclipse may also be used. The tool chain should be under the GIT version control system [10].

To allow robust distributed development, the environment provides :

- a continuous automated build system,
- an issue tracker,
- a request/extension tracker,
- a backlog
- a documentation system.

The environment will also provides the infrastructure for a SCRUM development.

5.2 Design Method

The tool chain will be designed with SysML ([8]) Block diagram with associated interface specifications following the SysML syntax. The interface will provide the set of artifacts produced or consumed by each tools.

More information may be added to the tool chain model in order to facilitate the design, the certification and the analysis of the openETCS tool chain following the approach [9] or [12].

5.3 Tool chain Development

Following the scrum process, there will be regular releases for the tool chain.

The tool chain is the integration of tools, thus the tool chain development takes care of all the aspects of the tool integration problem (see [11]): (1) Platform integration, (2) Data integration, (3) Presentation integration, (4) Control integration, and (5) Process integration. The point (1) is resolved by the choice of the tool platform. The point (5) is addressed during the design phase of the tool chain.

It has been design that the tool platform hosting the tools will be eclipse with eclipse modeling framework (EMF) plug-in. To ensure a good integration, the tool chain development should follow these steps:

1. Register tool in Eclipse (3)
2. Defining the communication with other tools: method of data exchange (2), Control Integration (4)

5.4 Tool chain verification

The validation and verification plan of the tool chain is described in the a separate document [].

6 References

[9] Oscar Slotosch. Model-based tool qualification : The roadmap of eclipse towards tool qualification. *Springer*, 2012.

[10] Scott Chacon. *Pro Git*. Apress, Berkely, CA, USA, 1st edition, 2009.

[11] AnthonyI. 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.

[12] Fredrik Asplund, Matthias Biehl, and Frédéric Loiret. Towards the automated qualification of tool chain design. In *Computer Safety, Reliability, and Security*, volume 7613 of *Lecture Notes in Computer Science*, page 392–399. Springer, 2012.

Check	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. . .).