

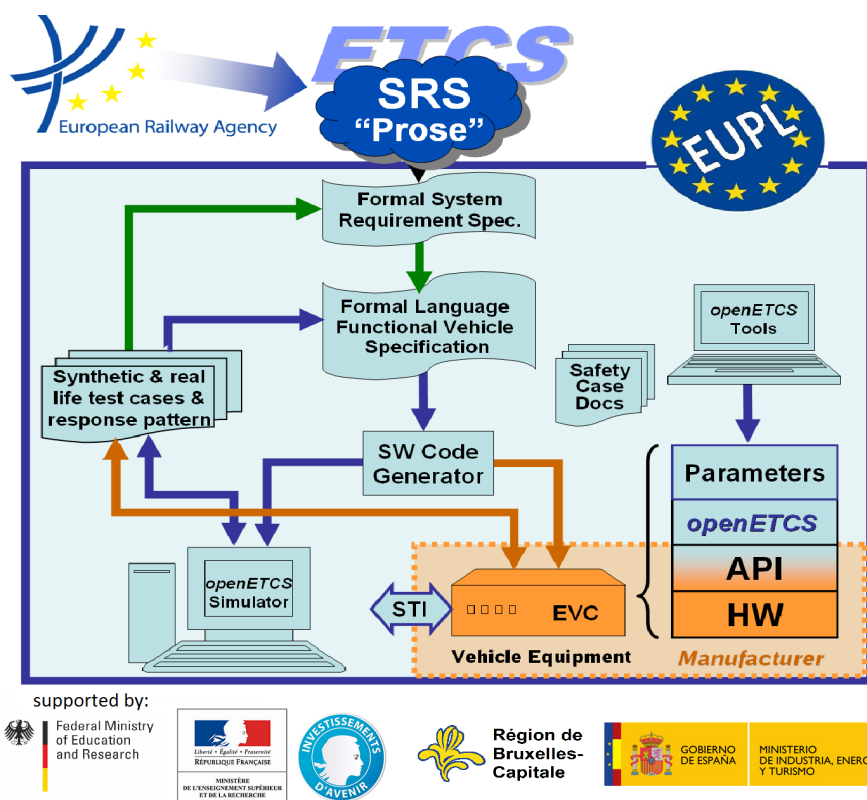
## Work-Package 7: “Primary tool chain”

## 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



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.

**Disclaimer:** This work is licensed under the "openETCS Open License Terms" (oOLT) dual Licensing: European Union Public Licence (EURL v.1.1+) AND Creative Commons Attribution-ShareAlike 3.0 – (cc by-sa 3.0)

THE WORK IS PROVIDED UNDER openETCS OPEN LICENSE TERMS (oOLT) WHICH IS A DUAL LICENSE AGREEMENT INCLUDING THE TERMS OF THE EUROPEAN UNION PUBLIC LICENSE (VERSION 1.1 OR ANY LATER VERSION) AND THE TERMS OF THE CREATIVE COMMONS PUBLIC LICENSE ("CCPL"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS OLT LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. TO THE EXTENT THIS LICENSE MAY BE CONSIDERED TO BE A CONTRACT, THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

<http://creativecommons.org/licenses/by-sa/3.0/>  
<http://joinup.ec.europa.eu/software/page/eupl/licence-eupl>

# Table of Contents

<b>Figures and Tables.....</b>	<b>v</b>
<b>1 Introduction.....</b>	<b>1</b>
1.1 T7.1 objective.....	1
1.2 T7.1 activities .....	1
1.3 Glossary .....	1
<b>2 Results on means and tools for primary tool chain .....</b>	<b>4</b>
2.1 Initial list of candidates .....	4
2.2 Evaluation results .....	4
2.3 Short list .....	7
<b>3 Results on tool platform.....</b>	<b>9</b>
3.1 Initial list of candidates .....	9
3.2 Eclipse.....	9
3.3 Version Management .....	9
3.4 Topcased and Polarsys.....	9
3.4.1 State of Topcased and Polarsys .....	10
<b>4 Decision .....</b>	<b>11</b>
4.1 Decision on the tool platform .....	11
4.2 Decisions for high level step .....	11
4.3 Propositions of approach to cover all the design process .....	11
4.4 Conclusion .....	11
<b>Appendix A: SysML and Scade .....</b>	<b>12</b>
A.0.1 Requirements management .....	13
A.0.2 Semiformal System and Subsystem Modelling with SCADE System / Papyrus .....	13
A.0.3 Formal Modelling with SCADE Suite .....	13
A.0.4 Model Verification .....	14
A.1 Description of the approach for OpenETCS design process.....	14
A.2 Integration of the approach with SysML/Papyrus .....	14
A.3 Integration of the approach with Eclipse .....	14
A.4 Benefits versus OpenETCS requirements.....	15
A.5 Shortcomings versus OpenETCS requirements.....	15
A.6 On going work for openETCS project.....	15
A.7 Conclusion and other comments.....	15
<b>Appendix B: SysML, ErtmsFormalSpec and Topcased .....</b>	<b>16</b>
B.1 Description of the approach for OpenETCS design process.....	16
B.2 Integration of the approach with SysML/Papyrus .....	16
B.3 Integration of the approach with Eclipse .....	16
B.4 Benefits versus OpenETCS requirements.....	16
B.5 Shortcomings versus OpenETCS requirements.....	16
B.6 On going work for openETCS project.....	16
B.7 Conclusion and other comments.....	16
<b>Appendix C: SysML and ClassicalB.....</b>	<b>17</b>

C.1	Description of the approach for OpenETCS design process .....	17
C.2	Integration of the approach with SysML/Papyrus .....	17
C.3	Integration of the approach with Eclipse .....	17
C.4	Benefits versus OpenETCS requirements .....	17
C.5	Shortcomings versus OpenETCS requirements .....	17
C.6	On going work for openETCS project .....	17
C.7	Conclusion and other comments .....	17

# Figures and Tables

## Figures

Figure 1. Main OpenETCS process ..... 2

Figure 2. Repository of models ..... 5

Figure 3. Results of candidates ..... 6

Figure 4. Short list of candidates ..... 8

Figure A1. SysML SCADE Toolchain ..... 12

## 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 [?] and D2.6-9 [?]. Means and tools for other activities described on figure 1 (mainly verification, validation and safety activities) are going to be discussed during the task 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 [?]), 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 [?]. The results are record in the outputs O7.1.3-O7.1.7 [?] for means and tools and O7.1.9 [?] 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

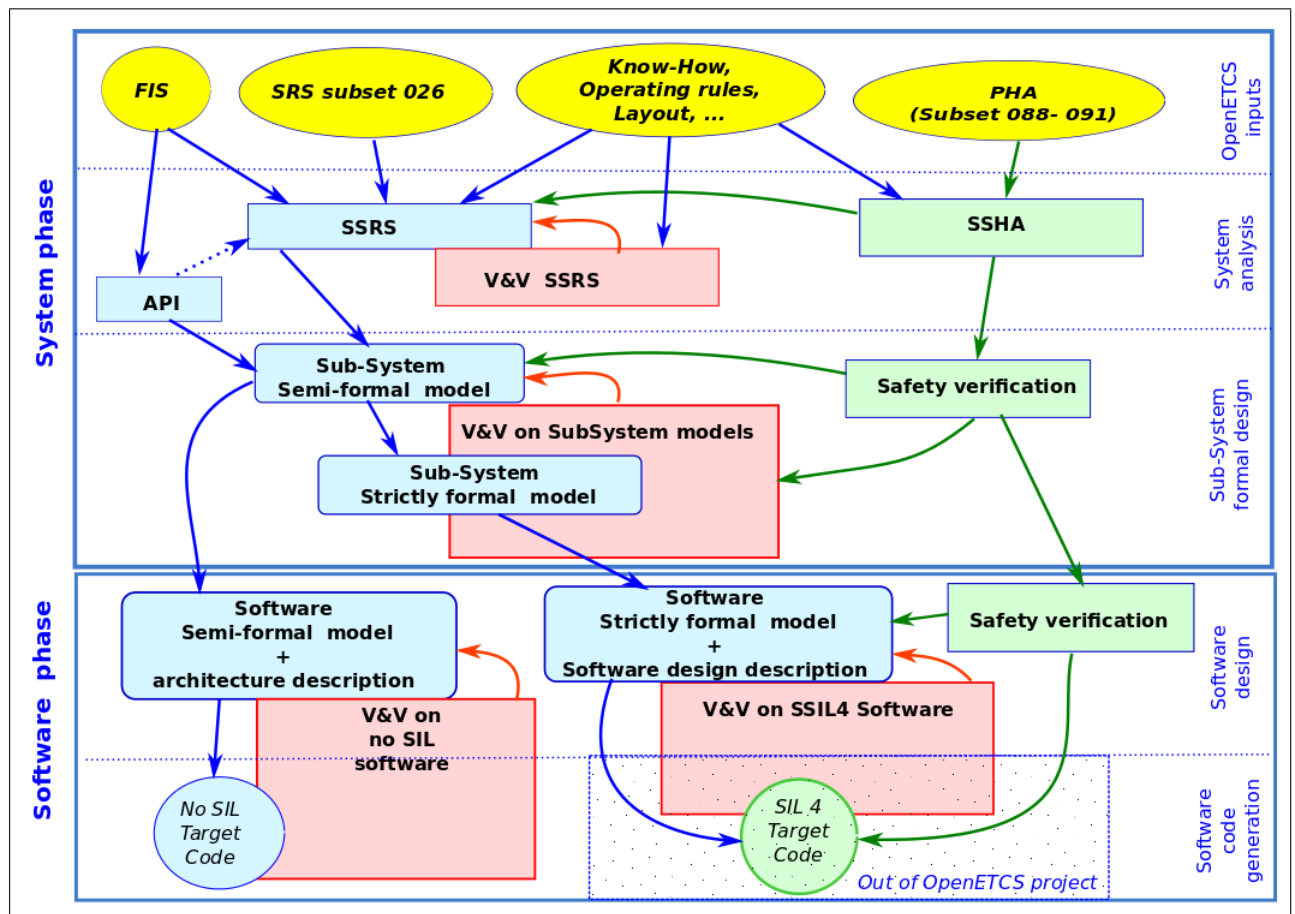


Figure 1. Main OpenETCS process

**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 Enterprise 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 [? ].

### 2.2 Evaluation results

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

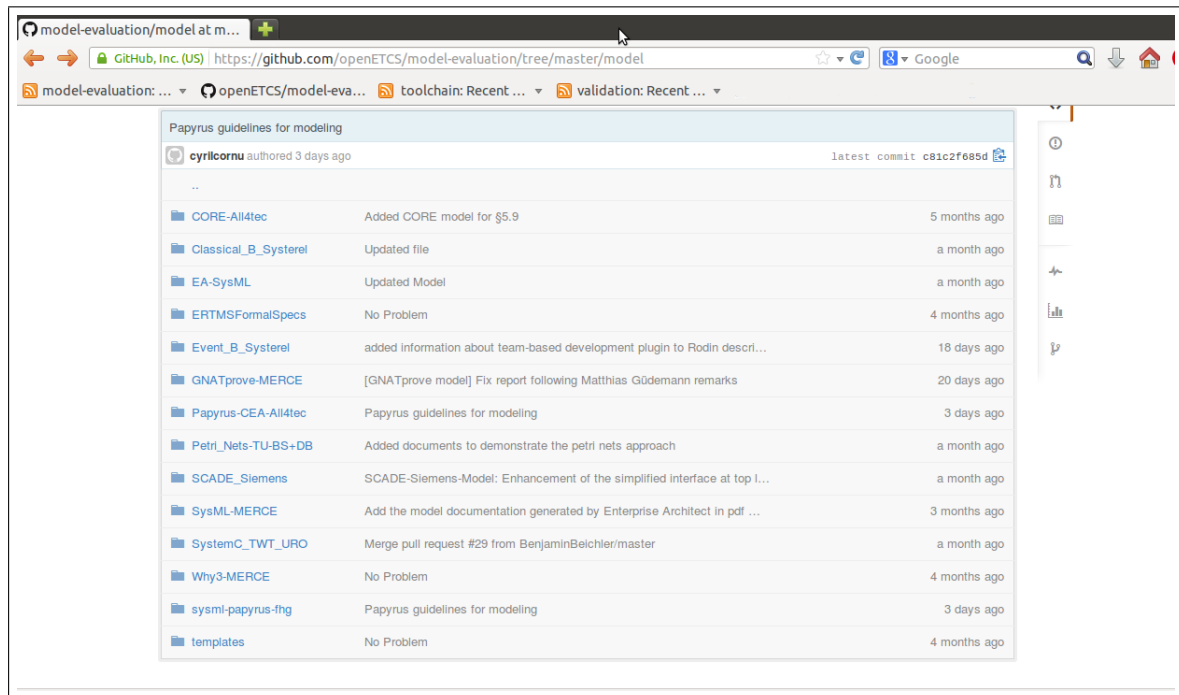


Figure 2. Repository of models

	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 [? ], shows that all evaluated approaches, except GnatProve, are adapted to modeling and design activities:

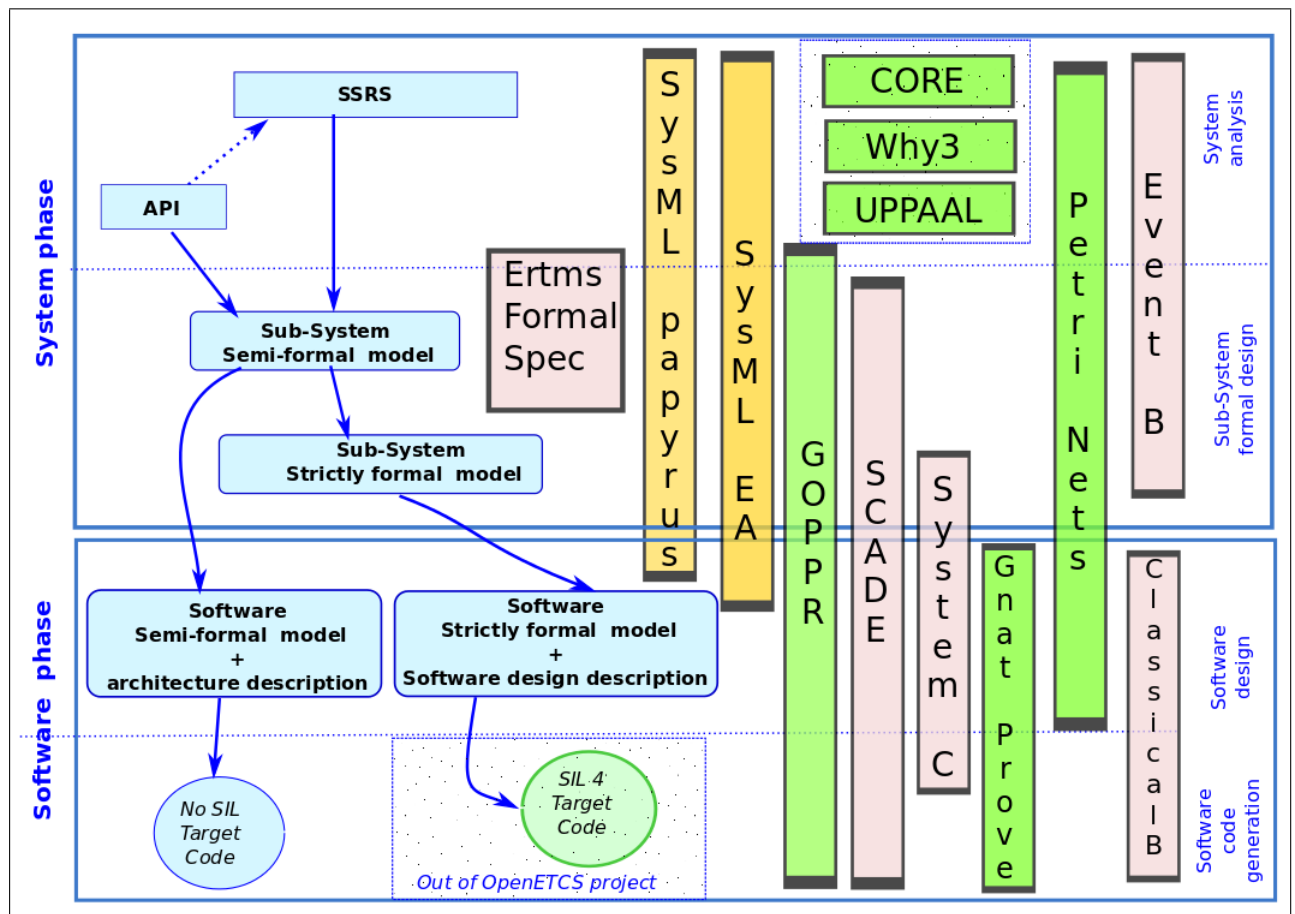


Figure 3. Results of candidates

	GOPRR	ERTMS Formal Specs	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 to this result, GantProve has been proposed to join 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*

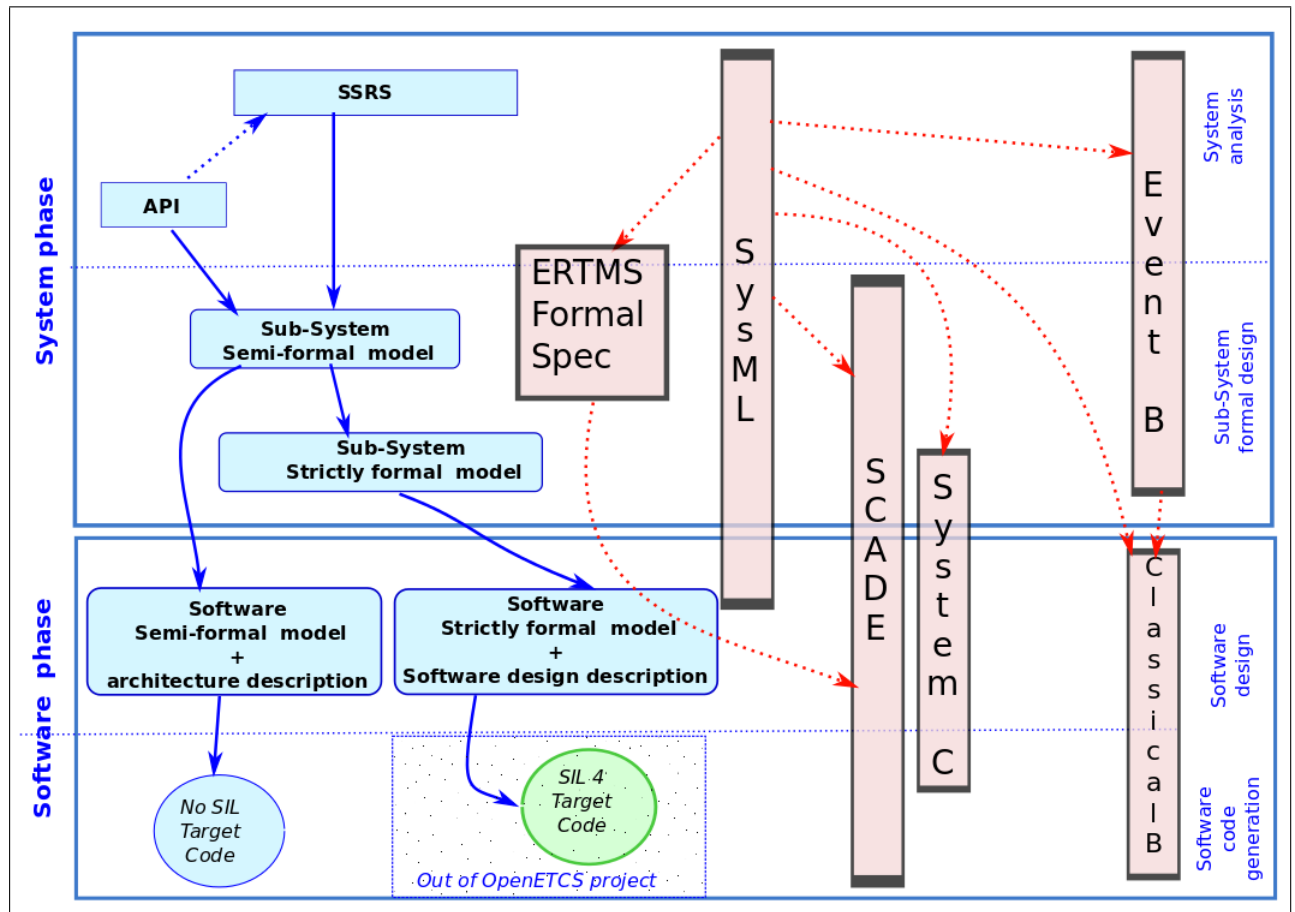


Figure 4. Short list of candidates



## 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 discarded because they do not comply to our tool platform definition. RTP-CESAR was also discarded, the maturity of this project is not yet usable. Finally, **Eclipse with the modeling framework (EMF)** has been chosen as a tool platform, the possibility to use Polarsys and take some part of the TopCased tool chain as well as which version of Eclipse and EMF to choose are discussed in the next sections.

Eclipse can also integrate framework, It has also been decided that any framework added to the Eclipse platform within the OpenETCS tool chain should be documented (version, usage ...) and clearly justify.

### 3.2 Eclipse

Eclipse is an open source Tool Platform originally developed at IBM. It has been explicitly designed as an extensible platform to enable different tools to exchange data and share common functionality. Additionally Eclipse is a rich open source ecosystem with a variety of frameworks for different purposes, such as versioning, code generation, language support and many more. The Eclipse Modeling Framework (EMF) as a top level project bundles all modeling frameworks at Eclipse. Additionally it technically provides a common data format for modeling purposes. Originally it has implemented the OMG Standard Meta-Object Facility (MOF) and has then been reduced to the OMG standard essential MOF (eMOF). EMF provides a model-driven approach to develop modeling languages. It allows to define custom meta-models and generate code from them. Additionally it provides common features such as command-based editing and XMI serialization for generated models. In the following we show how Eclipse and EMF aligns with the openETCS requirements.

## Open Source

All Eclipse core components including EMF are open source and under the Eclipse Public licenses, which allows for commercial use and is compatible to the EUPL. The Eclipse Foundation and the Eclipse Development process assure the management of the intellectual properties for all Eclipse projects. Additionally all Eclipse projects follow a common infrastructure and process allowing external partners to contribute and maintain projects.

## Long-Term Maintenance

The Eclipse Foundation also provides infrastructure and a process for Long-Term Maintenance for all Eclipse projects. It enables users of a technology to contract service providers to maintain current and older versions of these technologies. These service providers do not necessarily have to be committers on the original projects.

## Portability

Eclipse itself is implemented in Java and therefore portable to all major operating systems. The underlying UI technology SWT is implemented for all major and even most uncommon window kits. As SWT uses native widgets, the performance of the UI is close to native applications. The Eclipse Java IDE has a user base of several million developers, which ensures, that the platform runs stable on the supported platforms. Since version 4.2, EMF is part of the core platform. However, EMF does not contain any OS specific components and is therefore highly portable.

## Tools Interoperability

The Eclipse Platform has been explicitly designed to enable various tools of the software lifecycle to collaborate. It provides mechanisms, such as a service oriented architecture and extension points to enable the communication between different parts of a tool chain. EMF is well-suited as a common data-format. The collaboration of a large number of tools is shown and validated in the various Eclipse packages, which are released in the yearly release train.

## Modularity

Eclipse is based on OSGi, a standard for modularization of Java applications. The Eclipse OSGi runtime Equinox is the reference implementation of OSGi. OSGi enables to modularize a system, in this case the tool chain. Additionally it allows to specify the API of modules and the dependencies between them. Additionally, the existing platform provides many possibilities to be extended by new features. The extensibility and OSGi as an underlying technology allow fully customizing the Eclipse Platform. Existing pieces and frameworks can be added to a tool chain, new parts can be developed.

## Framework Support

Over the last ten years, a rich ecosystem of frameworks has developed around the Eclipse Platform. All these frameworks are developed under the EPL and checked for IP cleanliness. Eclipse frameworks cover all different kinds of purposes, however there is a strong focus in support for tool development and modeling. Modeling technologies are almost all compatible with EMF as a common data format. Technologies provided by Eclipse projects include:

1. Textual Modeling and DSL (e.g. Xtext)

2. Language Support (e.g. CDT, JDT)
3. Source Code Versioning Clients (e.g. Egit, Subclipse, Subversion)
4. Model Repositories and Versioning (e.g. EMFCompare, EMF Diff/merge, EMFStore and CDO)
5. Code Generation (e.g. Xpand, Xtend)
6. Model Transformation (e.g. ATL, QVT)
7. Model Development Tools (e.g. Papyrus, OCL, RMF, Sphinx, eTrice)
8. Graphical Modeling (e.g. Graphiti, GMF)
9. User Interfaces (e.g. JFace, Databinding, EMF Client Platform, EEF)
10. ALM Tooling (e.g. Mylyn)

### 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 Papyrus 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

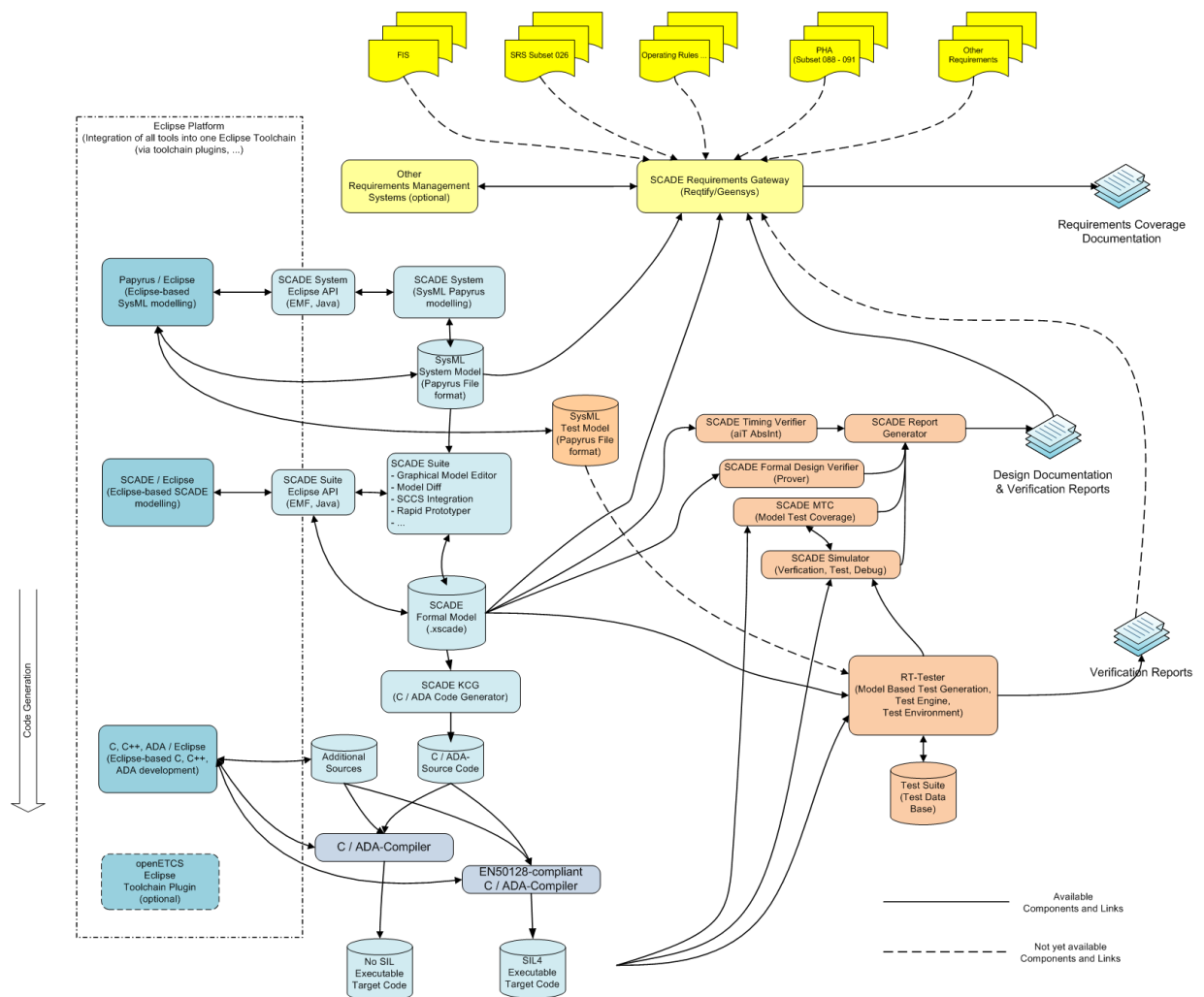
*Todo: MPD: list of on going propositions to cover low levels of design process*

### 4.4 Conclusion

## 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 and 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, verification 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 indispensable 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 worthwhile 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 Shortcomings versus OpenETCS requirements

*Todo: Discuss the shortcomings 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 Shortcomings versus OpenETCS requirements

*Todo: Discuss the shortcomings 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 Shortcomings versus OpenETCS requirements

*Todo: Discuss the shortcomings 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