OETCS/WP7/D7.1 – 00/01

openETCS

ITEA2 Project
Call 6 11025
2012 – 2015

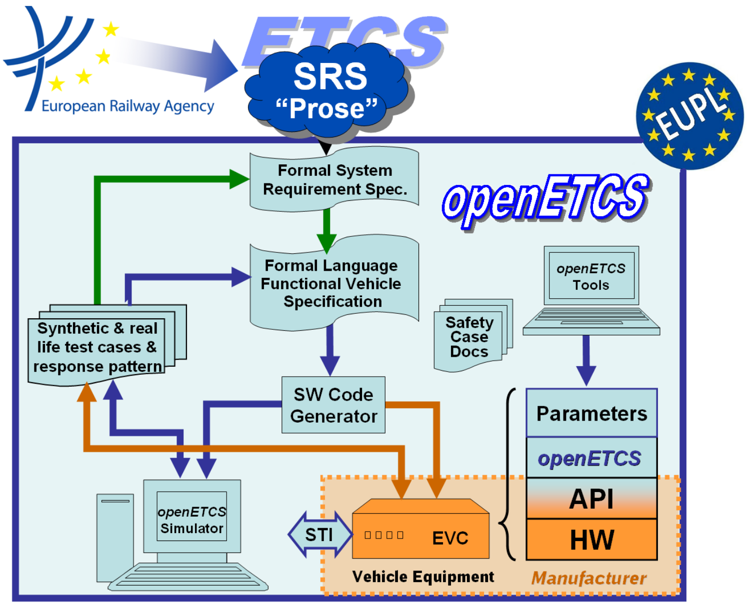
Work-Package 7: “Primary tool chain”

Report on the final choices for the primary toolchain

Decision on the final choice for the means of description (O7.1.4), tools (O7.1.8) and tool platform (O7.1.11)

Marielle Petit-Doche

July 2013



Funded by:

Federal Ministry of Education and Research

Région de Bruxelles-Capitale

Gobierno de España

Ministerio de Industria, Energía y Turismo

This page is intentionally left blank

Work-Package 7: "Primary tool chain"

OETCS/WP7/D7.1 – 00/01
July 2013

Report on the final choices for the primary toolchain

Decision on the final choice for the means of description (O7.1.4), tools (O7.1.8) and tool platform (O7.1.11)

Marielle Petit-Doche
Systerel

Deliverable

Prepared for openETCS@ITEA2 Project

Abstract: This document gives a description and the results of the first task of WP7. The objectives of the task are to analyse and recommend, means, tools and platform to develop the primary tool chain of Open ETCS.

Table of Contents

Figures and Tables

Figures

Tables

Document information	
Work Package	WP7
Deliverable ID or doc. ref.	D.7.1 (including O7.1.4, O7.1.8, O7.1.11)
Document title	Report on the final choices for the primary toolchain
Document version	00.01
Document authors (org.)	Marielle Petit-Doche (Systerel)

Review information	
Last version reviewed	00.01
Main reviewers	

Approbation			
	Name	Role	Date
Written by	Marielle Petit-Doche	WP7-T7.1 Sub-Task Leader	
Approved by	Michael Jastram	WP7 leader	

Document evolution			
Version	Date	Author(s)	Justification
00.01	08/07/2013	M. Petit-Doche	Document creation

1 Introduction

The aim of this document is to report the results of the task T7.1 of WP7 : "Primary tool Chain analyses and recommendations".

1.1 T7.1 objective

The objectives of this task are to identify the modelling languages, the tools and the tool platform suitable to define the primary tool chain of OpenETCS project. This primary tool chain shall cover all specification and design activities of the OpenETCS process (part in blue in 1). For more details see D2.3 [1] and D2.6-9 [2]. Means and tools for other activities described on figure 1 (mainly verification, validation and safety activities) are going to be discussed during the tast T7.2 of WP7.

1.2 T7.1 activities

The activities have started in November 2012, with a proposal of benchmark organisation. After selection of a set of case studies (specified in D2.5 [3]), different approaches have been proposed and models have been stored on a common open github repository. All the methods have been presented during a public meeting in April 2013.

Besides, a set of criteria have been defined according the D2.6-9 requirement document [2]. The results are record in the outputs O7.1.3-O7.1.7 [4] for means and tools and O7.1.9 [5] for tool platform.

A decision meeting took place the 4th of July 2013 to analyse the results of the benchmark and to decide which means and tools will be retained during the process.

Results of the decision are given in this current document.

1.3 Glossary

API Application Programming Interface

FIS Functional Interface Specification

HW Hardware

I/O Input/Output

OBU On-Board Unit

PHA Preliminary Hazard Analysis

SIL Safety Integrity Level

SRS System Requirement Specification

SSHA Sub-System Hazard Analysis

SSRS Sub-System Requirement Specification

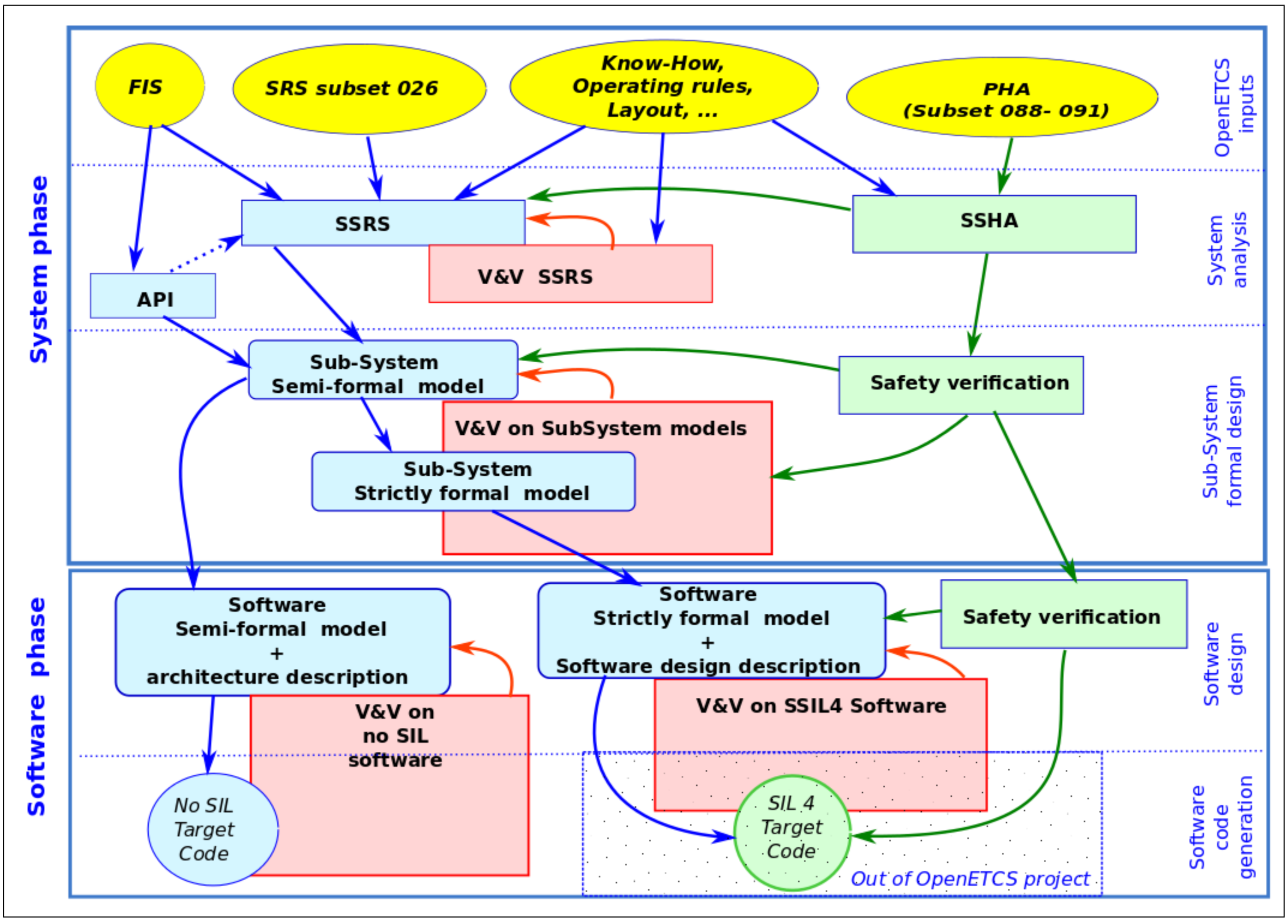


Figure 1. Main OpenETCS process

SW Software

V&V Verification & Validation

2 Results on means and tools for primary tool chain

2.1 Initial list of candidates

The initial list of candidates is the following:

- GOPRR
- CORE
- ERTMSFormalSpecs
- SysML with Papyrus
- SysML with Entreprise Architect
- SCADE
- EventB with Rodin
- Classical B with Atelier B
- Petri Nets
- System C
- UPPAAL
- Why3
- GNATprove

For each approach and tool, the initial author of the evaluation is the partner in charge of the modelling. Two assessors, for each approaches, are in charge of the review of the evaluation and can correct it or add comments. For each approaches, the models are available on the public github `https://github.com/openETCS/model-evaluation/tree/master/model` (see 2). Scores of each approaches according the evaluation criteria are record in appendix of the outputs O7.1.3-O7.1.7 [4].

2.2 Evaluation results

In the conclusion part of O7.1.3-O7.1.7 [4], the first table show how the evaluated approaches cover the openETCS design process:

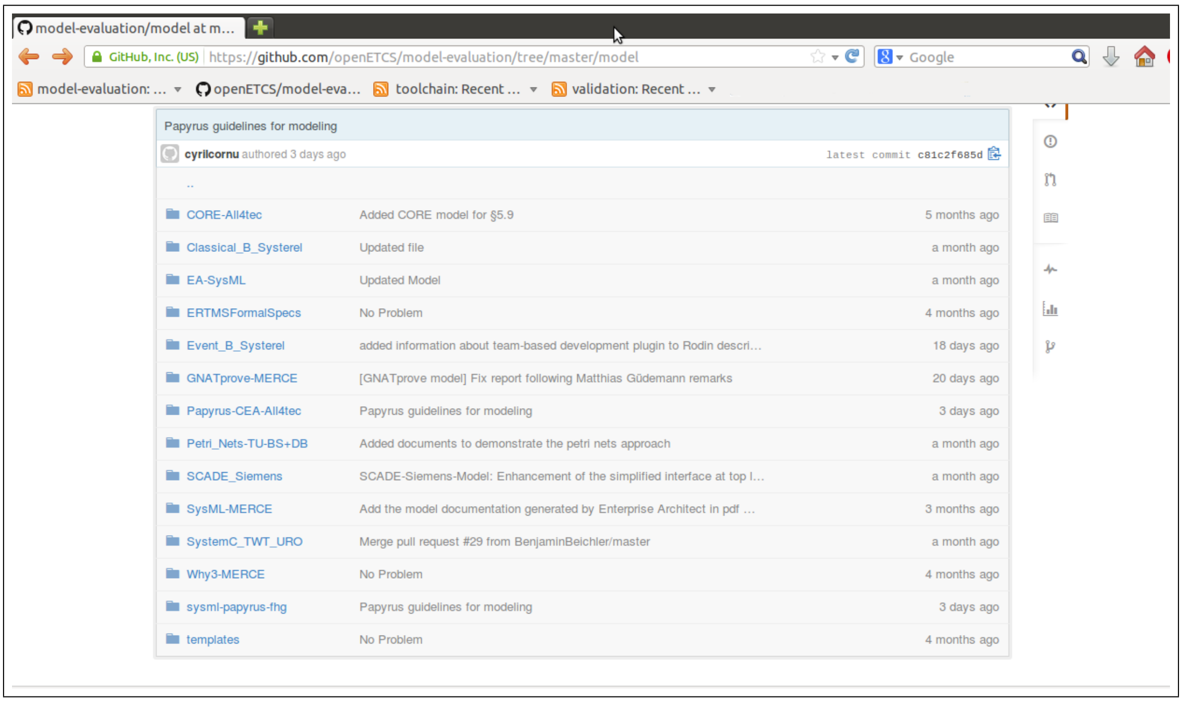


Figure 2. Repository of models

	GOPRR	ERTMSFormalSpecs	SysML with Papyrus	SysML with EA	SCADE	EventB	Classical B	System C	Petri Nets	GNATprove
System Analysis	5	1	7	**9**	3	**9**	3	2	**6(9)**	2 (3)
Sub-system formal design	**9**	**9**	6	7	**9**	**9**	5	5	**6(9)**	3 (4)
Software design	**9**	0	6	7	**9**	6	**9**	**9**	**6(9)**	**6(9)**
Software code generation	**9**	0	3	3	**9**	3	**9**	6	2 (3)	**6(9)**

The highest score is 9 and means that the criteria is fully respected, the lowest score is 0. The higher scores for each approach is graphically represented on figure 3.

The second table in the conclusion part of O7.1.3-O7.1.7 [4], shows that all evaluated approaches, except GnatProve, are adapted to modeling and design activities:

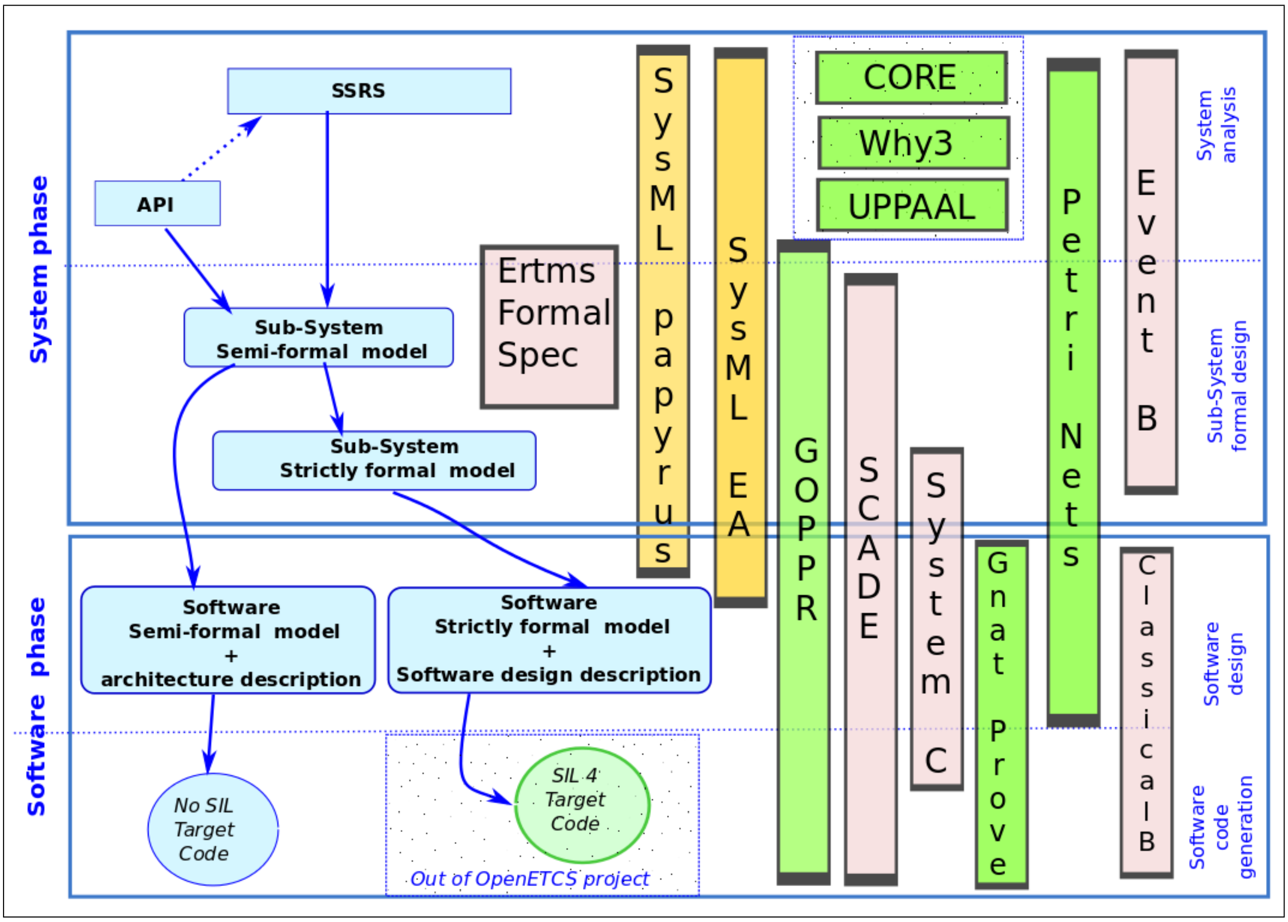


Figure 3. Results of candidates

	GOPRR	ERTMSFormalSpecs	SysML with Papyrus	SysML with EA	SCADE	EventB	Classical B	System C	Petri Nets	GNATprove
Documentation	3	7	6	7	8	7	0	0	2 (3)	2 (3)
Modeling	**9**	**9**	**9**	**9**	**9**	**9**	**9**	8	**6(9)**	2 (3)
Design	6	**9**	6	7	**9**	7	8	**9**	5(7)	3 (4)
Code generation	**9**	1	3	4	**9**	3	**9**	5 *	2 (3)	**6(9)**
Verification	0	7	6	3	8	**9**	**9**	4 *	**6(9)**	**6(9)**
Validation	0	**9**	5	4	8	**9**	4	7	**6(9)**	**6(9)**
Safety analyses	0	0	4 *	6	1	6	3	3 *	5(7)	2(3)

According this result, GantProve has been proposed to joined the evaluation of secondary tools (task T7.2). During the benchmark activities, UPPAAL, which is a tool dedicated to the verification and validation of time-constraints properties, has also been proposed for the benchmark on secondary tools.

2.3 Short list

Todo: MPD: Description of discarded approach and short list of approaches

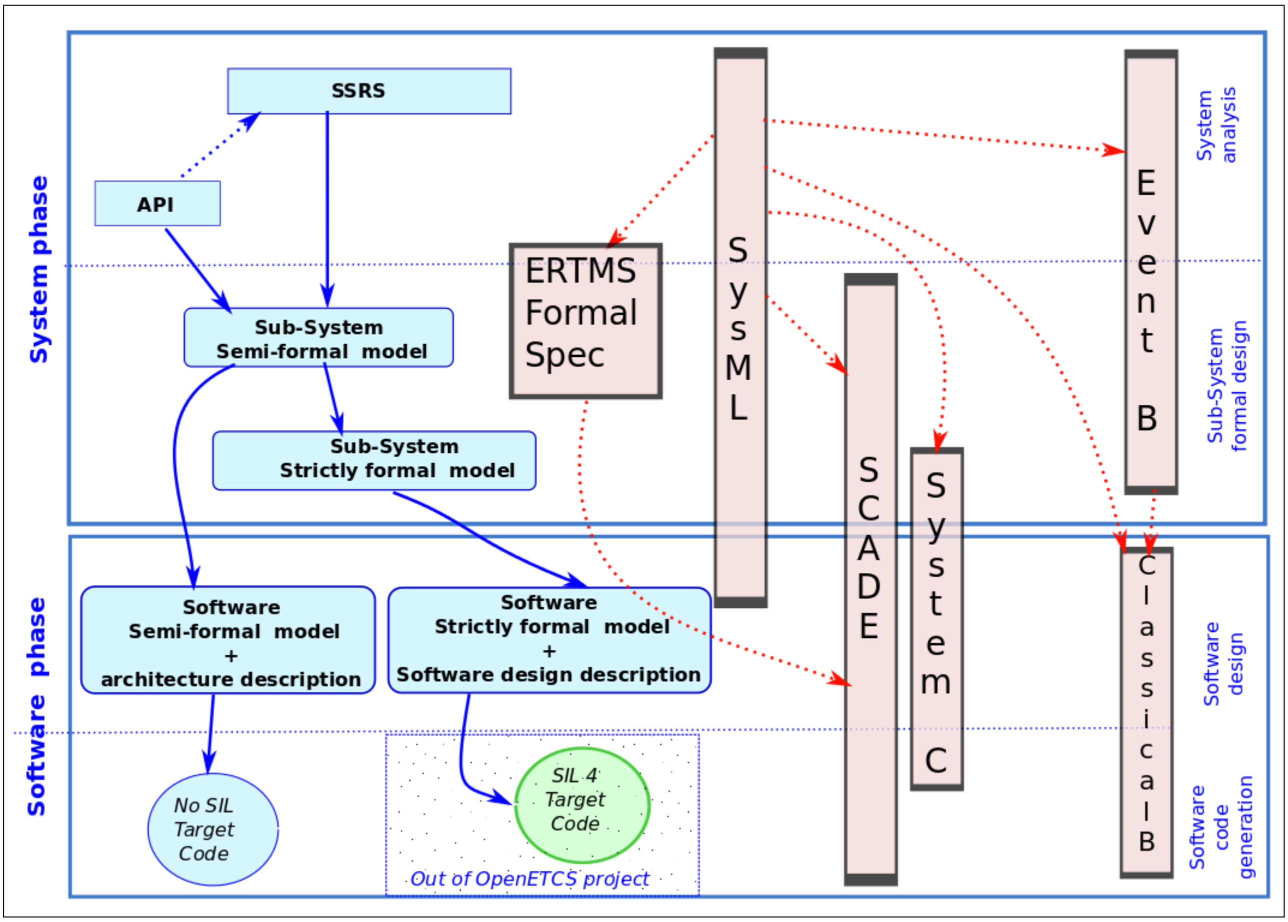


Figure 4. Short list of candidates

3 Results on tool platform

The tool platform should provide mechanisms to integrate various tools. The tool platform is not the primary nor secondary tools, nor the tool chain. It is the support for the tool chain implementation, it shall help to integrate the tools into a seamless tool chain. The evaluation will focus on the integration capabilities of the tool platform.

Todo: Description of the candidates by Cecile Braunstein

3.1 Initial list of candidates

- Eclipse
- TopCased/Polarsys
- RTP-Cesar
- Mono/.NET
- SCADE

After a first round, Mono/.NET and SCADE were discard because they do not comply to our tool platform definition. RTP-CESAR was also discard, the maturity of this project is not yet usable. Finally, Eclipse has been chosen as a tool platform, the possibility to use Polarsys and take some part of the TopCased tool chain are discussed in the next sections.

3.2 Eclipse

3.3 Version Management

3.4 Topcased and Polarsys

Topcased is a tool for systems engineering, based on Eclipse and various Eclipse projects. Polarsys is a project concerned with the long term support of the Topcased tool chain. There is an overlap between Topcased and the openETCS tool chain. There is also an overlap between the objectives of openETCS and Polarsys:

Topcased and openETCS tool chain. Both, Topcased and the openETCS tool chain are based on Eclipse. Further, the openETCS tool chain will definitely use Papyrus, which is also part of Topcased. And last, both are concerned with covering all aspects of the V-Model, although for different domains (aviation vs. rail).

Polarsys and openETCS. The objectives of Polarsys and openETCS overlap significantly as well: Both are concerned with tools in a safety-critical domain, requiring tool qualification, etc. They are also concerned with long term support through open source.

3.4.1 State of Topcased and Polarsys

While the state of the art document mentions Topcased [], it was not evaluated as a whole. Merely the Papyurus component of Topcased was evaluated, but a newer version than the one used by Topcased.

Topcased is using a fork of an old version of Papyrus (Ver. 0.8.2) which is no more supported by the CEA (actual version 0.10.X) and, as the CEA is not part of Topcased, no more code development over this version/Topcased will be done by CEA. Unfortunately, the development on Topcased modeler (forked version of Papyrus) is not so active anymore: 60 commits on the 3 last months (as of July 2013) against more than 1600 commit for Papyrus. Further, the actual version of Papyrus have been greatly improved with respect to stability since version 0.8.2, and some stability issues may have not been corrected in Topcased.

To conclude, Topcased requires Eclipse 3.7.2 Indigo (1.5 year-old version) which is no more supported by the Eclipse foundation. Some part of Topcased initiative (plugins/add-ons) may still be very useful to the openETCS project, so we will reach out to the Polarsys community to see whether there is an interest in aligning versions for long term support. The versions currently used in Topcased are not suitable for the openETCS tool chain, unfortunately.

4 Decision

4.1 Decision on the tool platform

Todo: MPD: Choice of Eclipse (version Kepler 4.3 RC3 ?)

4.2 Decisions for high level step

Todo: MPD : Choice of SysML with Papyrus (release 1.2 for SysML and 0.10.0 for papyrus ?)

4.3 Propositions of approach to cover all the design process

Todo: MPD: list of on going propositions to cover low levels of design process

4.4 Conclusion

References

[1] Marielle Petit-Doche and Matthias Güdemann. openETCS process. Technical Report D2.3, OpenETCS, 2013.

[2] Sylvain Baro and Jan Welte. Requirements for openETCS. Technical Report D2.6, OpenETCS, 2013.

[3] Guillaume Pottier David Mentre, Stanislas Pinte and WP2 participants. Methods and tools benchmarking methodology. Technical Report D2.5, OpenETCS, 2013.

[4] Marielle Petit-Doche and WP7 partners. Evaluation of the means and tools against the WP2 requirements. Results O7.1.3-O7.1.7, OpenETCS, 2013.

[5] Cécile Braunstein and WP7 partners. Evaluation of tool platforms against the WP2 requirements. Results O7.1.9, OpenETCS, 2013.

[6] Michael Jastram, Marielle Petit-Doche, Jonas Helming, and Jan Peleska. openETCS toolchain WP7 descritpiton of work. Defin D01, OpenETCS, February 2013.

[7] Rico Kaseroni. Project quality assurance plan. Technical Report D1.3.1, OpenETCS, 2013.

[8] Merlin Pokam and Norbert Schäfer. Report on CENELEC standards. Technical Report D2.2, OpenETCS, 2013.

[9] Jan Welte and Hansjörg Manz. Report on existing methodologies. Technical Report D2.1, OpenETCS, 2013.

[10] Hardi Hungar. Report on v&v plan and methodology. Technical Report D4.1, openETCS, 2013.

[11] Jani Welte. Safety plan. Technical Report O 4.4.1, openETCS, October 2013.

[12] Klaus-Rüdiger Hase. Project outline full project proposal annex openETCS. Technical Report v2.2, openETCS, 2011.

[13] UNISIG. SUBSET-026 – System Requirements Specification. Technical Report 3.3.0, ERA, March 2012.

[14] UNISIG. SUBSET-076 – Test related ERTMS documentation (this version is related to version 2.3.y of SUBSET-026). Technical Report 2.3.y, ERA.

[15] UNISIG. SUBSET-088 2.3.0 - ETCS Application Levels 1 & 2 - Safety Analysis. Technical Report 2.3.0, ERA.

[16] UNISIG. SUBSET-091 3.2.0 — Safety Requirements for the Technical Interoperability of ETCS in Levels 1 & 2. Technical Report 3.2.0, ERA.

[17] UNISIG. SUBSET-034 3.0.0 — Train interface FIS. Technical Report 3.0.0, ERA.

[18] Comission Decision. CCS TSI for HS and CR transeuropean rail. Technical Report 2012/88/EU, EU, January 2012.

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

[20] European Standard. *Railways applications — The specification and demonstration of Reliability, Availability, Maintainability and Safety (RAMS) — Part 1: Basic requirements and generic process*. CENELEC EN 50126_1. DIN, January 2000.

[21] European Standard. *Railway applications — Communication, signalling and processing systems — Safety related electronic systems for signalling*. CENELEC EN 50129. DIN, May 2003.

Appendix A: SysML and Scade

Todo: Description of the approach by Uwe Steinke.

Diagram A1 illustrates the most important components and operational relationships of a system and software modelling toolchain based on SysML, Papyrus, SCADE and Eclipse. All components and links shown with solid lines are available, while the dashed ones are intended to be implemented within the openETCS project.

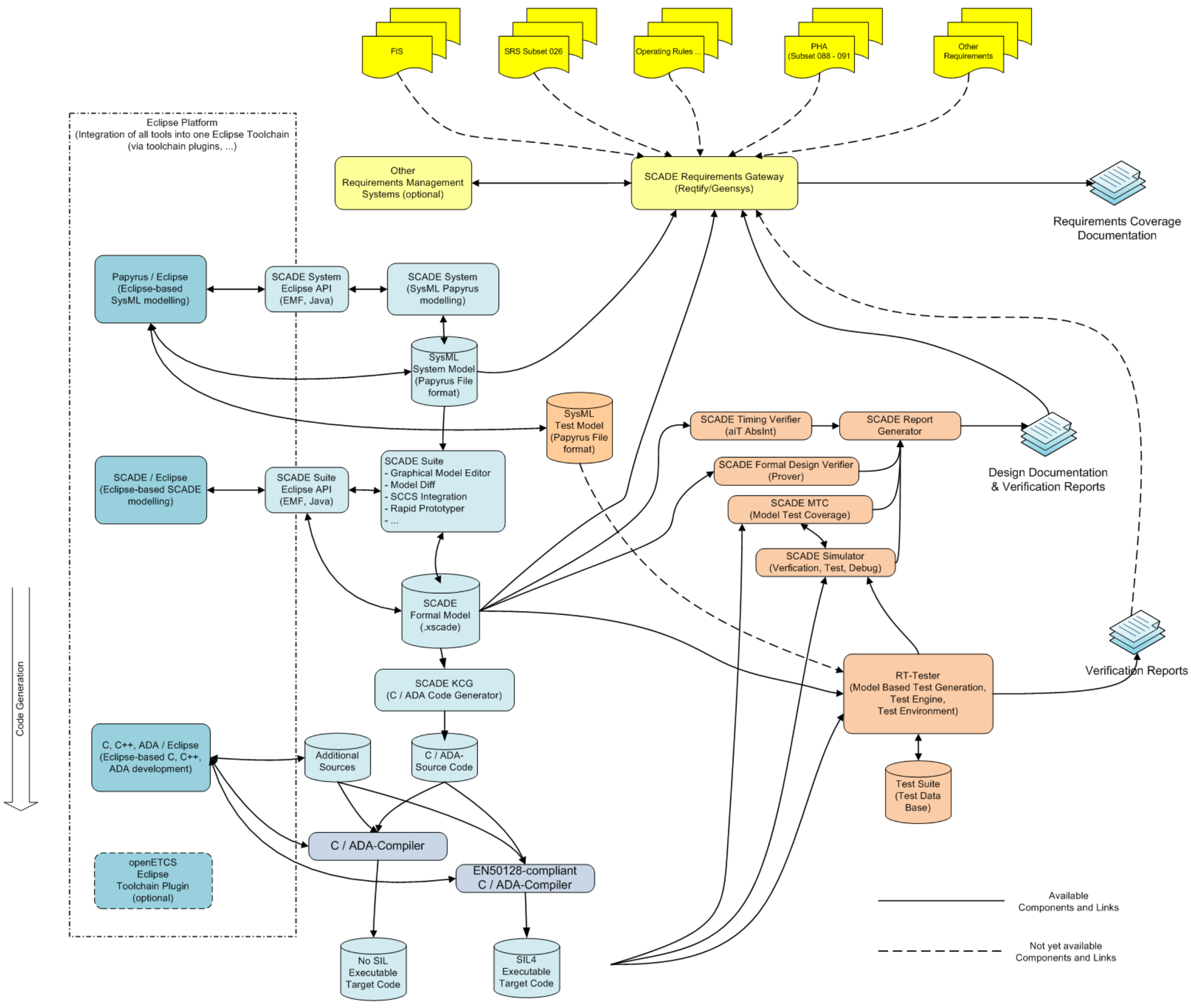


Figure A1. SysML SCADE Toolchain

The diagram colors are chosen related to the colors in 1:

- Requirements and requirements management components in yellow
- "Blue" openETCS design process (see 1) elements in light and dark blue
- Eclipse is painted dark blue

- Verification elements in red

Within the following paragraphs and subsections a short description of this approach will be given by walking through the tool chain and the design process.

A.0.1 Requirements management

The SCADE Requirements Management Gateway is based on Reqtify from Geensoft / Dassault Systems and serves to collect and link all requirements from the openETCS input documents and related objects as design und verification documents, model and source code artefacts, test cases, test protocols etc. It supports impact analyses and generates requirements traceability and requirements coverage reports. If needed for the openETCS process, it can be complemented with other requirement management systems and already comes with interfaces to these.

A.0.2 Semiformal System and Subsystem Modelling with SCADE System / Papyrus

SCADE System is an integration of Papyrus into the SCADE IDE intended for SysML system modelling. It allows to modelize the interactions and hierarchical dependencies between the various parts of a complex system through design elements representing functions, data and interfaces.

The idea is to model system structures, data types / data dictionaries, inputs, outputs, interfaces and relationships between blocks with SysML and transfer it to native SCADE for behavioral modelling automatically. Since Papyrus and SCADE System are using the same file formats, there is no prevention of using all SysML capabilities that Papyrus supports, but in this case without automatic transfer to native SCADE.

SCADE System supports SysML Block Definition Diagrams (BDD) and Internal Block Diagrams (IBD).

A.0.3 Formal Modelling with SCADE Suite

SCADE Suite integrates all modelling, verifation and supporting SCADE tools under the roof of the SCADE IDE. The components relevant for descending part of the development "‘V"’ process are

- SCADE Suite Editor (Graphical and textual modelling)
- SCADE Requirements Gateway Integration (Linking of model artefacts with requirements)
- SCADE Model Check (model syntax check)
- SCADE Model Diff (model comparison)
- SCADE Simulator (graphical debugging, simulation and testing)
- SCADE Rapid Prototyper (quick control and display elements for rapid prototyping, optional)
- SCADE Code Generator KCG (C / ADA code generation)
- SCADE Reporter ((Design) report generator)

The most important tools for modelling are editor and code generator. The others mentioned are mainly verification tools, but very useful and practically indespensible for agile development.

At least, to cover all elements of the "blue" design process in 1 a C / C++ / ADA compiler is required. For building not safety-relevant executables any C-Compiler (gnu c, ...) is suitable, for safety relevant executables the compiler must be compliant with EN50128.

A.0.4 Model Verification

Will be completed soon.

A.1 Description of the approach for OpenETCS design process

Todo: How the proposed approach covers all "blue" design process (see 1) ?

The approach as specified in the previous subsections (Chapt. A0) (insert ref xxx) covers all elements of the "blue" design process (see 1) by using the SCADE tool chain including requirements management, semiformal system and formal subsystem/software modelling, code and executable generation. An Eclipse integration is provided (see following Chapt. xxx).

A.2 Integration of the approach with SysML/Papyrus

Todo: How the proposed approach can be integrated with the SysML/ Papyrus selected for the high level of design process ?

Because SCADE System is an integration of Papyrus into the SCADE IDE and SCADE System uses Papyrus file formats, the integration with SysML / Papyrus is available. A thrilling question for the openETCS process might be, if and - if yes - which of the artefacts on system level should be modelled with SysML, that can not be transferred to native SCADE automatically.

A.3 Integration of the approach with Eclipse

Todo: How the proposed approach can be integrated with the Eclipse, selected as platform for OpenETCS tool chain ?

Most of the the SCADE tools can be run and controlled via command line and/or via automation interfaces.

SCADE System (SysML modelling) and SCADE Suite (SCADE modelling) already come with Eclipse API plugins based on EMF. These enable to access (read and modify) the model project information, meta and model data from within Eclipse. The plugins additionally display the model structure, but they don't show the model graphic in Eclipse.

If graphical modelling should be done within Eclipse, this has to be implemented by openETCS. It is in doubt, if the effort for this activity would be applicable; without any effort, the SCADE editor should be used instead.

Nevertheless, the provided Eclipse integration is worthwile to supply all openETCS users, that are not directly working on the SCADE models, with an integrated Eclipse tool chain. The idea of such an integration is to have one build tool chain, that starts and runs an openETCS executable build process with one button click beginning from all (heterogeneous) sources and performing all necessary model transformations, code and executable generation. This could be achieved with an "‘openETCS Eclipse Tool Chain Plugin"’, implemented as part of the openETCS project with the goals ease-of-use and convenience.

In summary, an Eclipse integration is available. An optional "‘openETCS Eclipse Tool Chain Plugin"’ could improve the convenience for openETCS tool chain users.

A.4 Benefits versus OpenETCS requirements

> Todo: *Discuss the benefits in regards of OpenETCS requirements and expected results.*

The most important benefit of the SysML/SCADE approach is its seamless integration, completeness, maturity and qualification for safety critical development: it covers almost all aspects of the openETCS process and lets expect to fill gaps with manageable effort.

Therefore, the modelling work for openETCS can begin immediately.

A.5 Shortcommings versus OpenETCS requirements

> Todo: *Discuss the shortcommings in regards of OpenETCS requirements and expected results.*

The main drawback of the approach: it is mainly not open Source.

A.6 On going work for openETCS project

Will be completed soon.

> Todo: *Which are the elements to clarify, to specify or to develop, in order the approach suit the openETCS process ?*
>
> *How can we evaluate and plan this work ?*
>
> *which skills is needed ?*

A.7 Conclusion and other comments

Appendix B: SysML, ErtmsFormalSpec and Topcased

Todo: Description of the approach by Stanislas Pinte.

B.1 Description of the approach for OpenETCS design process

Todo: How the proposed approach covers all "blue" design process (see 1) ?

B.2 Integration of the approach with SysML/Papyrus

Todo: How the proposed approach can be integrated with the SysML/ Papyrus selected for the high level of design process ?

B.3 Integration of the approach with Eclipse

Todo: How the proposed approach can be integrated with the Eclipse, selected as platform for OpenETCS tool chain ?

B.4 Benefits versus OpenETCS requirements

Todo: Discuss the benefits in regards of OpenETCS requirements and expected results.

B.5 Shortcommings versus OpenETCS requirements

Todo: Discuss the shortcommings in regards of OpenETCS requirements and expected results.

B.6 On going work for openETCS project

Todo: Which are the elements to clarify, to specify or to develop, in order the approach suit the openETCS process ?

How can we evaluate and plan this work ?

which skills is needed ?

B.7 Conclusion and other comments

Appendix C: SysML and ClassicalB

Todo: Description of the approach by Alexander Stante.

C.1 Description of the approach for OpenETCS design process

Todo: How the proposed approach covers all "blue" design process (see 1) ?

C.2 Integration of the approach with SysML/Papyrus

Todo: How the proposed approach can be integrated with the SysML/ Papyrus selected for the high level of design process ?

C.3 Integration of the approach with Eclipse

Todo: How the proposed approach can be integrated with the Eclipse, selected as platform for OpenETCS tool chain ?

C.4 Benefits versus OpenETCS requirements

Todo: Discuss the benefits in regards of OpenETCS requirements and expected results.

C.5 Shortcommings versus OpenETCS requirements

Todo: Discuss the shortcommings in regards of OpenETCS requirements and expected results.

C.6 On going work for openETCS project

Todo: Which are the elements to clarify, to specify or to develop, in order the approach suit the openETCS process ?

How can we evaluate and plan this work ?

which skills is needed ?

C.7 Conclusion and other comments