OETCS/WP7/O7.2.1 – 00/05

openETCS

ITEA2 Project
Call 6 11025
2012 – 2015

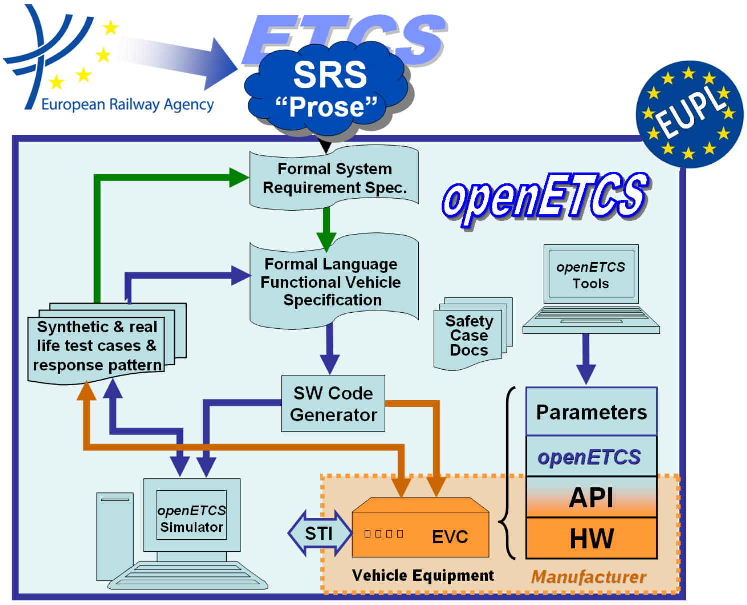
Work-Package 7: “Secondary tools - Verification and Validation ”

Evaluation of supporting tools and methods against the WP2 requirements and task 1

Means and tools for Verification and Validation

Marielle Petit-Doche, all participants of the benchmark and all participants of VnV and Safety process

October 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: "Secondary tools - Verification and Validation "

OETCS/WP7/O7.2.1 – 00/05
October 2013

Evaluation of supporting tools and methods against the WP2 requirements and task 1

Means and tools for Verification and Validation

Marielle Petit-Doche
Systerel

all participants of the benchmark
WP7 partners

all participants of VnV and Safety process
WP4 partners

Evaluation

Prepared for openETCS@ITEA2 Project

Abstract: This document gives elements to evaluate the tools and methods to complete the primary toolchain and to support verification and validation activities, safety activities, moodel transformation and data management for the whole project. Evaluation on the means and tools of benchmark is also described.

Table of Contents

Figures and Tables

Figures

Tables

Document information	
Work Package Deliverable ID or doc. ref.	WP7 O7.2.1
Document title Document version Document authors (org.)	Evaluation of supporting tools and methods against the WP2 requirements and task 1 - Vnv 00.05 Marielle Petit-Doche (Systerel)

Review information	
Last version reviewed	00.04
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	19/07/2013	M. Petit-Doche	Document creation
00.02	09/09/2013	M. Petit-Doche	Major evolutions in all document
00.03	19/09/2013	M. Petit-Doche	Issues: 167, 168, 170
00.04	23/09/2013	M. Petit-Doche	Issues: 164, 169, 174, 175, 177, 178
00.05	01/10/2013	M. Petit-Doche	Split of document O7.2.1. Verification and Validation part
00.06	18/10/2013	M. Petit-Doche	Issues: 174, 178, 167
00.07	08/11/2013	M. Petit-Doche	Issues: 177, 179, 176, 180

1 Introduction

The aim of this document is to report the results of the evaluation of means and tools for the secondary means and tools, i.e. the means and tools which complete the primary tool chain dedicated to formal model and software design.

This evaluation task is part of work package WP7, task 2 "Secondary tools analyses and recommendations". According to the results of WP2, especially the OpenETCS process and the requirements on language and tools [**?**], and the results of T7.1 on the primary toolchain [**?**], the aim of this task is to determine the best candidates to complete and support the primary toolchain for the following activities:

- verification and validation (WP4)
- safety activities support (WP4)
- data, function and requirement management (SSRS, WP3 and WP4)
- model transformation and code generation (WP3 and WP4)

This document is dedicated to tools and means for verification and validation.

1.1 Organisation of the document

The chapter 2 provides a template to describe the means and tools and a list of criteria according WP2 requirements on language, models and tools, and T7.1 primary tool chain decision. 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 chapter 3 resumes the results of the evaluation at the end of the benchmark activities.

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

- Scade Suite
- System C
- UPPAAL
- Rodin and Pluggins
- Tools around Classical B (ProB, SMT solver,...)
- CPN tools
- Matelo
- RT-Tester

- Fiacre and Tina
- Frama-C
- Diversity

2 Template

2.1 Instructions

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 at any place.

Assessor 1: First assessor can add comments using this format at any place.

Assessor 2: Second assessor can add comments using this format at any place.

When a note is required, please follow this list (inspired from Technology Readiness Level, see `http://en.wikipedia.org/wiki/Technology_readiness_level`) :

0 not recommended / rejected / no integration possible or valuable / not adapted for this topic / not available for this topic

1 weakly recommended / adapted after major improvements / weakly rejected / concept of integration roughly defined / adapted after major improvements / available after major developments

2 recommended / adapted (with light improvements if necessary) weakly accepted / integration prototyped or defined in details / adapted after small improvements / available after small developments or tests

3 highly recommended / well adapted / strongly accepted / integration done and tested / well adapted to the purpose / available and suitable for the purpose All the notes can be commented under each table.

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

All the notes can be commented under each table.

This section defines the criteria for the means and tools dedicated to verification and validation activities, in the WP4 workpackage.

Criteria of this section are defined according [**?**].

2.2 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)

2.3 Common criteria on secondary means and tools

This section discusses the common criteria of the means and tools according to the project requirements on tools and the results of T7.1.

2.3.1 Project and WP2 requirements

The objectives of this list of criteria is to check if the proposed means and tools meet the main criteria of the project: open-source approaches, usability, modularity, coverage of the objectives,...

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 (D2.6-02-074)				
Portability to operating systems (D2.6-02-075)				
Cooperation of tools (D2.6-02-076)				
Robustness (D2.6-02-078)				
Modularity (D2.6-02-078.1)				
Documentation management (D2.6-02-078.02)				
Distributed software development (D2.6-02-078.03)				
Simultaneous multi-users (D2.6-02-078.04)				
Issue tracking (D2.6-02-078.05)				
Differences between models (D2.6-02-078.06)				
Version management (D2.6-02-078.07)				
Concurrent version development (D2.6-02-078.08)				
Model-based version control (D2.6-02-078.09)				
Role traceability (D2.6-02-078.10)				
Safety version traceability (D2.6-02-078.11)				
Model traceability (D2.6-02-079)				
Tool chain integration				
Scalability				
User Friendliness				

2.3.2 Qualification

This section discusses how the tool can be classified according EN50128 requirements (D2.6-02-085). Some qualification shall be mandatory if the tool is involved to design a SIL4 software.

	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				

Which level of tool qualification has been reached or will be reached within the next year ?

Score :

3 already qualified for this level

2 qualification possible to this level, but some elements shall be provided

0 qualification not recommended for this level

	Author	Assessor 1	Assessor 2	Total
class T1				
class T2				
class T3				

Other elements for tool certification

2.3.3 Complementarity with primary toolchain

The objectives of this list of criteria is to check if the proposed means and tools can be easily integrated to the primary toolchain.

2.3.3.1 Language

According to the decisions and the propositions of T7.1, how the mean and approach can be adapted to or can complete the chosen language and methods:

	Author	Assessor 1	Assessor 2	Total
SysML				
Scade method				
EFS language				
B Method				
C language				

SysML

How the means or tools can complete SysML ?

Scade, EFS, Classical B

How the means or tools can complete the current proposals for formal modeling language ?

C language

How the means or tools can complete or be adapted to SIL4 software in C language ?

2.3.3.2 Tools and platforms

According to the decisions and the propositions of T7.1, how the mean and approach can be integrated to or can complete the chosen tools and platforms:

	Author	Assessor 1	Assessor 2	Total
Eclipse				
Papyrus				
Scade				
EFS tools				
B tools				

Eclipse

How the means or tools can be integrated to the Eclipse platform ?

Papyrus

How the means or tools can complete Papyrus ?

Scade, EFS, Classical B

How the means or tools can complete the current proposals for formal modeling tools ?

2.4 VnV Activities

The VnV activities are described in details in the verification and Validation Plan [?].

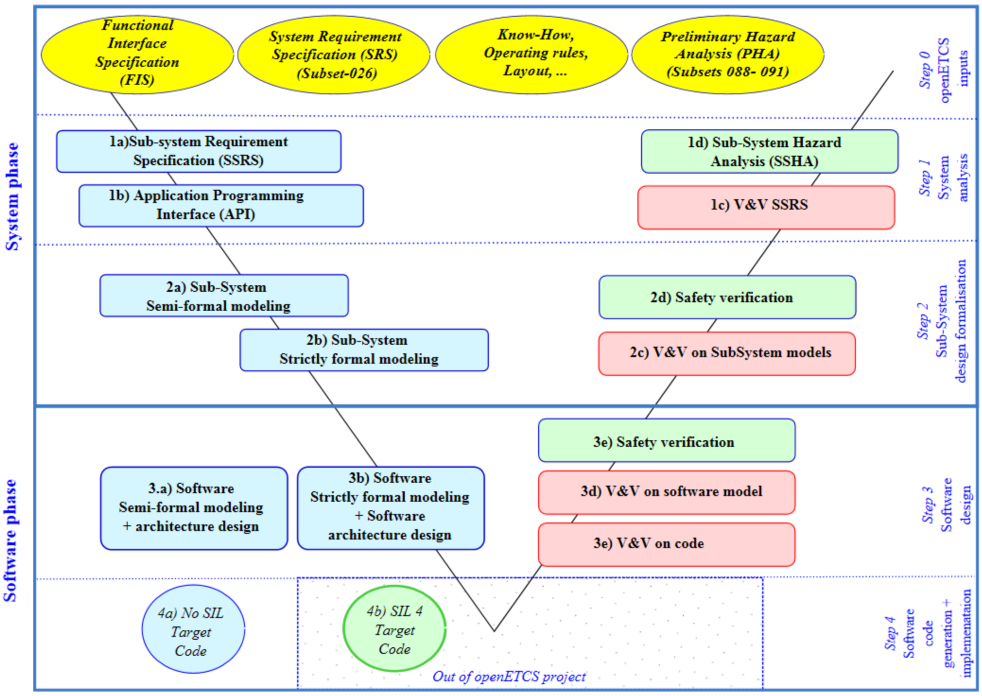

Functional Interface Specification (FIS)
System Requirement Specification (SRS) (Subset-026)
Know-How, Operating rules, Layout, ...
Preliminary Hazard Analysis (PHA) (Subsets 088- 091)
Step 0 openETCS inputs
System phase
1a)Sub-system Requirement Specification (SSRS)
1b) Application Programming Interface (API)
1d) Sub-System Hazard Analysis (SSHA)
1c) V&V SSRS
Step 1 System analysis
2a) Sub-System Semi-formal modeling
2b) Sub-System Strictly formal modeling
2d) Safety verification
2c) V&V on SubSystem models
Step 2 Sub-System design formalisation
Software phase
3.a) Software Semi-formal modeling + architecture design
3b) Software Strictly formal modeling + Software architecture design
3e) Safety verification
3d) V&V on software model
3e) V&V on code
Step 3 Software design
4a) No SIL Target Code
4b) SIL 4 Target Code
Out of openETCS project
Step 4 Software code generation + implemenataion

Figure 1. openETCS Process (rough view)

According figure L1, for which activities is the mean or tool suitable (see also [?] section 5.1.2 for more details)[1] ?

[1] DAS2V : Design Artifact Subject to Verification and Validation, see [?]

	Author	Assessor 1	Assessor 2	Total
1c SSRS Verification				
1c SSRS Validation				
2c SFM Verification				
2c SFM Validation				
3d SW-SFM Verification				
3d SW-SFM Validation				
3d SW-FFM Verification				
3d SW-FFM Validation				
3e Code Verification				
3e Code Validation				
DAS2V Verification				
DAS2V Validation				
Automatic model transformation verification				
Automatic code generation verification				

2.5 Properties

Which kind of properties or elements are verified or validated by the mean or tool (see also [**?**] section 4) ?

	Author	Assessor 1	Assessor 2	Total
Functionalities of the system and sub-system				
System and sub-system architecture				
External and internal interfaces of sub-system				
Software components				
Performance constraints				
Safety objectives				
Functional properties				
Safety properties				

2.6 Verification methods and tools

Which kind of methods is proposed (see also [**?**] section 5.3) ?

	Author	Assessor 1	Assessor 2	Total
Reviews				
Inspections				
Software Architecture Analysis Method				
Architecture Tradeoff Analysis Method				
Model-Based System Integration Testing				
Model-Based Testing of Generated High-Level Code				
Abstract Interpretation				
Deductive Verification				
Model Checking				
Correct by Construction Formal Methods				
Verification with Formal Methods				
Simulation-based				

2.7 Validation means and tools

The following list of criteria focuss on means and tools to support validation activities, according WP2 requirements :

	Author	Assessor 1	Assessor 2	Total
Simulation-based				
Step-by-step simulation (D2.6-01-036)				
Environment emulation (D2.6-01-037 and D2.6-02-080)				
Time-based test case (D2.6-02-081)				
Test cases writing (D2.6-01-038)				
Test cases execution (D2.6-01-038)				
Test cases storage (D2.6-01-038)				
Version management of test cases (D2.6-02-082)				
Test generation from independant test model (D2.6-02-083)				
Test sequences writing (D2.6-02-084)				
Test sequences execution (D2.6-02-084)				
Test sequences storage (D2.6-02-084)				

2.8 VnV artifacts

Concerning the artifacts used or produced by the mean or tool, please to detail:

Input

Which is the list of the input artifacts for the mean or tools ?

Output

Which is the list of the output artifacts for the mean or tools ?

Syntax

Which are the reference documents which give a description of the artifacts syntax ?

Semantic

Which are the reference documents which give a description of the artifacts semantic ?

Integration

How these artifacts can be integrated with the elements of the toolchain (language, mangement,...) ?

2.9 Detailled Criterias for VnV

Please fill only the section concerning the proposed mean or tool, other section can be skipped (see issue `https://github.com/openETCS/toolchain/issues/180` for details and discussions)

2.9.1 System Modelling simulation

	Author	Assessor 1	Assessor 2	Total
User Scenario Modelling				
Test Case Modelling				
Test Sequence Modelling				

2.9.2 System Model Verification

	Author	Assessor 1	Assessor 2	Total
Input/ Output checking				
System Behavior Simulation (Mathematical)				
System Behavior Simulation (Animated)				

2.9.3 Software Model Verification

	Author	Assessor 1	Assessor 2	Total
Static Model Verification				
Property Proofing				
Dynamic Testing				
Automatic Test Generation				
Input/ Output checking				
Software Behaviour Simulation (Mathematical)				
Software Behaviour Simulation (Animated)				

2.9.4 Source Code

	Author	Assessor 1	Assessor 2	Total
Traceability to Model				

2.9.5 Code Verification

	Author	Assessor 1	Assessor 2	Total
Formal Proof				
Programming by contract				
Static Analysis				
Dynamic Analysis				
Dynamic Testing				
Automatic Test Generation				
Performance Testing				
Interface Testing				

2.9.6 Validation System/Software/Code/ Validation

	Author	Assessor 1	Assessor 2	Total
Test Coverage				
Use Case Validation of Model				
Functional or Black-box Testing				
User Scenario Testing				
Traceability				
Schedulability Analyzer / UseCase Check all				
Schedulability Analyzer / UseCase Check single mode				

2.10 Other comments

Comment. This section is available for the author or the assessors to complete the description and criteria.

3 Conclusion

The process of evaluation of secondary tools has evolved during the task: means and tools have been presented to all the partners but partly evaluated. Partners decided to based the evaluation and selection on the needs which are raised during the development of the toolchain or its use in the OpenETCS project.

In Appendix there are some results of the evaluation.

Minus mark "-" means this criteria as not been evaluated for this approach.

Star mark "*" means this criteria has been difficult to evaluate for this approach.

The highest score is **9** and means that the criteria is fully respected, the lowest score is 0.

Appendix A: Scade

No results of evaluation.

Appendix B: SystemC

B.1 Instructions

Author Alexander Nitsch (URO), Benjamin Beichler (URO), Stefan Rieger (TWT)

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 at any place.

Assessor 1: First assessor can add comments using this format at any place.

Assessor 2: Second assessor can add comments using this format at any place.

When a note is required, please follow this list (inspired from Technology Readiness Level, see `http://en.wikipedia.org/wiki/Technology_readiness_level`) :

0 not recommended / rejected / no integration possible or valuable / not adapted for this topic / not available for this topic

1 weakly recommended / adapted after major improvements / weakly rejected / concept of integration roughly defined / adapted after major improvements / available after major developments

2 recommended / adapted (with light improvements if necessary) weakly accepted / integration prototyped or defined in details / adapted after small improvements / available after small developments or tests

3 highly recommended / well adapted / strongly accepted / integration done and tested / well adapted to the purpose / available and suitable for the purpose All the notes can be commented under each table.

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

All the notes can be commented under each table.

This section defines the criteria for the means and tools dedicated to verification and validation activities, in the WP4 workpackage.

Criteria of this section are defined according [**?**].

B.2 Presentation

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

Name SystemC

Web site `www.accellera.org/downloads/standards/systemc/about_systemc/`

Licence SystemC Open Source License

Abstract

SystemC is a C++ library providing an event-driven simulation interface suitable for electronic system level design. It enables a system designer to simulate concurrent processes. SystemC processes can communicate in a simulated real-time environment, using channels of different datatypes (all C++ types and user defined types are supported). SystemC supports hardware and software synthesis (with the corresponding tools). SystemC models are executable.

Publications

- D. C. Black, SystemC: From the ground up. Springer, 2010.
- IEEE 1666 Standard SystemC Language Reference Manual, `http://standards.ieee.org/getieee/1666/`
- The ITEA MARTES Project, from UML to SystemC, `http://www.martes-itea.org/`
- J. Bhasker, A SystemC Primer, Second Edition, Star Galaxy Publishing, 2004
- F. Ghenassia (Editor), Transaction-Level Modeling with SystemC: TLM Concepts and Applications for Embedded Systems, Springer 2006

B.3 Common criteria on secondary means and tools

This section discusses the common criteria of the means and tools according to the project requirements on tools and the results of T7.1.

B.3.1 Project and WP2 requirements

The objectives of this list of criteria is to check if the proposed means and tools meet the main criteria of the project: open-source approaches, usability, modularity, coverage of the objectives,...

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 (D2.6-02-074)	3			
Portability to operating systems (D2.6-02-075)	3			
Cooperation of tools (D2.6-02-076)	2			
Robustness (D2.6-02-078)	3			
Modularity (D2.6-02-078.1)	3			
Documentation management (D2.6-02-078.02)	2			
Distributed software development (D2.6-02-078.03)	3			
Simultaneous multi-users (D2.6-02-078.04)	3			
Issue tracking (D2.6-02-078.05)	2*			
Differences between models (D2.6-02-078.06)	3			
Version management (D2.6-02-078.07)	3**			
Concurrent version development (D2.6-02-078.08)	3			
Model-based version control (D2.6-02-078.09)	3			
Role traceability (D2.6-02-078.10)	2			
Safety version traceability (D2.6-02-078.11)	2			
Model traceability (D2.6-02-079)	2			
Tool chain integration	3			
Scalability	3			
User Friendliness	3			

Author:

* *Not directly; by means of external tools such as Doxygen (or in the case of issue tracking, e.g., GitHub)*

** *By means of versioning systems such as Git or SVN*

B.3.2 Qualification

This section discusses how the tool can be classified according EN50128 requirements (D2.6-02-085). Some qualification shall be mandatory if the tool is involved to design a SIL4 software.

	Author	Assessor 1	Assessor 2	Total
Tool manual (D.2.6-01-42.02)	3			
Proof of correctness (D.2.6-01-42.03)	1			
Existing industrial usage	3			
Model verification	3			
Test generation	0			
Simulation, execution, debugging	3			
Formal proof	1			

Which level of tool qualification has been reached or will be reached within the next year ?

Score :

3 already qualified for this level

2 qualification possible to this level, but some elements shall be provided

0 qualification not recommended for this level

	Author	Assessor 1	Assessor 2	Total
class T1				
class T2				
class T3				

Other elements for tool certification

B.3.3 Complementarity with primary toolchain

The objectives of this list of criteria is to check if the proposed means and tools can be easily integrated to the primary toolchain.

B.3.3.1 Language

According to the decisions and the propositions of T7.1, how the mean and approach can be adapted to or can complete the chosen language and methods:

	Author	Assessor 1	Assessor 2	Total
SysML	2			
Scade method	2			
EFS language	1			
B Method	1			
C language	3			

SysML

How the means or tools can complete SysML ?

Author:

- *Transformation from SysML, e.g., by using Acceleo*
- *SystemC provide executable models*
- *allows performance evaluation with target hardware*

Scade, EFS, Classical B

How the means or tools can complete the current proposals for formal modeling language ?

Author:

- *SystemC provide executable models*
- *allows performance evaluation with target hardware*
- *providing a SystemC Testenviroment for generated C/C++ code*

C language

How the means or tools can complete or be adapted to SIL4 software in C language ?

Author:

- *allows performance evaluation with target hardware*
- *providing a test environment for generated C/C++ code*

B.3.3.2 Tools and platforms

According to the decisions and the propositions of T7.1, how the mean and approach can be integrated to or can complete the chosen tools and platforms:

	Author	Assessor 1	Assessor 2	Total
Eclipse	3			
Papyrus	1			
Scade	1			
EFS tools	1			
B tools	1			

Eclipse

How the means or tools can be integrated to the Eclipse platform ?

Author:

- *basically modeling with SystemC is the development of C++ code, therefore the CDT tools provide already a good integration in to eclipse*

Papyrus

How the means or tools can complete Papyrus ?

Author:

- *both papyrus and SystemC(CDT) are parts of the eclipse IDE*
- *transformation from SysML to SystemC with Acceleo*

Scade, EFS, Classical B

How the means or tools can complete the current proposals for formal modeling tools ?

Author:

- *due to the widespread usage of C++, many libraries are available for adaptions of other software, e.g. XML parser, json, java bridges, web services, ...*

B.4 VnV Activities

The VnV activities are described in details in the verification and Validation Plan [**?**].

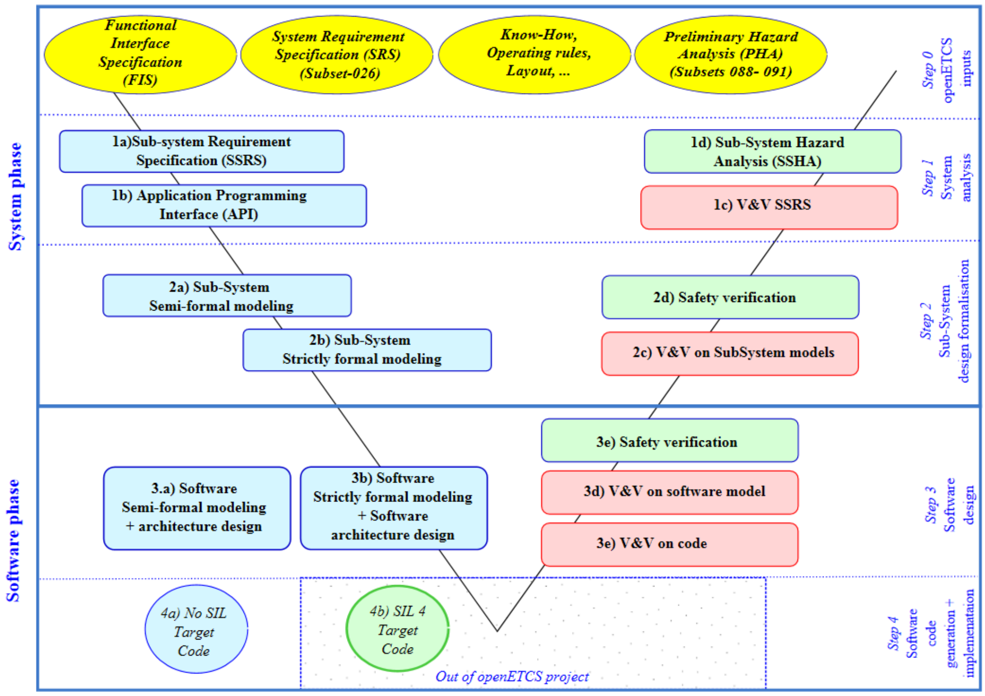

Functional Interface Specification (FIS)
System Requirement Specification (SRS) (Subset-026)
Know-How, Operating rules, Layout, ...
Preliminary Hazard Analysis (PHA) (Subsets 088- 091)
Step 0 openETCS inputs
System phase
1a)Sub-system Requirement Specification (SSRS)
1b) Application Programming Interface (API)
1d) Sub-System Hazard Analysis (SSHA)
1c) V&V SSRS
Step 1 System analysis
2a) Sub-System Semi-formal modeling
2b) Sub-System Strictly formal modeling
2d) Safety verification
2c) V&V on SubSystem models
Step 2 Sub-System design formalisation
Software phase
3.a) Software Semi-formal modeling + architecture design
3b) Software Strictly formal modeling + Software architecture design
3e) Safety verification
3d) V&V on software model
3e) V&V on code
Step 3 Software design
4a) No SIL Target Code
4b) SIL 4 Target Code
Out of openETCS project
Step 4 Software code generation + implementation

Figure B1. openETCS Process (rough view)

According figure L1, for which activities is the mean or tool suitable (see also [**?**] section 5.1.2 for more details)[2] ?

[2] DAS2V : Design Artifact Subject to Verification and Validation, see [**?**]

	Author	Assessor 1	Assessor 2	Total
1c SSRS Verification	2			
1c SSRS Validation	3			
2c SFM Verification	2			
2c SFM Validation	3			
3d SW-SFM Verification	3			
3d SW-SFM Validation	3			
3d SW-FFM Verification	3			
3d SW-FFM Validation	3			
3e Code Verification	0			
3e Code Validation	0			
DAS2V Verification	3			
DAS2V Validation	3			
Automatic model transformation verification	0			
Automatic code generation verification	0			

B.5 Properties

Which kind of properties or elements are verified or validated by the mean or tool (see also [**?**] section 4) ?

	Author	Assessor 1	Assessor 2	Total
Functionalities of the system and sub-system	3			
System and sub-system architecture	3			
External and internal interfaces of sub-system	3			
Software components	3			
Performance constraints	3			
Safety objectives	1			
Functional properties	3			
Safety properties	1			

B.6 Verification methods and tools

Which kind of methods is proposed (see also [**?**] section 5.3) ?

	Author	Assessor 1	Assessor 2	Total
Reviews	0			
Inspections	0			
Software Architecture Analysis Method	2			
Architecture Tradeoff Analysis Method	2			
Model-Based System Integration Testing	1			
Model-Based Testing of Generated High-Level Code	1			
Abstract Interpretation	0			
Deductive Verification	0			
Model Checking	1			
Correct by Construction Formal Methods	0			
Verification with Formal Methods	0			
Simulation-based	3			

B.7 Validation means and tools

The following list of criteria focuss on means and tools to support validation activities, according WP2 requirements :

	Author	Assessor 1	Assessor 2	Total
Simulation-based	3			
Step-by-step simulation (D2.6-01-036)	3			
Environment emulation (D2.6-01-037 and D2.6-02-080)	2			
Time-based test case (D2.6-02-081)	3			
Test cases writing (D2.6-01-038)	1			
Test cases execution (D2.6-01-038)	3			
Test cases storage (D2.6-01-038)	1			
Version management of test cases (D2.6-02-082)	3			
Test generation from independant test model (D2.6-02-083)	1			
Test sequences writing (D2.6-02-084)	3			
Test sequences execution (D2.6-02-084)	3			
Test sequences storage (D2.6-02-084)	3			

B.8 VnV artifacts

Concerning the artifacts used or produced by the mean or tool, please to detail:

Input

Which is the list of the input artifacts for the mean or tools ?

Author:

- *due to the abilities of an universal programming language, many different types of inputs are feasible and could be implemented, e.g. XML structures, transformed SysML models, ...*

Output

Which is the list of the output artifacts for the mean or tools ?

Author:

- *due to the abilities of an universal programming language, many different types of outputs are feasible and could be implemented, e.g. XML structures, source code, ...*
- *already included part of SystemC: value dump files of variable and signals after SystemC simulation*

Syntax

Which are the reference documents which give a description of the artifacts syntax ?

Author:

- `http://standards.ieee.org/findstds/standard/1666-2011.html`

Semantic

Which are the reference documents which give a description of the artifacts semantic ?

Author:

- `http://standards.ieee.org/findstds/standard/1666-2011.html`

Integration

How these artifacts can be integrated with the elements of the toolchain (language, mangement,...) ?

B.9 Detailled Criterias for VnV

Please fill only the section concerning the proposed mean or tool, other section can be skipped (see issue `https://github.com/openETCS/toolchain/issues/180` for details and discussions)

B.9.1 System Modelling simulation

	Author	Assessor 1	Assessor 2	Total
User Scenario Modelling	3			
Test Case Modelling	3			
Test Sequence Modelling	3			

B.9.2 System Model Verification

	Author	Assessor 1	Assessor 2	Total
Input/ Output checking				
System Behavior Simulation (Mathematical)	3			
System Behavior Simulation (Animated)	3			

B.9.3 Software Model Verification

	Author	Assessor 1	Assessor 2	Total
Static Model Verification				
Property Proofing				
Dynamic Testing				
Automatic Test Generation				
Input/ Output checking				
Software Behaviour Simulation (Mathematical)	3			
Software Behaviour Simulation (Animated)	3			

B.9.4 Source Code

	Author	Assessor 1	Assessor 2	Total
Traceability to Model				

B.9.5 Code Verification

	Author	Assessor 1	Assessor 2	Total
Formal Proof				
Programming by contract				
Static Analysis				
Dynamic Analysis				
Dynamic Testing				
Automatic Test Generation				
Performance Testing				
Interface Testing				

B.9.6 Validation System/Software/Code/ Validation

	Author	Assessor 1	Assessor 2	Total
Test Coverage	1			
Use Case Validation of Model	1			
Functional or Black-box Testing	3			
User Scenario Testing	3			
Traceability	1			
Schedulability Analyzer / UseCase Check all	1			
Schedulability Analyzer / UseCase Check single mode	1			

B.10 Other comments

Comment. This section is available for the author or the assessors to complete the description and criteria.

Appendix C: UPPAAL

C.1 Instructions

Author Stefan Rieger (TWT)

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 at any place.

Assessor 1: First assessor can add comments using this format at any place.

Assessor 2: Second assessor can add comments using this format at any place.

When a note is required, please follow this list (inspired from Technology Readiness Level, see `http://en.wikipedia.org/wiki/Technology_readiness_level`) :

0 not recommended / rejected / no integration possible or valuable / not adapted for this topic / not available for this topic

1 weakly recommended / adapted after major improvements / weakly rejected / concept of integration roughly defined / adapted after major improvements / available after major developments

2 recommended / adapted (with light improvements if necessary) weakly accepted / integration prototyped or defined in details / adapted after small improvements / available after small developments or tests

3 highly recommended / well adapted / strongly accepted / integration done and tested / well adapted to the purpose / available and suitable for the purpose All the notes can be commented under each table.

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

All the notes can be commented under each table.

This section defines the criteria for the means and tools dedicated to verification and validation activities, in the WP4 workpackage.

Criteria of this section are defined according [**?**].

C.2 Presentation

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

Name UPPAAL

Web site www.uppaal.org

Licence Academic free or commercial license

Abstract

Uppaal is an integrated tool environment for modeling, validation and verification of real-time systems modeled as networks of timed automata, extended with data types (bounded integers, arrays, etc.).

Publications

Short list of publications on the approach (5 max) Please refer to `http://dblp.org/search/#query=uppaal`

C.3 Common criteria on secondary means and tools

This section discusses the common criteria of the means and tools according to the project requirements on tools and the results of T7.1.

C.3.1 Project and WP2 requirements

The objectives of this list of criteria is to check if the proposed means and tools meet the main criteria of the project: open-source approaches, usability, modularity, coverage of the objectives,...

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 (D2.6-02-074)	0			
Portability to operating systems (D2.6-02-075)	3			
Cooperation of tools (D2.6-02-076)	2			
Robustness (D2.6-02-078)	2*			
Modularity (D2.6-02-078.1)	2			
Documentation management (D2.6-02-078.02)	0**			
Distributed software development (D2.6-02-078.03)	0**			
Simultaneous multi-users (D2.6-02-078.04)	0**			
Issue tracking (D2.6-02-078.05)	0**			
Differences between models (D2.6-02-078.06)	0**			
Version management (D2.6-02-078.07)	0**			
Concurrent version development (D2.6-02-078.08)	0**			
Model-based version control (D2.6-02-078.09)	0**			
Role traceability (D2.6-02-078.10)	0**			
Safety version traceability (D2.6-02-078.11)	0**			
Model traceability (D2.6-02-079)	0**			
Tool chain integration	2			
Scalability	***			
User Friendliness	3			

Author:

* *Sub-criteria of robustness in D2.6 do not make sense here, e.g., version management is not a sub-criterion to robustness.*

** *Out of scope of this tool. The requirements address an tool chain, so other tools should be used to cover these aspects.*

*** *Scalability is difficult to judge and has not been evaluated. As with most tools for model checking it is important how a system model is specified (e.g., bounded datatypes, etc.).*

C.3.2 Qualification

This section discusses how the tool can be classified according EN50128 requirements (D2.6-02-085). Some qualification shall be mandatory if the tool is involved to design a SIL4 software.

	Author	Assessor 1	Assessor 2	Total
Tool manual (D.2.6-01-42.02)	2*			
Proof of correctness (D.2.6-01-42.03)	0			
Existing industrial usage	**			
Model verification	3			
Test generation	2			
Simulation, execution, debugging	2			
Formal proof	3			

> Author: *The above table is not entirely clear to me. I filled the items 4-7 according to applicability of the tool.*
>
> * *Several tutorial papers available.*
>
> ** *Not checked in this context.*

Which level of tool qualification has been reached or will be reached within the next year ?

> Author: *The possible answers below are not aligned with the above question and thus make no sense. The tool is not / will not be pre-qualified by the tool author. Tool qualification by a third party is not possible because the tool is closed source at the moment.*

Score :

3 already qualified for this level

2 qualification possible to this level, but some elements shall be provided

0 qualification not recommended for this level

	Author	Assessor 1	Assessor 2	Total
class T1				
class T2				
class T3				

Other elements for tool certification

C.3.3 Complementarity with primary toolchain

The objectives of this list of criteria is to check if the proposed means and tools can be easily integrated to the primary toolchain.

C.3.3.1 Language

According to the decisions and the propositions of T7.1, how the mean and approach can be adapted to or can complete the chosen language and methods:

	Author	Assessor 1	Assessor 2	Total
SysML	2*			
Scade method	**			
EFS language	**			
B Method	**			
C language	0			

Author:

* *Due to XML input and output formats it can be adapted to be combined with SysML, e.g., generating timed automata from SysML Statecharts (Acceleo seems suitable for that)*

** *I am lacking the information to judge these items regarding direct integration. Please also read my answers below.*

SysML

How the means or tools can complete SysML ?

Author:

- *Transformation from SysML, e.g., by using Acceleo*
- *Use UPPAAL to simulate and verify timed aspects of the primary model*
- *Timed model checking using a (manual) abstraction of the SysML-Model*
- *Test cases from counter examples*

Scade, EFS, Classical B

How the means or tools can complete the current proposals for formal modeling language ?

Author:

- *Use UPPAAL to simulate and verify timed aspects of the primary model*
- *Timed model checking using a (manual) abstraction of the primary Model*
- *Test cases from counter examples*

C language

How the means or tools can complete or be adapted to SIL4 software in C language ?

Author: This is not the goal in proposing this tool.

C.3.3.2 Tools and platforms

According to the decisions and the propositions of T7.1, how the mean and approach can be integrated to or can complete the chosen tools and platforms:

Author: This section in my opinion is redundant for UPPAAL, see my answers above (the answers for Eclipse and Papyrus are the same as for SysML).

	Author	Assessor 1	Assessor 2	Total
Eclipse				
Papyrus				
Scade				
EFS tools				
B tools				

Eclipse

How the means or tools can be integrated to the Eclipse platform ?

Author: See comments regarding SysML above.

Papyrus

How the means or tools can complete Papyrus ?

Author: See comments regarding SysML above.

Scade, EFS, Classical B

How the means or tools can complete the current proposals for formal modeling tools ?

Author: See comments regarding cade, EFS, Classical B above.

C.4 VnV Activities

The VnV activities are described in details in the verification and Validation Plan [**?**].

According figure L1, for which activities is the mean or tool suitable (see also [**?**] section 5.1.2 for more details)[3] ?

[3] DAS2V : Design Artifact Subject to Verification and Validation, see [**?**]

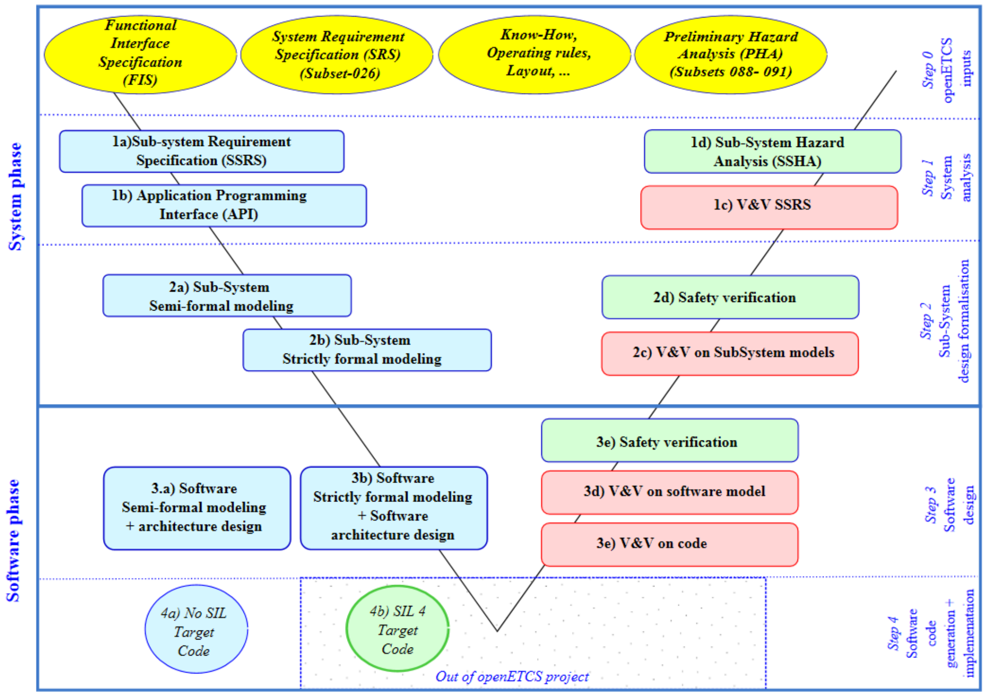

Functional Interface Specification (FIS)
System Requirement Specification (SRS) (Subset-026)
Know-How, Operating rules, Layout, ...
Preliminary Hazard Analysis (PHA) (Subsets 088- 091)
Step 0 openETCS inputs
1a)Sub-system Requirement Specification (SSRS)
1b) Application Programming Interface (API)
1d) Sub-System Hazard Analysis (SSHA)
1c) V&V SSRS
Step 1 System analysis
System phase
2a) Sub-System Semi-formal modeling
2b) Sub-System Strictly formal modeling
2d) Safety verification
2c) V&V on SubSystem models
Step 2 Sub-System design formalisation
Software phase
3.a) Software Semi-formal modeling + architecture design
3b) Software Strictly formal modeling + Software architecture design
3e) Safety verification
3d) V&V on software model
3e) V&V on code
Step 3 Software design
4a) No SIL Target Code
4b) SIL 4 Target Code
Out of openETCS project
Step 4 Software code generation + implemenataion

Figure C1. openETCS Process (rough view)

	Author	Assessor 1	Assessor 2	Total
1c SSRS Verification	3			
1c SSRS Validation	3			
2c SFM Verification	3			
2c SFM Validation	3			
3d SW-SFM Verification	2*			
3d SW-SFM Validation	2*			
3d SW-FFM Verification	2*			
3d SW-FFM Validation	2*			
3e Code Verification	0			
3e Code Validation	0			
DAS2V Verification	3			
DAS2V Validation	3			
Automatic model transformation verification	0			
Automatic code generation verification	0			

Author:

* *Assuming that SW-SFM means Software Semi Formal Model and SW-FFM Software Fully Formal Model*

C.5 Properties

Which kind of properties or elements are verified or validated by the mean or tool (see also [?] section 4) ?

	Author	Assessor 1	Assessor 2	Total
Functionalities of the system and sub-system	3			
System and sub-system architecture	0			
External and internal interfaces of sub-system	0			
Software components	2			
Performance constraints	2			
Safety objectives	3			
Functional properties	3			
Safety properties	3			

C.6 Verification methods and tools

Which kind of methods are proposed (see also [?] section 5.3) ?

	Author	Assessor 1	Assessor 2	Total
Reviews	0			
Inspections	0			
Software Architecture Analysis Method	0			
Architecture Tradeoff Analysis Method	0			
Model-Based System Integration Testing	0			
Model-Based Testing of Generated High-Level Code	0			
Abstract Interpretation	0			
Deductive Verification	0			
Model Checking	3			
Correct by Construction Formal Methods	0			
Verification with Formal Methods	3			
Simulation-based	2			

C.7 Validation means and tools

The following list of criteria focuss on means and tools to support validation activities, according WP2 requirements :

	Author	Assessor 1	Assessor 2	Total
Simulation-based	3			
Step-by-step simulation (D2.6-01-036)	3			
Environment emulation (D2.6-01-037 and D2.6-02-080)	0			
Time-based test case (D2.6-02-081)	0			
Test cases writing (D2.6-01-038)	0			
Test cases execution (D2.6-01-038)	0			
Test cases storage (D2.6-01-038)	0			
Version management of test cases (D2.6-02-082)	0			
Test generation from independant test model (D2.6-02-083)	2			
Test sequences writing (D2.6-02-084)	0			
Test sequences execution (D2.6-02-084)	0			
Test sequences storage (D2.6-02-084)	0			

C.8 VnV artifacts

Concerning the artifacts used or produced by the mean or tool, please to detail:

Input

Which is the list of the input artifacts for the mean or tools ?

Author: Network of timed automata in XML format, this could be a transformed SysML diagram (e.g., statechart).

Output

Which is the list of the output artifacts for the mean or tools ?

Author: UPPAAL is an analysis tool that does not provide a single output. Possible results may include:

- *Identification of timing issues in the system design or the specification*
- *Specification findings due to simulation/validation of the ETCS specificaiton*
- *Results from model verification with possible error traces / bad states*

Syntax

Which are the reference documents which give a description of the artifacts syntax ? `http://www.it.uu.se/resear`

Semantic

Which are the reference documents which give a description of the artifacts semantic ? `http://www.it.uu.se/resea`

Integration

How these artifacts can be integrated with the elements of the toolchain (language, mangement,...) ?

Author: See above.

C.9 Detailled Criterias for VnV

Please fill only the section concerning the proposed mean or tool, other section can be skipped (see issue `https://github.com/openETCS/toolchain/issues/180` for details and discussions)

C.9.1 System Modelling simulation

	Author	Assessor 1	Assessor 2	Total
User Scenario Modelling	2			
Test Case Modelling	0			
Test Sequence Modelling	0			

C.9.2 System Model Verification

	Author	Assessor 1	Assessor 2	Total
Input/ Output checking	0			
System Behavior Simulation (Mathematical)	3			
System Behavior Simulation (Animated)	3			

C.9.3 Software Model Verification

	Author	Assessor 1	Assessor 2	Total
Static Model Verification	3			
Property Proofing	3			
Dynamic Testing	0			
Automatic Test Generation	0			
Input/ Output checking	0			
Software Behaviour Simulation (Mathematical)	2			
Software Behaviour Simulation (Animated)	2			

C.9.4 Source Code

	Author	Assessor 1	Assessor 2	Total
Traceability to Model				

C.9.5 Code Verification

	Author	Assessor 1	Assessor 2	Total
Formal Proof				
Programming by contract				
Static Analysis				
Dynamic Analysis				
Dynamic Testing				
Automatic Test Generation				
Performance Testing				
Interface Testing				

C.9.6 Validation System/Software/Code/ Validation

	Author	Assessor 1	Assessor 2	Total
Test Coverage				
Use Case Validation of Model				
Functional or Black-box Testing				
User Scenario Testing				
Traceability				
Schedulability Analyzer / UseCase Check all				
Schedulability Analyzer / UseCase Check single mode				

C.10 Other comments

Comment. This section is available for the author or the assessors to complete the description and criteria.

Appendix D: Rodin

No results of evaluation. Slides available on github https://github.com/openETCS/model-evaluation/blob/master/Telco_Secondary_slides/Systerel_Event-B.pdf.

Appendix E: Tools for classical B

No results of evaluation.

Slides available on github `https://github.com/openETCS/model-evaluation/blob/master/Telco_Secondary_slides/ClassicalB_VnV.pdf`.

Appendix F: CPN Tools

Slides available on github `https://github.com/openETCS/model-evaluation/blob/master/Telco_Secondary_slides/b-Introduction_CPNTools.pdf`.

F.1 Instructions

Author Stefan Rieger (TWT), Jan Welte (TUBS)

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 at any place.

Assessor 1: First assessor can add comments using this format at any place.

Assessor 2: Second assessor can add comments using this format at any place.

When a note is required, please follow this list (inspired from Technology Readiness Level, see `http://en.wikipedia.org/wiki/Technology_readiness_level`) :

0 not recommended / rejected / no integration possible or valuable / not adapted for this topic / not available for this topic

1 weakly recommended / adapted after major improvements / weakly rejected / concept of integration roughly defined / adapted after major improvements / available after major developments

2 recommended / adapted (with light improvements if necessary) weakly accepted / integration prototyped or defined in details / adapted after small improvements / available after small developments or tests

3 highly recommended / well adapted / strongly accepted / integration done and tested / well adapted to the purpose / available and suitable for the purpose All the notes can be commented under each table.

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

All the notes can be commented under each table.

This section defines the criteria for the means and tools dedicated to verification and validation activities, in the WP4 workpackage.

Criteria of this section are defined according [**?**].

F.2 Presentation

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

Name CPN Tools

Website http://cpntools.org/

Licence Open Source (GPL/LGPL)

Abstract

CPN Tools is a tool for editing, simulating, and analyzing Colored Petri nets.

The tool features incremental syntax checking and code generation, which take place while a net is being constructed. A fast simulator efficiently handles untimed and timed nets. Full and partial state spaces can be generated and analyzed, and a standard state space report contains information, such as boundedness properties and liveness properties.

Publications

Please refer to http://cpntools.org/publications

F.3 Common criteria on secondary means and tools

This section discusses the common criteria of the means and tools according to the project requirements on tools and the results of T7.1.

F.3.1 Project and WP2 requirements

The objectives of this list of criteria is to check if the proposed means and tools meet the main criteria of the project: open-source approaches, usability, modularity, coverage of the objectives,...

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 (D2.6-02-074)	3			
Portability to operating systems (D2.6-02-075)	2			
Cooperation of tools (D2.6-02-076)	2			
Robustness (D2.6-02-078)	2*			
Modularity (D2.6-02-078.1)	3			
Documentation management (D2.6-02-078.02)	0**			
Distributed software development (D2.6-02-078.03)	0**			
Simultaneous multi-users (D2.6-02-078.04)	0**			
Issue tracking (D2.6-02-078.05)	0**			
Differences between models (D2.6-02-078.06)	0**			
Version management (D2.6-02-078.07)	0**			
Concurrent version development (D2.6-02-078.08)	0**			
Model-based version control (D2.6-02-078.09)	0**			
Role traceability (D2.6-02-078.10)	0**			
Safety version traceability (D2.6-02-078.11)	0**			
Model traceability (D2.6-02-079)	0**			
Tool chain integration	2			
Scalability	2***			
User Friendliness	3			

Author:

* *Sub-criteria of robustness in D2.6 do not make sense here, e.g., version management is not a sub-criterion to robustness.*

** *Out of scope of this tool. The requirements address an tool chain, so other tools should be used to cover these aspects.*

*** *For simulation it seems to scale well. State space generation/exhaustive verification scalability was not evaluated so far.*

F.3.2 Qualification

This section discusses how the tool can be classified according EN50128 requirements (D2.6-02-085). Some qualification shall be mandatory if the tool is involved to design a SIL4 software.

	Author	Assessor 1	Assessor 2	Total
Tool manual (D.2.6-01-42.02)	3			
Proof of correctness (D.2.6-01-42.03)	0			
Existing industrial usage	*			
Model verification	3			
Test generation	2			
Simulation, execution, debugging	3			
Formal proof	3			

> *Author: The above table is not entirely clear to me. I filled the items 4-7 according to applicability of the tool.*

Which level of tool qualification has been reached or will be reached within the next year ?

> *Author: The possible answers below are not aligned with the above question and thus make no sense. This is an open source tool that is not / will not be pre-qualified by the tool author (as is, e.g., gcc). As the tool is open source a qualification should be possible but may involve considerable effort.*

Score :

3 already qualified for this level

2 qualification possible to this level, but some elements shall be provided

0 qualification not recommended for this level

	Author	Assessor 1	Assessor 2	Total
class T1				
class T2				
class T3				

Other elements for tool certification

F.3.3 Complementarity with primary toolchain

The objectives of this list of criteria is to check if the proposed means and tools can be easily integrated to the primary toolchain.

F.3.3.1 Language

According to the decisions and the propositions of T7.1, how the mean and approach can be adapted to or can complete the chosen language and methods:

	Author	Assessor 1	Assessor 2	Total
SysML	2*			
Scade method	**			
EFS language	**			
B Method	**			
C language	1*			

Author:

* *Due to XML input and output formats it can be adapted to be combined with SysML, e.g., generating CPNs from SysML Statecharts (Acceleo seems suitable for that) or C-Code from CPN models (we do not plan this for the project; it is not necessary in the context of the project).*

** *I am lacking the information to judge these items regarding direct integration. Please also read my answers below.*

SysML

How the means or tools can complete SysML ?

Author:

- *Transformation from SysML, e.g., by using Acceleo*
- *Use CPN model to simulate and debug SysML models*
- *Model checking using an abstraction of the SysML-Model (behavioural parts)*
- *Independent test model to validate primary SysML model*
- *Visualisation of system behaviour*

Scade, EFS, Classical B

How the means or tools can complete the current proposals for formal modeling language ?

Author:

- *Model checking using an abstraction of the primary models (behavioural parts)*
- *Independent test model to validate primary model*
- *Visualisation of system behaviour*

C language

How the means or tools can complete or be adapted to SIL4 software in C language ?

Author: This is not the goal in proposing this tool.

F.3.3.2 Tools and platforms

According to the decisions and the propositions of T7.1, how the mean and approach can be integrated to or can complete the chosen tools and platforms:

Author: This section in my opinion is redundant for CPN Tools, see my answers above (the answers for Eclipse and Papyrus are the same as for SysML).

	Author	Assessor 1	Assessor 2	Total
Eclipse				
Papyrus				
Scade				
EFS tools				
B tools				

Eclipse

How the means or tools can be integrated to the Eclipse platform ?

Author: See comments regarding SysML above.

Papyrus

How the means or tools can complete Papyrus ?

Author: See comments regarding SysML above.

Scade, EFS, Classical B

How the means or tools can complete the current proposals for formal modeling tools ?

Author: See comments regarding cade, EFS, Classical B above.

F.4 VnV Activities

The VnV activities are described in details in the verification and Validation Plan [**?**].

According figure L1, for which activities is the mean or tool suitable (see also [**?**] section 5.1.2 for more details)[4] ?

[4] DAS2V : Design Artifact Subject to Verification and Validation, see [**?**]

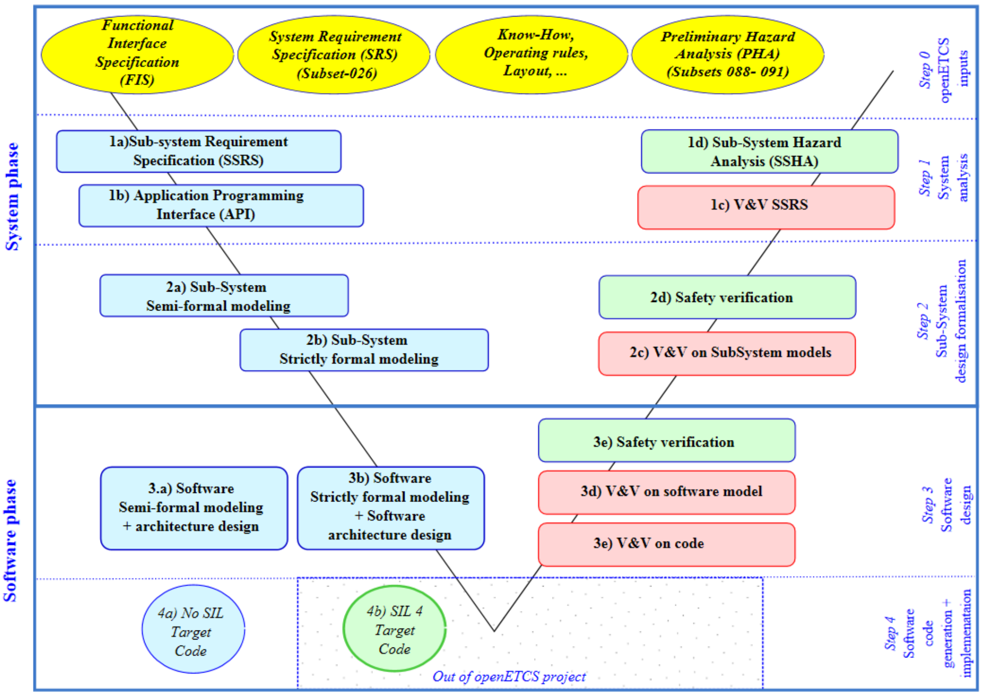

Functional Interface Specification (FIS)
System Requirement Specification (SRS) (Subset-026)
Know-How, Operating rules, Layout, ...
Preliminary Hazard Analysis (PHA) (Subsets 088- 091)
Step 0 openETCS inputs
System phase
1a)Sub-system Requirement Specification (SSRS)
1b) Application Programming Interface (API)
1d) Sub-System Hazard Analysis (SSHA)
1c) V&V SSRS
Step 1 System analysis
2a) Sub-System Semi-formal modeling
2b) Sub-System Strictly formal modeling
2d) Safety verification
2c) V&V on SubSystem models
Step 2 Sub-System design formalisation
Software phase
3.a) Software Semi-formal modeling + architecture design
3b) Software Strictly formal modeling + Software architecture design
3e) Safety verification
3d) V&V on software model
3e) V&V on code
Step 3 Software design
4a) No SIL Target Code
4b) SIL 4 Target Code
Out of openETCS project
Step 4 Software code generation + implemenataion

Figure F1. openETCS Process (rough view)

	Author	Assessor 1	Assessor 2	Total
1c SSRS Verification	3			
1c SSRS Validation	3			
2c SFM Verification	3			
2c SFM Validation	3			
3d SW-SFM Verification	3*			
3d SW-SFM Validation	3*			
3d SW-FFM Verification	3*			
3d SW-FFM Validation	3*			
3e Code Verification	0			
3e Code Validation	0			
DAS2V Verification	3			
DAS2V Validation	3			
Automatic model transformation verification	0			
Automatic code generation verification	0			

Author:

- * *Assuming that SW-SFM means Software Semi Formal Model and SW-FFM Software Fully Formal Model*

F.5 Properties

Which kind of properties or elements are verified or validated by the mean or tool (see also [?] section 4) ?

	Author	Assessor 1	Assessor 2	Total
Functionalities of the system and sub-system	3			
System and sub-system architecture	0			
External and internal interfaces of sub-system	0			
Software components	3			
Performance constraints	2*			
Safety objectives	3			
Functional properties	3			
Safety properties	3			

*Author: * By introducing timing*

F.6 Verification methods and tools

Which kind of methods are proposed (see also [?] section 5.3) ?

	Author	Assessor 1	Assessor 2	Total
Reviews	0			
Inspections	0			
Software Architecture Analysis Method	0			
Architecture Tradeoff Analysis Method	0			
Model-Based System Integration Testing	0			
Model-Based Testing of Generated High-Level Code	0			
Abstract Interpretation	0			
Deductive Verification	0			
Model Checking	3			
Correct by Construction Formal Methods	0			
Verification with Formal Methods	3			
Simulation-based	3			

F.7 Validation means and tools

The following list of criteria focus on means and tools to support validation activities, according to WP2 requirements :

	Author	Assessor 1	Assessor 2	Total
Simulation-based	3			
Step-by-step simulation (D2.6-01-036)	3			
Environment emulation (D2.6-01-037 and D2.6-02-080)	0			
Time-based test case (D2.6-02-081)	2			
Test cases writing (D2.6-01-038)	0			
Test cases execution (D2.6-01-038)	0			
Test cases storage (D2.6-01-038)	0			
Version management of test cases (D2.6-02-082)	0			
Test generation from independant test model (D2.6-02-083)	2			
Test sequences writing (D2.6-02-084)	0			
Test sequences execution (D2.6-02-084)	0			
Test sequences storage (D2.6-02-084)	0			

F.8 VnV artifacts

Concerning the artifacts used or produced by the mean or tool, please to detail:

Input

Which is the list of the input artifacts for the mean or tools ?

Author: CPN in XML format, this could be a transformed SysML diagram (e.g., statechart). CPN Tools is based on the functional language ML and thus the input may contain ML elements.

Output

Which is the list of the output artifacts for the mean or tools ?

Author: CPN Tools is an analysis tool that does not provide a single output. Possible results of a CPN-analysis may include:

- *Specification findings due to simulation/validation of the ETCS specificaiton*
- *Results from model verification with possible error traces / bad states*
- *Visualisation of system execution*

Syntax

Which are the reference documents which give a description of the artifacts syntax ?

Author: See http://cpntools.org/documentation/start

Semantic

Which are the reference documents which give a description of the artifacts semantic ?

Author: See http://cpntools.org/documentation/start

Integration

How these artifacts can be integrated with the elements of the toolchain (language, mangement,...) ?

Author: See above.

F.9 Detailled Criteria for VnV

Please fill only the section concerning the proposed mean or tool, other section can be skipped (see issue `https://github.com/openETCS/toolchain/issues/180` for details and discussions)

F.9.1 System Modelling simulation

	Author	Assessor 1	Assessor 2	Total
User Scenario Modelling	3			
Test Case Modelling	3			
Test Sequence Modelling	0			

F.9.2 System Model Verification

	Author	Assessor 1	Assessor 2	Total
Input/ Output checking	3			
System Behavior Simulation (Mathematical)	3			
System Behavior Simulation (Animated)	3			

F.9.3 Software Model Verification

	Author	Assessor 1	Assessor 2	Total
Static Model Verification	3			
Property Proofing	3			
Dynamic Testing	0			
Automatic Test Generation	0			
Input/ Output checking	0			
Software Behaviour Simulation (Mathematical)	3			
Software Behaviour Simulation (Animated)	3			

F.9.4 Source Code

	Author	Assessor 1	Assessor 2	Total
Traceability to Model				

F.9.5 Code Verification

	Author	Assessor 1	Assessor 2	Total
Formal Proof				
Programming by contract				
Static Analysis				
Dynamic Analysis				
Dynamic Testing				
Automatic Test Generation				
Performance Testing				
Interface Testing				

F.9.6 Validation System/Software/Code/ Validation

	Author	Assessor 1	Assessor 2	Total
Test Coverage				
Use Case Validation of Model				
Functional or Black-box Testing				
User Scenario Testing				
Traceability				
Schedulability Analyzer / UseCase Check all				
Schedulability Analyzer / UseCase Check single mode				

F.10 Other comments

Comment. This section is available for the author or the assessors to complete the description and criteria.

Appendix G: Matelo

No results of evaluation.

Appendix H: RT-Tester

No results of evaluation.

Appendix I: Fiacre and Tina

No results of evaluation.

Appendix J: Frama-C

J.1 Presentation

Author Virgile Prevosto - CEA LIST

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

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

Name Frama-C

Web site `http://frama-c.com`

Licence LGPL 2.1

Abstract

Frama-C is a framework dedicated to the analysis of C programs. It comes with a formal specification language, ACSL, that allows to describe the contracts that each function is supposed to fulfill It features a number of plug-ins that perform various verification tasks. In particular, Value analysis is an abstract interpretation-based plugin that can verify the absence of run-time error for any execution of the program, and WP is an Hoare-logic based plug-in that can modularly check that an implementation is conforming to its ACSL specification with the help of automated theorem provers. Frama-C is also meant to be extensible, and it is easy to tailor the generic plugins toward specific VnV tasks.

Publications

- Pascal Cuoq, Florent Kirchner, Nikolai Kosmatov, Virgile Prevosto, Julien Signoles and Boris Yakobowski. Frama-C: a Software Analysis Perspective. Proceedings of SEFM 2012.
- Loïc Correnson and Julien Signoles. Combining Analysises for C Program Verification. Proceedings of FMICS 2012
- Jochen Burghardt, Jens Gerlach, Kerstin Hartig, Hans Pohl and Juan Soto. ACSL by Example, a fairly complete tour of ACSL features through various functions inspired from C++ STL.
- Patrick Baudin, Loïc Correnson and Zaynah Dargaye, WP plugin manual. `http://frama-c.com/download/wp-manual-Fluorine-20130601.pdf`
- Pascal Cuoq, Boris Yakobowski and Virgile Prevosto. Frama-C's value analysis plugin manual. `http://frama-c.com/download/frama-c-value-analysis.pdf`

J.2 Common criteria on secondary means and tools

This section discusses the common criteria of the means and tools according to the project requirements on tools and the results of T7.1.

J.2.1 Project and WP2 requirements

The objectives of this list of criteria is to check if the proposed means and tools meet the main criteria of the project: open-source approaches, usability, modularity, coverage of the objectives,...

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 (D2.6-02-074)	3			
Portability to operating systems (D2.6-02-075)	2			
Cooperation of tools (D2.6-02-076)	1			
Robustness (D2.6-02-078)	2			
Modularity (D2.6-02-078.1)	2			
Documentation management (D2.6-02-078.02)	0			
Distributed software development (D2.6-02-078.03)	0			
Simultaneous multi-users (D2.6-02-078.04)	*			
Issue tracking (D2.6-02-078.05)	0			
Differences between models (D2.6-02-078.06)	0			
Version management (D2.6-02-078.07)	0			
Concurrent version development (D2.6-02-078.08)	0			
Model-based version control (D2.6-02-078.09)	0			
Role traceability (D2.6-02-078.10)	0			
Safety version traceability (D2.6-02-078.11)	0			
Model traceability (D2.6-02-079)	0			
Tool chain integration	1			
Scalability	2			
User Friendliness	1			

Author:

- *Frama-C works on Linux, Windows and MacOS X, but must be compiled from source, which is not always very easy on Windows.*
- *Cooperation can mainly be envisaged in the form of generation of ACSL specification from SysML/Scade, but there is no existing plug-in for that.*
- *An Eclipse plug-in is available (FCDT: `http://gforge.enseeiht.fr/projects/fcdt/`) for launching some analyses from Eclipse.*

J.2.2 Qualification

This section discusses how the tool can be classified according EN50128 requirements (D2.6-02-085). Some qualification shall be mandatory if the tool is involved to design a SIL4 software.

	Author	Assessor 1	Assessor 2	Total
Tool manual (D.2.6-01-42.02)	2			
Proof of correctness (D.2.6-01-42.03)	2			
Existing industrial usage	1			
Model verification	0			
Test generation	*			
Simulation, execution, debugging	2			
Formal proof	3			

> *Author: There exists a test-case generator plugin, but it is not distributed under an Open-Source licence. Value Analysis plugin can be tweaked into an interpreter/simulator if needed*

Which level of tool qualification has been reached or will be reached within the next year ?

Score :

3 already qualified for this level

2 qualification possible to this level, but some elements shall be provided

0 qualification not recommended for this level

	Author	Assessor 1	Assessor 2	Total
class T1	1			
class T2	1			
class T3	1			

> *Author: There has been some reflection on the qualification of Frama-C with respect to DO-178 verification activities, but no formal process has taken place yet.*

Other elements for tool certification

J.2.3 Complementarity with primary toolchain

The objectives of this list of criteria is to check if the proposed means and tools can be easily integrated to the primary toolchain.

J.2.3.1 Language

According to the decisions and the propositions of T7.1, how the mean and approach can be adapted to or can complete the chosen language and methods:

	Author	Assessor 1	Assessor 2	Total
SysML	1			
Scade method	2			
EFS language	1			
B Method	2			
C language	3			

SysML

It should be possible to translate a subset of SysML specifications into ACSL contracts, in order to verify that an implementation is correct with respect to such contracts.

Scade, EFS, Classical B

Similarly, Scade and B models could in principle be translated into ACSL specifications.

C language

Frama-C takes C programs as its primary input (possibly together with ACSL specifications).

J.2.3.2 Tools and platforms

According to the decisions and the propositions of T7.1, how the mean and approach can be integrated to or can complete the chosen tools and platforms:

	Author	Assessor 1	Assessor 2	Total
Eclipse	2			
Papyrus	1			
Scade	1			
EFS tools	1			
B tools	1			

Eclipse

There exists an Eclipse plugin (FCDT, see above) to interact with the Value analysis plugin directly from the Eclipse platform.

Papyrus

See previous section. It could be envisaged to translate SysML to ACSL specifications from Papyrus.

Scade, EFS, Classical B

See previous section

J.3 VnV Activities

The VnV activities are described in details in the verification and Validation Plan [**?**].

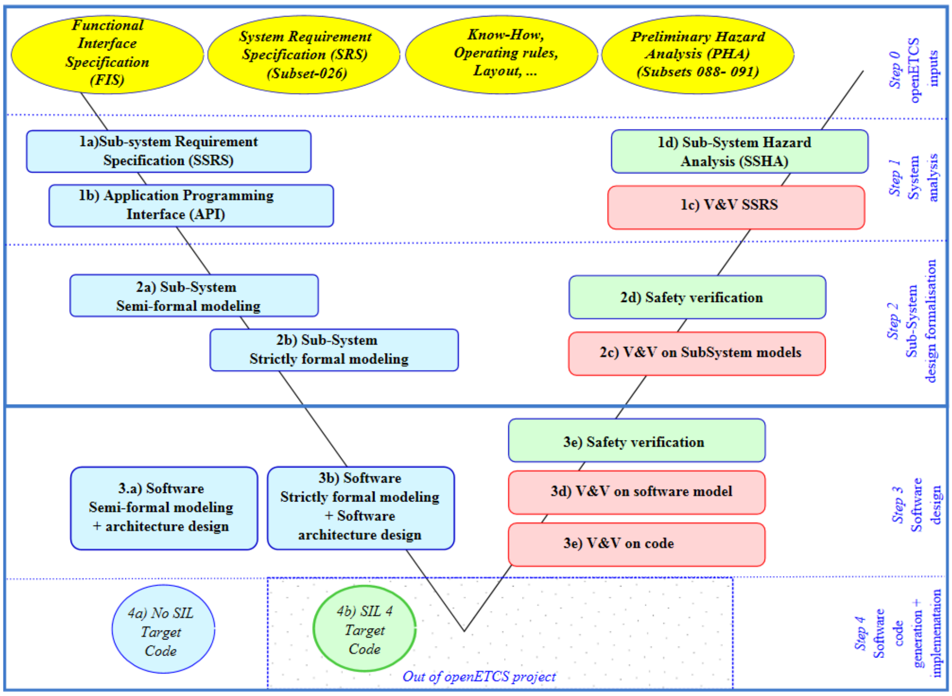

Functional Interface Specification (FIS)
System Requirement Specification (SRS) (Subset-026)
Know-How, Operating rules, Layout, ...
Preliminary Hazard Analysis (PHA) (Subsets 088- 091)
Step 0 openETCS inputs
System phase
1a)Sub-system Requirement Specification (SSRS)
1b) Application Programming Interface (API)
1d) Sub-System Hazard Analysis (SSHA)
1c) V&V SSRS
Step 1 System analysis
2a) Sub-System Semi-formal modeling
2b) Sub-System Strictly formal modeling
2d) Safety verification
2c) V&V on SubSystem models
Step 2 Sub-System design formalisation
Software phase
3.a) Software Semi-formal modeling + architecture design
3b) Software Strictly formal modeling + Software architecture design
3e) Safety verification
3d) V&V on software model
3e) V&V on code
Step 3 Software design
4a) No SIL Target Code
4b) SIL 4 Target Code
Out of openETCS project
Step 4 Software code generation + implemenataion

Figure J1. openETCS Process (rough view)

According figure J1, for which activities is the mean or tool suitable (see also [**?**] section 5.1.2 for more details)[5] ?

	Author	Assessor 1	Assessor 2	Total
1c SSRS Verification	0			
1c SSRS Validation	0			
2c SFM Verification	0			
2c SFM Validation	0			
3d SW-SFM Verification	0			
3d SW-SFM Validation	0			
3d SW-FFM Verification	2			
3d SW-FFM Validation	2			
3e Code Verification	3			
3e Code Validation	3			
DAS2V Verification	0			
DAS2V Validation	0			
Automatic model transformation verification	0			
Automatic code generation verification	3			

[5] DAS2V : Design Artifact Subject to Verification and Validation, see [**?**]

J.4 Properties

Which kind of properties or elements are verified or validated by the mean or tool (see also [?] section 4) ?

	Author	Assessor 1	Assessor 2	Total
Functionalities of the system and sub-system	1			
System and sub-system architecture	0			
External and internal interfaces of sub-system	0			
Software components	3			
Performance constraints	1			
Safety objectives	0			
Functional properties	3			
Safety properties	0			

J.5 Verification methods and tools

Which kind of methods is proposed (see also [?] section 5.3) ?

	Author	Assessor 1	Assessor 2	Total
Reviews	0			
Inspections	0			
Software Architecture Analysis Method	0			
Architecture Tradeoff Analysis Method	0			
Model-Based System Integration Testing	0			
Model-Based Testing of Generated High-Level Code	0			
Abstract Interpretation	3			
Deductive Verification	3			
Model Checking	1			
Correct by Construction Formal Methods	0			
Verification with Formal Methods	3			
Simulation-based	1			

Author: Value analysis plug-in can be tweaked to perform some model-checking and simulation tasks

J.6 Validation means and tools

The following list of criteria focuss on means and tools to support validation activities, according WP2 requirements :

	Author	Assessor 1	Assessor 2	Total
Simulation-based	1			
Step-by-step simulation (D2.6-01-036)	1			
Environment emulation (D2.6-01-037 and D2.6-02-080)	1			
Time-based test case (D2.6-02-081)	0			
Test cases writing (D2.6-01-038)	1			
Test cases execution (D2.6-01-038)	1			
Test cases storage (D2.6-01-038)	0			
Version management of test cases (D2.6-02-082)	0			
Test generation from independant test model (D2.6-02-083)	0			
Test sequences writing (D2.6-02-084)	0			
Test sequences execution (D2.6-02-084)	0			
Test sequences storage (D2.6-02-084)	0			

J.7 VnV artifacts

Concerning the artifacts used or produced by the mean or tool, please to detail:

Input

- C code under analysis
- for functional properties: ACSL contracts specifying the properties of interest
- set of parameters for the plugins that are used.

Output

- validity status of each ACSL property
- (hopefully empty) list of all potential run-time errors that might occur during an execution.

Syntax

- ISO/IEC JTC1/SC22/WG14. 9899:TC3: Programming Languages—C
- ACSL: ANSI/ISO C Specification Language. `http://frama-c.com/acsl.html`
- Frama-C manuals available at `http://frama-c.com/`

Semantic

See above.

Integration

How these artifacts can be integrated with the elements of the toolchain (language, mangement,...) ?

J.8 Detailled Criterias for VnV

Please fill only the section concerning the proposed mean or tool, other section can be skipped (see issue `https://github.com/openETCS/toolchain/issues/180` for details and discussions)

J.8.1 System Modelling simulation

	Author	Assessor 1	Assessor 2	Total
User Scenario Modelling				
Test Case Modelling				
Test Sequence Modelling				

J.8.2 System Model Verification

	Author	Assessor 1	Assessor 2	Total
Input/ Output checking				
System Behavior Simulation (Mathematical)				
System Behavior Simulation (Animated)				

J.8.3 Software Model Verification

	Author	Assessor 1	Assessor 2	Total
Static Model Verification				
Property Proofing				
Dynamic Testing				
Automatic Test Generation				
Input/ Output checking				
Software Behaviour Simulation (Mathematical)				
Software Behaviour Simulation (Animated)				

J.8.4 Source Code

	Author	Assessor 1	Assessor 2	Total
Traceability to Model	1			

Author: *If ACSL specifications are generated from the model (see section J.2.3.1) with proper traceability artifacts, it should be possible to trace back each verification condition assessed by Frama-C to the original constraint in the model.*

J.8.5 Code Verification

	Author	Assessor 1	Assessor 2	Total
Formal Proof	3			
Programming by contract	3			
Static Analysis	3			
Dynamic Analysis	1			
Dynamic Testing	*			
Automatic Test Generation	*			
Performance Testing	1			
Interface Testing	1			

> *Author: Value analysis' interpreter mode can be used to perform some dynamic analysis. As mentioned above, there is a test case generator plugin, but it is not distributed as free software.*

J.8.6 Validation System/Software/Code/ Validation

	Author	Assessor 1	Assessor 2	Total
Test Coverage				
Use Case Validation of Model				
Functional or Black-box Testing				
User Scenario Testing				
Traceability				
Schedulability Analyzer / UseCase Check all				
Schedulability Analyzer / UseCase Check single mode				

J.9 Other comments

> *Comment. This section is available for the author or the assessors to complete the description and criteria.*

Slides of Frama-C presentation are available on github `https://github.com/openETCS/model-evaluation/blob/master/Telco_Secondary_slides/c-frama-c-secondary-tools-presentati` `pdf`.

Specification and verification activities over the `bitwalker` code from Siemens have been conducted by Fraunhofer FOKUS with support from CEA LIST. The development is available on github: `https://github.com/openETCS/validation/tree/master/VnVUserStories/VnVUserStoryFraunhoferFOKUS`

Appendix K: Diversity

No results of evaluation.

Appendix L: IF toolset

L.1 Instructions

Author Huu Nghia Nguyen (Instiut Télécom-SudParis)

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 at any place.

Assessor 1: First assessor can add comments using this format at any place.

Assessor 2: Second assessor can add comments using this format at any place.

When a note is required, please follow this list (inspired from Technology Readiness Level, see `http://en.wikipedia.org/wiki/Technology_readiness_level`) :

0 not recommended / rejected / no integration possible or valuable / not adapted for this topic / not available for this topic

1 weakly recommended / adapted after major improvements / weakly rejected / concept of integration roughly defined / adapted after major improvements / available after major developments

2 recommended / adapted (with light improvements if necessary) weakly accepted / integration prototyped or defined in details / adapted after small improvements / available after small developments or tests

3 highly recommended / well adapted / strongly accepted / integration done and tested / well adapted to the purpose / available and suitable for the purpose All the notes can be commented under each table.

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

All the notes can be commented under each table.

This section defines the criteria for the means and tools dedicated to verification and validation activities, in the WP4 workpackage.

Criteria of this section are defined according [**?**].

L.2 Presentation

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

Name IF toolset (IF)

Web site `http://www-if.imag.fr`

Licence This software may be freely used, copied and redistributed without fee for non-commerical purpose

Abstract

The IF toolset which is an environment for modelling and validation of heterogeneous real-time systems. The toolset is built upon a rich formalism, the Intermediate Language (IF) notation, allowing structured automata-based system representations. Moreover, the IF notation is expressive enough to support real-time primitives and extensions of high-level modelling languages such as SDL and UML/SysML by means of structure preserving mappings. The core part of the IF toolset consists of a syntactic transformation component and an open exploration platform. The syntactic transformation component provides language level access to IF descriptions and has been used to implement static analysis and optimisation techniques. The exploration platform gives access to the graph of possible executions. It has been connected to different state-of-the-art model-checking and test-case generation tools.

Publications

Please refer to `http://www-if.imag.fr/papers.html`

L.3 Common criteria on secondary means and tools

This section discusses the common criteria of the means and tools according to the project requirements on tools and the results of T7.1.

L.3.1 Project and WP2 requirements

The objectives of this list of criteria is to check if the proposed means and tools meet the main criteria of the project: open-source approaches, usability, modularity, coverage of the objectives,...

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 (D2.6-02-074)	2*			
Portability to operating systems (D2.6-02-075)	3**			
Cooperation of tools (D2.6-02-076)	2			
Robustness (D2.6-02-078)	2			
Modularity (D2.6-02-078.1)	2			
Documentation management (D2.6-02-078.02)	0			
Distributed software development (D2.6-02-078.03)	0			
Simultaneous multi-users (D2.6-02-078.04)	1***			
Issue tracking (D2.6-02-078.05)	0			
Differences between models (D2.6-02-078.06)	0			
Version management (D2.6-02-078.07)	0			
Concurrent version development (D2.6-02-078.08)	0			
Model-based version control (D2.6-02-078.09)	0			
Role traceability (D2.6-02-078.10)	0			
Safety version traceability (D2.6-02-078.11)	0			
Model traceability (D2.6-02-079)	0			
Tool chain integration	2			
Scalability	2			
User Friendliness	2			

Author:

* *The source code can be retrieved from its API document* `http://www-if.imag.fr/apis.html`

** *Currently, it can run on Mac OSX 10.6 and above, Linux, Windows (thanks to Cygwin), and Solaris 2.8*

*** *Multi simulators can connect simultaneously to the tool*

L.3.2 Qualification

This section discusses how the tool can be classified according EN50128 requirements (D2.6-02-085). Some qualification shall be mandatory if the tool is involved to design a SIL4 software.

	Author	Assessor 1	Assessor 2	Total
Tool manual (D.2.6-01-42.02)	3			
Proof of correctness (D.2.6-01-42.03)	2			
Existing industrial usage	1			
Model verification	3			
Test generation	3			
Simulation, execution, debugging	2			
Formal proof	1			

Author:

Which level of tool qualification has been reached or will be reached within the next year ?

Score :

3 already qualified for this level

2 qualification possible to this level, but some elements shall be provided

0 qualification not recommended for this level

	Author	Assessor 1	Assessor 2	Total
class T1	3			
class T2	3			
class T3	0			

Other elements for tool certification

L.3.3 Complementarity with primary toolchain

The objectives of this list of criteria is to check if the proposed means and tools can be easily integrated to the primary toolchain.

L.3.3.1 Language

According to the decisions and the propositions of T7.1, how the mean and approach can be adapted to or can complete the chosen language and methods:

	Author	Assessor 1	Assessor 2	Total
SysML	2			
Scade method	0			
EFS language	0			
B Method	0			
C language	3			

SysML

How the means or tools can complete SysML ?

Author:

- *Transformation from SysML, e.g., by using IFx-OMEGA (`http://www.irit.fr/ifx/`)*

Scade, EFS, Classical B

How the means or tools can complete the current proposals for formal modeling language ?

C language

How the means or tools can complete or be adapted to SIL4 software in C language ?

Author:

- *An IF description is translated into a C program which is then compiled and executed to get outputs*

L.3.3.2 Tools and platforms

According to the decisions and the propositions of T7.1, how the mean and approach can be integrated to or can complete the chosen tools and platforms:

	Author	Assessor 1	Assessor 2	Total
Eclipse	1			
Papyrus	3			
Scade	0			
EFS tools	0			
B tools	0			

Eclipse

How the means or tools can be integrated to the Eclipse platform ?

Author: We can use any text editor in Eclipse to create/modify IF description. Eclipse can compile and run this description by "External Tools" command that will call IF toolset.

Papyrus

How the means or tools can complete Papyrus ?

Author: The output of Papyrus is an input of the tool.

Scade, EFS, Classical B

How the means or tools can complete the current proposals for formal modeling tools ?

L.4 VnV Activities

The VnV activities are described in details in the verification and Validation Plan [**?**].

According figure L1, for which activities is the mean or tool suitable (see also [**?**] section 5.1.2 for more details)[6] ?

[6] DAS2V : Design Artifact Subject to Verification and Validation, see [**?**]

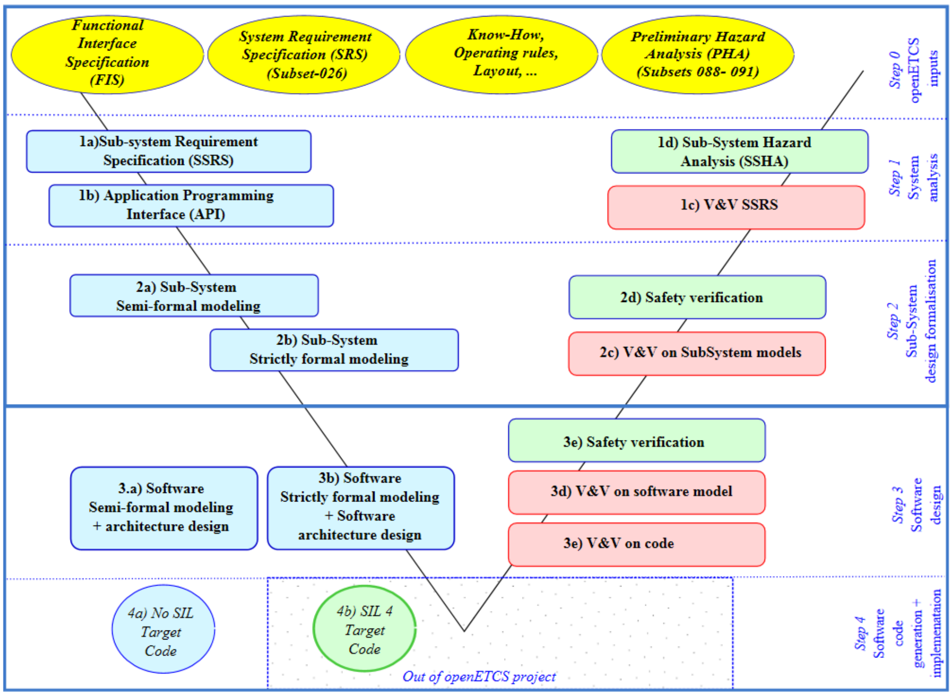

Functional Interface Specification (FIS)
System Requirement Specification (SRS) (Subset-026)
Know-How, Operating rules, Layout, ...
Preliminary Hazard Analysis (PHA) (Subsets 088- 091)
Step 0 openETCS inputs
System phase
1a)Sub-system Requirement Specification (SSRS)
1b) Application Programming Interface (API)
1d) Sub-System Hazard Analysis (SSHA)
1c) V&V SSRS
Step 1 System analysis
2a) Sub-System Semi-formal modeling
2b) Sub-System Strictly formal modeling
2d) Safety verification
2c) V&V on SubSystem models
Step 2 Sub-System design formalisation
Software phase
3.a) Software Semi-formal modeling + architecture design
3b) Software Strictly formal modeling + Software architecture design
3e) Safety verification
3d) V&V on software model
3e) V&V on code
Step 3 Software design
4a) No SIL Target Code
4b) SIL 4 Target Code
Out of openETCS project
Step 4 Software code generation + implemenataion

Figure L1. openETCS Process (rough view)

	Author	Assessor 1	Assessor 2	Total
1c SSRS Verification	3			
1c SSRS Validation	3			
2c SFM Verification	2			
2c SFM Validation	2			
3d SW-SFM Verification	2			
3d SW-SFM Validation	2			
3d SW-FFM Verification	3			
3d SW-FFM Validation	3			
3e Code Verification	0			
3e Code Validation	0			
DAS2V Verification	3			
DAS2V Validation	3			
Automatic model transformation verification	0			
Automatic code generation verification	0			

Author: Assuming that SW-SFM means Software Semi Formal Model and SW-FFM means Software Fully Formal Model

L.5 Properties

Which kind of properties or elements are verified or validated by the mean or tool (see also [**?**] section 4) ?

	Author	Assessor 1	Assessor 2	Total
Functionalities of the system and sub-system	3			
System and sub-system architecture	0			
External and internal interfaces of sub-system	1			
Software components	3			
Performance constraints	2*			
Safety objectives	3			
Functional properties	3			
Safety properties	3			

*Author: * e.g., execution time*

L.6 Verification methods and tools

Which kind of methods is proposed (see also [?] section 5.3) ?

	Author	Assessor 1	Assessor 2	Total
Reviews	0			
Inspections	0			
Software Architecture Analysis Method	0			
Architecture Tradeoff Analysis Method	0			
Model-Based System Integration Testing	0			
Model-Based Testing of Generated High-Level Code	0			
Abstract Interpretation	0			
Deductive Verification	0			
Model Checking	3			
Correct by Construction Formal Methods	0			
Verification with Formal Methods	3			
Simulation-based	3			

L.7 Validation means and tools

The following list of criteria focuses on means and tools to support validation activities, according WP2 requirements :

	Author	Assessor 1	Assessor 2	Total
Simulation-based	3			
Step-by-step simulation (D2.6-01-036)	3			
Environment emulation (D2.6-01-037 and D2.6-02-080)	2			
Time-based test case (D2.6-02-081)	2			
Test cases writing (D2.6-01-038)	1*			
Test cases execution (D2.6-01-038)	1*			
Test cases storage (D2.6-01-038)	1*			
Version management of test cases (D2.6-02-082)	0			
Test generation from independant test model (D2.6-02-083)	2			
Test sequences writing (D2.6-02-084)	1*			
Test sequences execution (D2.6-02-084)	1*			
Test sequences storage (D2.6-02-084)	1*			

Author: ** By encoding them into IF observers*

L.8 VnV artifacts

Concerning the artifacts used or produced by the mean or tool, please to detail:

Input

Which is the list of the input artifacts for the mean or tools ?

Author:

A text file, in IF syntax, containing:

- *models of components of systems in timed automaton extended with data*
- *optionally, observers representing properties to be verified*

The other input can be in format of SDL, UML or SysML which will be translated automatically into IF thanks to existing modules in IF toolset.

Output

Which is the list of the output artifacts for the mean or tools ?

Author: *Depending on usage, the outputs may be:*

- *Results from model verification with error traces / states*
- *Test cases*
- *Visualisation of system execution*

Syntax

Which are the reference documents which give a description of the artifacts syntax ?

Author:

- *http://www-if.imag.fr/tutorials/syntax.ps.gz*
- *http://www-if.imag.fr/tutorials.html*

Semantic

Which are the reference documents which give a description of the artifacts semantic ?

Author:

- *http://www-if.imag.fr/tutorials/semantics.ps.gz*
- *http://www-if.imag.fr/tutorials.html*

Integration

How these artifacts can be integrated with the elements of the toolchain (language, mangement,...) ?

Author: See above.

L.9 Detailled Criterias for VnV

Please fill only the section concerning the proposed mean or tool, other section can be skipped (see issue `https://github.com/openETCS/toolchain/issues/180` for details and discussions)

L.9.1 System Modelling simulation

	Author	Assessor 1	Assessor 2	Total
User Scenario Modelling	3			
Test Case Modelling	2			
Test Sequence Modelling	2			

L.9.2 System Model Verification

	Author	Assessor 1	Assessor 2	Total
Input/ Output checking	3			
System Behavior Simulation (Mathematical)	3			
System Behavior Simulation (Animated)	2			

L.9.3 Software Model Verification

	Author	Assessor 1	Assessor 2	Total
Static Model Verification	3			
Property Proofing	2*			
Dynamic Testing	0			
Automatic Test Generation	3**			
Input/ Output checking	3			
Software Behaviour Simulation (Mathematical)	3			
Software Behaviour Simulation (Animated)	2			

Author:

** by mean of model-checking, i.e., give a counter example or exhaustive verification*

*** with the supports of TestGen-IF or TGV test generator*

L.9.4 Source Code

	Author	Assessor 1	Assessor 2	Total
Traceability to Model				

L.9.5 Code Verification

	Author	Assessor 1	Assessor 2	Total
Formal Proof				
Programming by contract				
Static Analysis				
Dynamic Analysis				
Dynamic Testing				
Automatic Test Generation				
Performance Testing				
Interface Testing				

L.9.6 Validation System/Software/Code/ Validation

	Author	Assessor 1	Assessor 2	Total
Test Coverage				
Use Case Validation of Model				
Functional or Black-box Testing				
User Scenario Testing				
Traceability				
Schedulability Analyzer / UseCase Check all				
Schedulability Analyzer / UseCase Check single mode				

L.10 Other comments

Comment. *This section is available for the author or the assessors to complete the description and criteria.*