openETCS

OETCS/WP7/O7.1.3_O7.1.7 – 00/01

ITEA2 Project
Call 6 11025
2012 – 2015

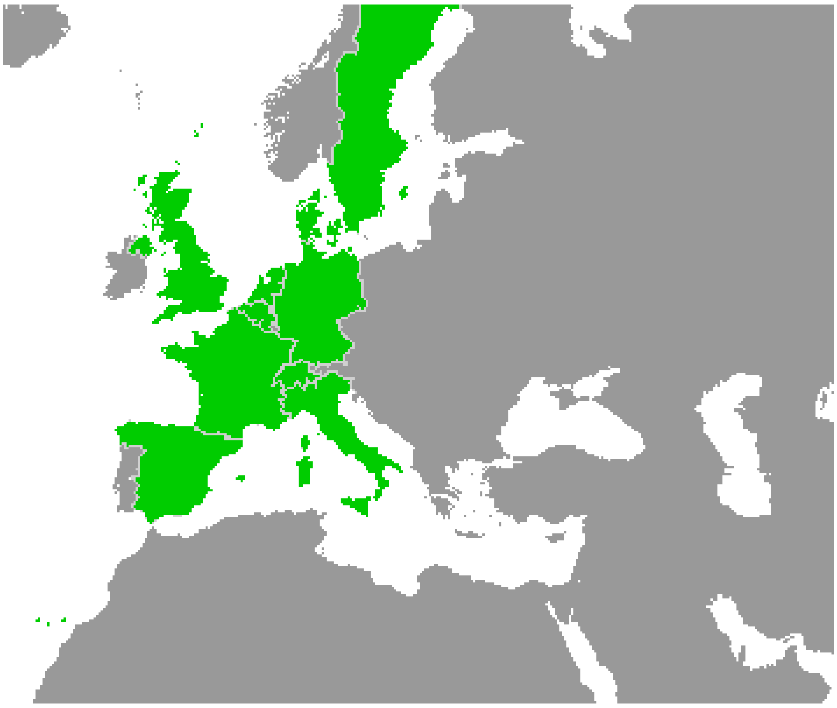
Work-Package 7: “Primary tool chain”

Evaluation of the models and tools against the WP2 requirements

List of criteria on means, models and tools and results on the benchmark

Marielle Petit-Doche

April 2013



This page is intentionally left blank

Work-Package 7: “Primary tool chain”

OETCS/WP7/O7.1.3_O7.1.7 – 00/01
April 2013

Evaluation of the models and tools against the WP2 requirements

List of criteria on means, models and tools and results on the benchmark

Marielle Petit-Doche

Systerel

Definition

Prepared for ITEA2 openETCS consortium
Europa

Abstract: This document gives elements to evaluate the means of modeling and the associated tools according WP2 requirements. Evaluation on the models and tools of benchmark is also described.

Disclaimer: This work is licensed under the European Union Public Licence (EUPL v.1.1) and a Creative Commons Attribution-ShareAlike 3.0 – (cc by-sa 3.0)

Table of Contents

Figures and Tables

Figures

Tables

Document information	
Work Package	WP7
Deliverable ID or doc. ref.	O7.3.1_O7.1.7
Document title	Evaluation of the models and toolsagainst the WP2 requirements
Document version	00.01
Document authors (org.)	Marielle Petit-Doche (Systerel)

Review information	
Last version reviewed	00.00.00
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
00.00.01	19/04/2013	M. Petit-Doche	Document creation by merging O7.1.3 and O7.1.7

1 Introduction

The aim of this document is to report the results of the evaluation of the means of description to model the requirements of SUBSET-026 concerning the on-board unit and their assocaited tools.

This evaluation task is part on work package WP7, task 1 "Primary tool Chain analyses and recommendations". According to the results of WP2, especially the OpenETCS process and the requirements on language, the aim of this task is to determine the best candidates to produce models of the on-board units, following the OpenETCS process

This process is described in détail in D2.3 " Description of the openETCS process" and is summed up in the figure 1.

Yellow elements are inputs, blue elements are part of the design process, red elements are verification and validation activities, green elements are safety activities. Each line (between dash or full blue lines) is a phase of the process, with a name on the right.

The second section of this document provides a template to describe the means and tools and a list of criteria according WP2 requirements on language, models and tools. The objectives of this description and criteria are to allow to determine the best means of description and associated tool for a given activities.

The third section resumes the results of the evaluation at the end of the benchmark activities.

In Appendix, a section is dedicated to each models produced during the benchmark activities :

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

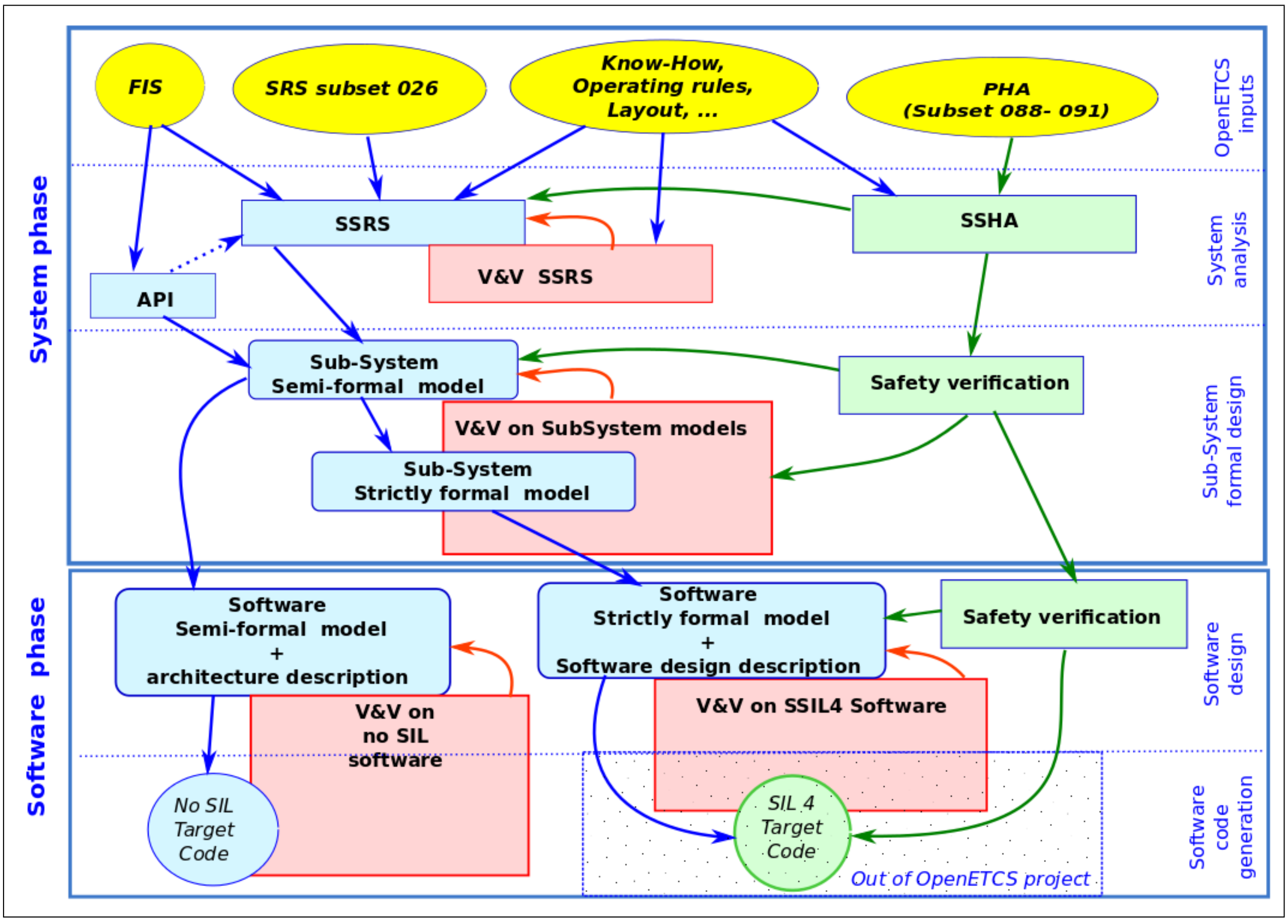


Figure 1. Main OpenETCS process

Tool platform are not covered by this document but in an other output of WP7 : O7.1.9 "Evaluation of each tool platform against WP2 requirements, independent of target tools". Besides, Task 7.1 is focussing on design activities : despite that some means can provide verification artefacts for example, tools and means for validation, verification, test generation,... are in the scope of task 2 and will be analysed later.

2 Templates

Author Author of the approaches description %%Name - Company%%

Assessor 1 First assessor of the approaches %%Name - Company%%

Assessor 2 Second assessor of the approaches %%Name - Company%%

In the sequel, main text is under the responsibilities of the author.

Author. Author can add comments using this format

Assessor 1. First assessor can add comments using this format

Assessor 2. Second assessor can add comments using this format

When a note is required, please follow this list :

0 not recommended, not adapted, rejected

1 weakly recommended, adapted after major improvements, weakly rejected

2 recommended, adapted (with light improvements if necessary) weakly accepted

3 highly recommended, well adapted,strongly accepted

***** difficult to evaluate with a note (please add a comment under the table)

All the notes can be commented under each table.

2.1 Presentation

This section gives a quick presentation of the approach and the tool.

Name Name of the approach and the tool

Web site if available, how to find information

Licence Kind of licence

Abstract

Short abstract on the approach and tool (10 lines max)

Publications

Short list of publications on the approach (5 max)

2.2 Main usage of the approach

This section discusses the main usage of the approach.

According to the figure 1, for which phases do you recommend the approach (give a note from 0 to 3) :

	Author	Assessor 1	Assessor 2	Total
System Analysis				
Sub-system formal design				
Software design				
Software code generation				

According to the figure 1, for which type of activities do you recommend the approach (give a note from 0 to 3) :

	Author	Assessor 1	Assessor 2	Total
Documentation				
Modeling				
Design				
Code generation				
Verification				
Validation				
Safety analyses				

Known usages

Have you some examples of usage of this approach to compare with the OpenETCS objectives ?

2.3 Language

This section discusses the main element of the language.

According WP2 requirements, give a note for the characteristics of the language (from 0 to 3) :

	Author	Assessor 1	Assessor 2	Total
Informal language				
Semi-formal language				
Formal language				
Structured language				
Modular language				
Extensible language (D.2.6-01-28)				
Textual language				
Mathematical symbols or code				
Graphical language				
Declarative and simple formalization of properties (D.2.6-X-27)				
Scalability : capability to design large model				
Easily translatable to other languages (D.2.6-X-28)				
Executable				
Simulation, animation				
Easily understandable (D.2.6-X-26)				
Expertise level needed (0 High level, 3 few level)				
Standardization				

Documentation

Describe how the language is documented, the existing guidelines, coding rules, standardization...

Language usage

Describe the possible restriction on the language

2.4 System Analysis

This section discusses the usage of the approach for system analysis. It can be skipped depending the results of 2.8.

Acoording WP2 requirements, how the approach can be involved for the sub-system requirement specification ?

	Author	Assessor 1	Assessor 2	Total
Independent System functions definition (D.2.6-X-10.1.1)				
System architecture design (D.2.6-X-10.1.2)				
System data flow identification (D.2.6-X-10.1.3)				
Sub-system focus (D.2.6-X-10.1.4)				
System interfaces definition (D.2.6-X-10.1.5)				
System requirement allocation (D.2.6-X-10.2)				
Traceability with SRS (D.2.6-X-10.3)				
Traceability with Safety activities (D.2.6-X-11)				

2.5 Sub-System formal design

This section discusses the usage of the approach for sub-system formal design. It can be skipped depending the results of 2.8.

2.5.1 Semi-formal model

Concerning semi-formal model, how the WP2 requirements are covered ?

	Author	Assessor 1	Assessor 2	Total
Consistency to SSRS (D.2.6-X-12.2)				
Coverage of SSRS (D.2.6-X-12.2.1)				
Traceability to SSRS (D.2.6-X-12.3)				
Simulation or animation (D.2.6-X-13 partial)				
Execution (D.2.6-X-13 partial)				
Extensible to strictly formal model (D.2.6-X-14.3)				
Easy to refine towards strictly formal model (D.2.6-X-14.4)				
Extensible and modular design (D.2.6-X-15)				
Extensible to software design (???)				
Safety properties formalisation (D.2.6-01-20)				
Safety properties validation (D.2.6-X-22 partial)				
Logical properties assertion (D.2.6-X-32)				
Check of assertions (D.2.6-X-32.1)				

Does the language allow to formalize (D.2.6-X-29):

	Author	Assessor 1	Assessor 2	Total
State machines				
Time-outs				
Truth tables				
Arithmetic				
Braking curves				
Logical statements				
Message and fields				

Additional comments on semi-formal model

Do you think your semi-formal model is sufficient to cover a safe design of the on-board unit until code generation ? All comments on links to other models, validation and verification activities are welcomed.

2.5.2 Strictly formal model

Concerning strictly formal model, how the WP2 requirements are covered ?

	Author	Assessor 1	Assessor 2	Total
Consistency to SFM (D.2.6-X-14.2)				
Coverage of SSRS (D.2.6-X-14.2)				
Traceability to SSRS (D.2.6-X-14.3)				
Extensible to software design (D.2.6-X-16)				
Safety properties formalisation (D.2.6-01-20)				
Safety properties validation (D.2.6-X-22 partial)				
Logical properties assertion (D.2.6-X-32)				
Check of assertions (D.2.6-X-32.2)				

Does the language allow to formalize (D.2.6-X-30):

	Author	Assessor 1	Assessor 2	Total
State machines				
Time-outs				
Truth tables				
Arithmetic				
Braking curves				
Logical statements				
Message and fields				

Additional comments on semi-formal model

Do you think your strictly formal model can be directly defined from the SSRS ? All comments on links to other models, validation and verification activities are welcomed.

2.6 Software design

This section discusses the usage of the approach for software design. It can be skipped depending the results of 2.8.

2.6.1 Functional design

How the approach allows to produce a functional software model of the on-board unit ?

	Author	Assessor 1	Assessor 2	Total
Derivation from system semi-formal model				
Software architecture description				
Software constraints				
Traceability				
Executable				

2.6.2 SSIL4 design

How the approach allows to produce in safety a software model ?

	Author	Assessor 1	Assessor 2	Total
Derivation from system semi-formal or strictly formal model				
Software architecture description				
Software constraints				
Traceability				
Executable				
Conformance to EN50128 § 7.2				
Conformance to EN50128 § 7.3				
Conformance to EN50128 § 7.4				

Which criteria for software architecture are covered by the methodology (see EN50128 table A.3) :

	Author	Assessor 1	Assessor 2	Total
Defensive programming				
Fault detection & diagnostic				
Error detecting code				
Failure assertion programming				
Diverse programming				
Memorising executed cases				
Software error effect analysis				
Fully defined interface				
Modelling				
Structured methodology				

2.7 Software code generation

This section discusses the usage of the approach for software code generation. It can be skipped depending the results of 2.8.

Which criteria for software design and implementation are covered by the methodology (see EN50128 table A.4) :

	Author	Assessor 1	Assessor 2	Total
Formal methods				
Modeling				
Modular approach (mandatory)				
Components				
Design and coding standards (mandatory)				
Strongly typed programming language				

2.8 Main usage of the tool

This section discusses the main usage of the tool.

Which task are covered by the tool ?

	Author	Assessor 1	Assessor 2	Total
Modelling support				
Automatic translation				
Code Generation				
Model verification				
Test generation				
Simulation, execution, debugging				
Formal proof				

Modelling support

Does the tool provide a textual or a graphical editor ?

Automatic translation and code generation

Which translation or code generation is supported by the tool ?

Model verification

Which verification on models are provided by the tool?

Test generation

Does the tool allow to generate tests ? For which purpose ?

Simulation, execution, debugging

Does the tool allow to simulate or to debbug step by step a model or a code ?

Formal proof

Does the tool allow formal proof ? How ?

2.9 Use of the tool

According WP2 requirements, give a note for characteristics of the use of the tool (from 0 to 3) :

	Author	Assessor 1	Assessor 2	Total
Open Source (D2.6-01-029)				
Portability to operating systems (D2.6-01-030)				
Cooperation of tools (D2.6-01-031)				
Robustness (D2.6-01-034)				
Modularity (D2.6-01-034.01)				
Documentation management (D.2.6-01-034.02)				
Distributed software development (D.2.6-01-034.03)				
Issue tracking (D.2.6-01-034.04)				
Differences between models (D.2.6-01-034.05)				
Version management (D.2.6-01-034.06)				
Concurrent version management (D.2.6-01-034.07)				
Model-based version control (D.2.6-01-034.08)				
Role traceability (D.2.6-01-034.09)				
Safety version traceability (D.2.6-01-034.10)				
Model traceability (D.2.6-01-035)				
Tool chain integration				
Scalability				

2.10 Certifiability

This section discusses how the tool can be classified according EN50128 requirements (D.2.6-X-49).

	Author	Assessor 1	Assessor 2	Total
Tool manual (D.2.6-01-42.02)				
Proof of correctness (D.2.6-01-42.03)				
Existing industrial usage				
Model verification				
Test generation				
Simulation, execution, debugging				
Formal proof				

Other elements for tool certification

2.11 Other comments

Please to give free comments on the approach.

3 Conclusion

%%To Be Defined%%

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