

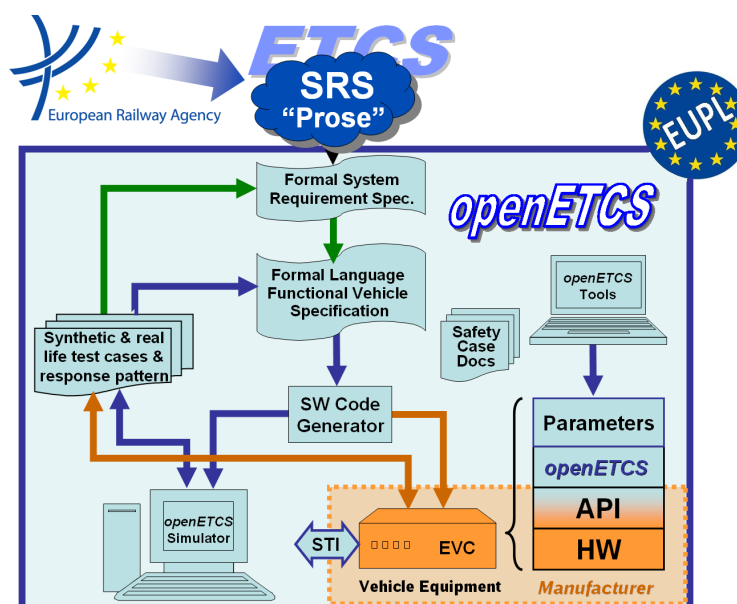
Work-Package 7: “Secondary tools - Management”

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

List of criteria on supporting tools and methods and results on the benchmark

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


République Française

 MINISTÈRE
 DE L'ÉDUCATION NATIONALE
 ET DE LA RECHERCHE

 Région de
 Bruxelles-
 Capitale

 GOBIERNO
 DE ESPAÑA

 MINISTERIO
 DE INDUSTRIA, ENERGÍA
 Y TURISMO

This page is intentionally left blank

Work-Package 7: “Secondary tools - Management”

**OETCS/WP7/O7.2.1 – 00/05
October 2013**

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

List of criteria on supporting tools and methods and results on the benchmark

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, model transformation and data management for the whole project. Evaluation on the means and tools of benchmark is also described.

This document focusses on means and tools to support management of data, functions, requirements and all the artifact during the OpenETCS process.

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

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

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

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

Table of Contents

Figures and Tables.....	v
1 Introduction.....	1
1.1 Organisation of the document	1
2 Template	2
2.1 Instructions	2
2.2 Presentation	2
2.3 Common criteria on secondary means and tools	3
2.3.1 Project and WP2 requirements	3
2.3.2 Qualification	4
2.4 Complementarity with primary toolchain	5
2.4.1 Language	5
2.4.2 Tools and platforms	5
2.5 Means and tools for data, function and requirement management	6
2.5.1 Management activities	6
2.5.2 Input Artifacts	7
2.5.3 Output Artifacts	7
2.5.4 Requirement Management	7
2.6 Other comments	8
3 Conclusion.....	9
Appendix A: ProR	10
A.1 Author and Assessors	10
A.2 Presentation	10
A.3 Common criteria on secondary means and tools	11
A.3.1 Project and WP2 requirements	11
A.3.2 Qualification	13
A.3.3 Complementarity with primary toolchain	14
A.4 Means and tools for data, function and requirement management	16
A.4.1 Management activities	17
A.4.2 Input Artifacts	17
A.4.3 Output Artifacts	18
A.4.4 Requirement Management	18
A.5 Other comments	19
Appendix B: Scade	20
Appendix C: Rodin.....	21
C.1 Author and Assessors	21
C.2 Presentation	21
C.3 Common criteria on secondary means and tools	22
C.3.1 Project and WP2 requirements	22
C.3.2 Qualification	23
C.4 Complementarity with primary toolchain	24
C.4.1 Language	24

C.4.2 Tools and platforms	25
C.5 Means and tools for data, function and requirement management	26
C.5.1 Management activities	26
C.5.2 Input Artifacts	27
C.5.3 Output Artifacts	27
C.5.4 Requirement Management	28
C.6 Other comments	28
Appendix D: Matelo	29
Appendix E: Goal Structuring Notation	30
Appendix F: EMF Store	31
Appendix G: EMF Client Platform	32
Appendix: References	33

Figures and Tables

Figures

Tables

Document information	
Work Package	WP7
Deliverable ID or doc. ref.	O7.2.1
Document title	Evaluation of supporting tools and methods against the WP2 requirements and task 1 - Management
Document version	00.05
Document authors (org.)	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. Management part
00.06	18/10/2013	M. Petit-Doche	Issues: 174, 178, 167, 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 [4], and the results of T7.1 on the primary toolchain [13], 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 to support management of data, functions requirements and other artifacts along the openETCS process.

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
- Rodin and Pluggins
- Matelo
- Goal Structuring Notation (GSN)
- Eclipse ProR
- Eclipse EMF Store
- Eclipse EMF Client Platform

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 :

- 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.

2.2 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)

For which activities are dedicated the means or tools (give a note from 0 to 3) :

	Author	Assessor 1	Assessor 2	Total
Data Management				
Function Management				
Requirement Management				
Version Management				
Other (give details below)				

According the results of this table, some of the following sections can be skipped.

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

2.4 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.

2.4.1 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 ?

2.4.2 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 ?

2.5 Means and tools for data, function and requirement management

This section defines the criteria for the means and tools dedicated to data, function and requirement management. These activities are shared by the work packages WP3, WP4 and the activities dedicated to SSRS. These means and tools shall integrate the primary toolchain to complete its gap and facilitate the integration of different activities. First of all, they allow the management of a common repository of data, functions and requirements, shared between the models (from SSRS informal specification to code) and the verification and validation activities. Then, they shall support traceability of requirements between models and activities, and facilitate the verification of the traceability. Besides they shall support the design of SIL4 software with model comparison or document production facilities, and version management.

2.5.1 Management activities

Which activities, linked to help the management of SSRS definition and whole process are covered by the mean or tool ?

	Author	Assessor 1	Assessor 2	Total
Requirement capturing				
Requirement management				
Data management				
Function management				
Requirement traceability				
Model traceability				
Function architecture				
Version management				
Model comparison				
Documentation production				
Others (give details)				

2.5.2 Input Artifacts

Which artifacts are used as input of the mean or tool ?

	Author	Assessor 1	Assessor 2	Total
Informal description				
Structured description				
Spread sheet				
XML files				
EFS model				
DSL				
Others (give details)				

2.5.3 Output Artifacts

Which artifacts are used as output of the mean or tool ?

	Author	Assessor 1	Assessor 2	Total
Informal description				
Structured description				
Spread sheet				
XML files				
EFS model				
DSL				
Others (give details)				

2.5.4 Requirement Management

This section is link to reauirement definition and management activities.

Are these criteria covered by the tool or mean ?

	Author	Assessor 1	Assessor 2	Total
Editing of Textual Requirements				
Represent Relations between Req (Textual-Based)				
Represent Relations between Req (Modelling-Based)				
Glossary and Abbreviation handling (Linked to Req))				
Traceability of Textual Requirements to Modelling				
Import/Export of Industrial Standard Data (e.g., REQIF)				
Documentation generation				
Search and Filtering functions				
Others (give details)				

2.6 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: ProR

A.1 Author and Assessors

Author Michael Jastram, Formal Mind GmbH

Assessor 1 First assessor of the approaches: Bernd Hekele - DB-Netz

Assessor 2 Second assessor of the approaches: Christophe Ponsard - for Alstom

A.2 Presentation

Name Eclipse ProR

Web site <http://eclipse.org/rmf/pror>

Licence Eclipse Public License

Abstract

ProR is an open source tool, which is part of the Eclipse Requirements Modeling Framework (RMF) [1]. As the underlying data format, ProR uses ReqIF, a standard for exchange of requirements with other tools. It therefore provides interoperability with industry-strength tools like Rational DOORS or MKS Integrity. It supports traceability within requirements, and traceability solutions for artefacts outside ProR exist.

Publications

- [8] contains a broad overview of ProR, as well as an approach to requirements structuring and model traceability.
- [7] looks at ProR as part of an Eclipse-based systems engineering environment.
- [9] suggests the integration of Topcased with ProR, an idea that has been picked up by the Topcased community.
- [11] is the openETCS benchmark, which contains a longer list of literature.

For which activities are dedicated the means or tools (give a note from 0 to 3) :

	Author	Assessor 1	Assessor 2	Total
Data Management	2	2	2	6
Function Management	2	2	2	6
Requirement Management	3	3	3	9
Version Management	2	1	1	4
Other (give details below)	3	1	3	7

Author: ProR clearly is a specialized tool for requirements management. While data and functions could be managed as plain text or numeric data in ProR, this would not be very productive.

Data and Function Management. However, ProR is capable of “pulling information together” from various sources. It would be feasible to show data and functions from Papyrus in-line in a ProR specification. Such an integration plug-in does not currently exist, but could be created with not too much effort.

Version Management. ProR does not contain version management, but works well with any underlying version management tool like Subversion or git, which already works today. However, the granularity of the versioning would be file-based, which is less than perfect. Versioning of individual requirements would be better. Further, higher-level versioning, like baselining, currently does not exist.

Other. Missing from the list is Traceability Management and Model Integration. ProR works very well in these two areas. Specialized traceability to Event-B exists and has been evaluated in [10]. At least two third-party traceability solutions exist: ReqCycle¹ is an open source solution for Toopcase (hosted on gitHub), and Yakindu Crema², which is commercial and closed-source.

Assessor 1: For me, Traceability is one of the bigger points which needs extension. Also, usability is not fully achieved with the current implementation (i.e., we do not have a proper way to make a document out of the database).

But, I see also a good chance to improve the functionality of the tool in an openSource project like openETCS.

Assessor 2: I agree with the author’s comments. Data and function management should probably not be managed at requirements level but requirements should be able to identify and reflect them as mentioned. I am more severe for version management as this feature is quite important for the industry so we probably need to check it in more details.

According the results of this table, some of the following sections can be skipped.

A.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.

A.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) :

¹<http://www.eclipsecon.org/france2013/sessions/reqcycle-coming-some-details>

²<http://www.guersoy.net/knowledge/crema>

	Author	Assessor 1	Assessor 2	Total
Open Source (D2.6-02-074)	3	3	3	9
Portability to operating systems (D2.6-02-075)	3	3	3	9
Cooperation of tools (D2.6-02-076)	3	3	3	9
Robustness (D2.6-02-078)	3	2	2	7
Modularity (D2.6-02-078.1)	3	2	3	8
Documentation management (D2.6-02-078.02)	2	2	2	6
Distributed software development (D2.6-02-078.03)	2	2	2	6
Simultaneous multi-users (D2.6-02-078.04)	2	2	2	6
Issue tracking (D2.6-02-078.05)	1	1	1	3
Differences between models (D2.6-02-078.06)	2	2	2	6
Version management (D2.6-02-078.07)	2	1	1	4
Concurrent version development (D2.6-02-078.08)	2	1	1	4
Model-based version control (D2.6-02-078.09)	3	2	2	7
Role traceability (D2.6-02-078.10)	1	1	1	3
Safety version traceability (D2.6-02-078.11)	2	2	2	6
Model traceability (D2.6-02-079)	2	2	2	6
Tool chain integration	3	3	2	8
Scalability	3	3	3	9
User Friendliness	2	1	2	5

Author: Generally, the data structures of the underlying requirements model will not change, as they are based on an international standard, ReqIF 1.0.1 [12]. These data structures are powerful enough to model pretty much everything described here. However, just the existence of the right data structures does not mean that they are used properly. In many cases, it would be preferable to integrate a separate tool (e.g. Mylyn for issue tracking), or to constrain the behavior of the tool programmatically. The bottom line is that for many questions the answer is: Yes, it is possible, but not very comfortable. For a comfortable solution, development resources are required.

Assessor 1: With ProR and its indirect interface to Doors I see a great chance for openETCS. We have the option to use existing requirements documents and standards in the project. But, on the other hand, I was not able to test this so far. The lack of some basic functions limits the useability in the SSRS team. We need a priority list on things to do.

Assessor 2: Although ProR is Open Source, it relies on closed source software for some features like importing/exporting in several (closed) formats. This is not a show stopper for using ProR as it supports ReqIF (enabling tool cooperation and most industrial own the software enabling those specific features. Documentation is still minimal however come with a tutorial and accessible within the tool. Robustness looks good but not sure about the test level as the project is still in incubation (test are present in sources but did not check in detail nor try to run them). About toolchain

integration, good through model (ReqIF exchange) but maybe check about support of protocols like OSLC. The project published some evidence of scalability and performance on the web: <http://nirmalsasidharan.wordpress.com/2011/08/03/2-fast-thrice-furious>. Support of multiple/simultaneous users is possible at EMF level but there is no mention of whether/how it is available. Same for versioning which is only available through generic support thus not aware of the requirements model structure.

A.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	2	2	6
Proof of correctness (D.2.6-01-42.03)	*	*	*	*
Existing industrial usage	2	1	1	4
Model verification	1	1	1	3
Test generation	*	*	*	*
Simulation, execution, debugging	*	*	*	*
Formal proof	*	*	*	*

Author: The sections marked with asterisk are not applicable.

Model verification. There is ongoing research in supporting V&V activities by establishing a traceability to the corresponding model [6].

Assessor 1: I fully support the arguments of my co-assessor.

Assessor 2: I agree with non-applicable sections as stated by the author. Note however that requirements can be traced to generated tests and to formal proofs (property to verify). About industrial usage, some interested parties are listed in: <http://www.eclipse.org/proposals/modeling.mdt.rmf> but some concrete reference would be welcome.

Which scope of qualification is expected according EN50128 (section 6.7)?

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	2	1	2	5
class T2	*			*
class T3	*			*

Author: I don't quite understand this section. ProR should be classified as T1 and has never been qualified.

Assessor 1: I would not underestimate the effort to achieve a classification. I also see eclipse not as structured as needed for the classification.

*Assessor 2: I agree with the author's comment: requirements management tool is T1
The strong process imposed by Eclipse should help to achieve qualification.*

Other elements for tool certification

A.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.

A.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	3	2	3	8
Scade method	3	2	2	7
EFS language	3	2	3	8
B Method	3	*	3	6*
C language	*	*	*	*

SysML

How the means or tools can complete SysML ?

Author: SysML itself provides a requirements element. However, this is little more than a box with text in it. We already realized a prototypical implementation, where the text in the SysML box is kept in sync with an attribute of a ProR specification.

A good SysML integration would go much further, allowing diagrams (e.g. state diagrams) to be inserted into the requirements text, or to color highlight symbols in the requirements text.

Assessor 1: I do believe we can have a nice and user-friendly interface to the languages. But I also believe there is still some way to walk.

Assessor 2: There is a large potential of complementarity as mentioned by the author.

Scade, EFS, Classical B

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

Author: By providing a traceability between textual requirements and model elements.

Assessor 1: I do believe we can have a nice and user-friendly interface to the languages. But I also believe there is still some way to walk.

Assessor 2: Evidence of traceability has been shown on Event-B (RODIN) and through CREMA in a more general perspective. It is feasible but not provided out-of-the box. It will probably be more difficult to achieve for Scade as it is closed source.

C language

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

Author: This is not clear. Conceivable are traceability to code, or incorporation of test results. How useful this would be and how much effort would be required depends on the underlying process.

A.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	3	3	9
Papyrus	2	2	2	6
Scade	2	2	2	6
EFs tools	2	2	2	6
B tools	*	*	*	*

Author: B tools. For Event-B, an integration already exists. Classical B is much trickier, as has been described below.

Assessor 1: I'm mostly interested in the integration with SysML Papyrus. We should have this part on priority. Scade might be something we can cover temporarily with closed source tools since Scade as such is closed source. But we need to cover the migration path e.g. from Reqtify to ProR.

Assessor 2: Eclipse integration is available. Papyrus, Scade (Papyrus), EFS looks feasible through EMF. B (if applied at component level) comes later in the toolchain and probably not directly integrated.

Eclipse

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

Author: ProR is Eclipse-based.

Papyrus

How the means or tools can complete Papyrus ?

Author: As Papyrus is based on EMF, it is possible to implement a traceability based on the EMF model. This has been done in the past (Event-B). It's fairly straight forward with textual elements, and a little more involved with graphical elements, like state diagrams.

Scade, EFS, Classical B

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

Author: To create a traceability, ProR would need access to the other tool's model. If the tool writes XML, this is relatively easy. For a textual language like Classical B thinks are trickier, as we would need a parser. Further, in Classical B, not all elements have unique identifiers, making things even more tricky.

A.4 Means and tools for data, function and requirement management

This section defines the criteria for the means and tools dedicated to data, function and requirement management. These activities are shared by the work packages WP3, WP4 and the activities dedicated to SSRS. These means and tools shall integrate the primary toolchain to complete its gap and facilitate the integration of different activities. First of all, they allow the management of a common repository of data, functions and requirements, shared between the models (from SSRS informal specification to code) and the verification and validation activities. Then, they shall support traceability of requirements between models and activities, and facilitate the verification of the traceability. Besides they shall support the design of SIL4 software with model comparison or document production facilities, and version management.

A.4.1 Management activities

Which activities, linked to help the management of SSRS definition and whole process are covered by the mean or tool ?

	Author	Assessor 1	Assessor 2	Total
Requirement capturing	3	3	3	9
Requirement management	3	3	3	9
Data management	1	1	1	3
Function management	1	1	1	3
Requirement traceability	3	3	3	9
Model traceability	2	2	2	6
Function architecture	1	1	1	3
Version management	2	2	2	6
Model comparison	2	2	2	6
Documentation production	2	2	2	6
Others (give details)	n/a		n/a	n/a

A.4.2 Input Artifacts

Which artifacts are used as input of the mean or tool ?

	Author	Assessor 1	Assessor 2	Total
Informal description	3	3	3	9
Structured description	3	3	3	9
Spread sheet	3	1	2	6
XML files	3	3	2	8
EFS model	1	1	1	3
DSL	2	2	2	6
Others (give details)	n/a			n/a

Author:

Informal/Structured description. The only input format ProR currently accepts is ReqIF. However, ReqIF can be created with many tools, including Rational DOORS. At least for the time being, this could be used as a universal converter, allowing Word, Spreadsheets, plain text, etc. to be converted to ReqIF.

EFS model. It would make little sense to “convert” EFS to ReqIF. Instead, it would make sense to create an adapter that would allow traceability to/from EFS.

DSL. There have been prototypical implementations of XText (DSL framework) and ProR. Such an implementation allows to edit in the ProR-cells with a DSL editor.

Assessor 1: I would like to see more flexibility in the interface to spreadsheets. Import of data from spreadsheets would allow for more efficiency in daily work.

Assessor 2: Format other than ReqIF require an external tool. The way to manage import/export (possibly partial) is not well documented.

A.4.3 Output Artifacts

Which artifacts are used as output of the mean or tool ?

	Author	Assessor 1	Assessor 2	Total
Informal description	2	1	2	5
Structured description	3	3	2	8
Spread sheet	2	1	2	5
XML files	3	3	3	9
EFS model	0	0	0	0
DSL	2	2	2	6
Others (give details)				

Assessor 1: We need some means to transport the collected requirements into a professional document. Example: printout of requirements for review with partners not being familiar with ProR. The interface to spreadsheets would allow for easy enhancement of quite sophisticated functionality (e.g., filtering, pivots).

Assessor 2: Export require external converter. There does not seem to be a report generator nor direct way to query the model at UI level (e.g. using an OCL query) although this is probably easy to achieve using the ReqIF API or EMF. The extension point provided may also help.

A.4.4 Requirement Management

This section is link to requirement definition and management activities.

Are these criteria covered by the tool or mean ?

	Author	Assessor 1	Assessor 2	Total
Editing of Textual Requirements	3	3	3	9
Represent Relations between Req (Textual-Based)	3	3	3	9
Represent Relations between Req (Modelling-Based)	2	2	2	6
Glossary and Abbreviation handling (Linked to Req))	2	2	2	6
Traceability of Textual Requirements to Modelling	2	2	2	6
Import/Export of Industrial Standard Data (e.g., REQIF)	3	3	3	9
Documentation generation	2	1	1	4
Search and Filtering functions	1	1	1	3
Others (give details)				

Assessor 1: We need to see how we can get an easy and user-friendly workaround for those missing functions.

Assessor 2: There does not seem to be a report generator nor direct way to query the model at UI level (e.g. using an OCL query) although this is probably easy to achieve using the ReqIF API or EMF. The extension point provided may also help.

A.5 Other comments

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

Appendix B: Scade

No results of evaluation.

Appendix C: Rodin

Partial evaluation. Slides available on gothub https://github.com/openETCS/model-evaluation/blob/master/Telco_Secondary_slides/Systerel_Event-B.pdf.

C.1 Author and Assessors

Author Matthias Güdemann — Systerel

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

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

C.2 Presentation

Name Event-B and the Rodin platform

Web site <http://www.event-b.org>

Licence Common Public License Version 1.0 (CPL)

Abstract

Rodin is an open source tool for formal modeling and verification on the system level using the Event-B formalism. Event-B is based on set-theoretic notation of first-order logic (FOL) and has its roots in the B method which has a long history of successful application in industry on software level development.

Rodin is fully integrated into the Eclipse platform and is therefore fully extensible through plug-ins. Existing plug-ins include graphical modeling using state-machines, model simulators, modern state-of-the art SMT solvers and Rational DOORS interoperable requirements tracing using ReqIf documents and ProR.

Publications

- The leaflet [14] contains a short overview of the Rodin tool
- The book [5] explains the usage of Rodin and serves as a gentle introduction into Event-B modeling in Rodin
- The book [2] contains an extensive presentation of Event-B an several modeling examples for different system
- The scientific journal article [3] contains an in-depth look at the integration of Event-B into the Rodin platform

For which activities are dedicated the means or tools (give a note from 0 to 3) :

	Author	Assessor 1	Assessor 2	Total
Data Management	0			
Function Management	2			
Requirement Management	2			
Version Management	2			
Other (give details below)	3			

Author: Rodin is a specialized tool to formally model and verify abstract functional behavior. Therefore data management is not in its scope, as this is clearly a lower level detail aspect, more on the implementation level.

Function Management: A Rodin model contains high level function descriptions, i.e., an abstract view of the observable system behavior and its effect on the system state. It is therefore well suited to be included in function management, by formalizing the abstract behavior of the functions, tracing any changes and observing their effect on the intended functioning of the system.

Version Management: Rodin does not contain a version management itself. Its files are based on XML, therefore any modern version control system can be used, in particular those (like svn/mercurial/git) for which an Eclipse plug-in exists. There also exists a pug-in that is compatible to model-compare in Eclipse, i.e., allows for comparison on the model level instead of text level.

Other: Rodin can provide an important support for **traceability**, which is missing here. It allows for linking formal model aspects to a requirements document, e.g., a ReqIf document in ProR. Any changes in the specification can therefore be traced in the formal Event-B model and system-level aspects can be formally verified.

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)	3			
Portability to operating systems (D2.6-02-075)	3			
Cooperation of tools (D2.6-02-076)	3			
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)	2			
Issue tracking (D2.6-02-078.05)	2			
Differences between models (D2.6-02-078.06)	2			
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)	2			
Role traceability (D2.6-02-078.10)	1			
Safety version traceability (D2.6-02-078.11)	3			
Model traceability (D2.6-02-079)	3			
Tool chain integration	3			
Scalability	2			
User Friendliness	2			

Author: Rodin is based on Eclipse, therefore existing plug-ins can be used for many of the above aspects. Many of those are applicable without any changes, for others, some Rodin / Event-B specific modifications might be necessary.

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)	3			
Proof of correctness (D.2.6-01-42.03)	2			
Existing industrial usage	3			
Model verification	3			
Test generation	0			
Simulation, execution, debugging	3			
Formal proof	3			

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	2			
class T2	2			
class T3	0			

Author: The Rodin tool aims at system-level analysis, therefore it will not be necessary to qualify it as T3 tool, as no output is generated that can directly contribute to the executable code.

Other elements for tool certification

C.4 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.4.1 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	1			
EFS language	0			
B Method	3			
C language	2			

SysML

How the means or tools can complete SysML ?

Author: Rodin allows graphical modeling of (UML) state machines, which are encoded into Event-B models. SysML state machines are very similar to this and with a bit of effort could be supported directly.

Scade, EFS, Classical B

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

Author: A light-weight interoperability with SCADE is possible, either via SCADE Systems which uses SysML or via SCADE state machines. This would allow a larger effort for integration. The data-flow part of SCADE does not seem to be applicable in an Event-B model.

As Event-B has its roots in the B language, several aspects of these languages are definitively compatible. For example the invariant predicates of Event-B can directly be used in a lower level B model. If the abstraction levels for data are not the same, an additional refinement step could be added to solve this problem.

There does not seem to be a good interoperation possibility with the EFS language.

C language

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

Author: A possible combination of an Event-B model and a C implementation is to use the predicate logic invariants as C asserts and the guards as preconditions of functions. As the abstraction level of the C implementation is much lower than the Event-B models, this would require some work to identify the right functions and data formats or to introduce higher level wrapper functions similar to Event-B events. Such asserts and pre-conditions could be verified by tools like SPARK, why3 etc.

C.4.2 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	2			
Scade	1			
EFS tools	1			
B tools	2			

Eclipse

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

Author: The Rodin platform is fully based on Eclipse.

Papyrus

How the means or tools can complete Papyrus ?

Author: The existing graphical modeling plug-ins for Rodin could be connected to Papyrus. This would require the development of a transformation of the different formats.

Scade, EFS, Classical B

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

Author: With SCADE there could be the possibility of interoperation via the SCADE System SysML framework.

With Classical B tools, there is the possibility to generate predicates for guards and invariants directly from the Event-B model. As classical B is based on text files and Event-B on XML file, there would be some development work to do.

For the EFS tools there are some interoperation possibilities on the EMF level, as both Rodin and EFS have an EMF model of the artifacts. However, as seen in the section above, how the two languages could interoperate is not clear.

C.5 Means and tools for data, function and requirement management

This section defines the criteria for the means and tools dedicated to data, function and requirement management. These activities are shared by the work packages WP3, WP4 and the activities dedicated to SSRS. These means and tools shall integrate the primary toolchain to complete its gap and facilitate the integration of different activities. First of all, they allow the management of a common repository of data, functions and requirements, shared between the models (from SSRS informal specification to code) and the verification and validation activities. Then, they shall support traceability of requirements between models and activities, and facilitate the verification of the traceability. Besides they shall support the design of SIL4 software with model comparison or document production facilities, and version management.

C.5.1 Management activities

Which activities, linked to help the management of SSRS definition and whole process are covered by the mean or tool ?

	Author	Assessor 1	Assessor 2	Total
Requirement capturing	3			
Requirement management	2			
Data management	0			
Function management	2			
Requirement traceability	3			
Model traceability	3			
Function architecture	2			
Version management	2			
Model comparison	2			
Documentation production	1			
Others (give details)				

Author: The requirements managing in Rodin is always done in close interoperation with ProR and one or more ReqIf documents. Via this plug-in, requirements can be linked directly to Event-B model artifacts, changes in either the model or the requirement document are marked for review.

The formalization of the requirements in Rodin allows for a much more accurate analysis of consistency and correctness. This can be particular helpful at times of changes to the requirements, which can be very difficult to assess without tool support.

Up to now, the document production integrated into Rodin is limited to the generation of a Latex representation of the model. This could be extended to also generate the information of elements linked to requirements in the requirements document.

C.5.2 Input Artifacts

Which artifacts are used as input of the mean or tool ?

	Author	Assessor 1	Assessor 2	Total
Informal description	1			
Structured description	1			
Spread sheet	1			
XML files	3			
EFS model	1			
DSL	1			
Others (give details)				

Author: Rodin uses XML files as input means. All other formats would require the development of input filters.

C.5.3 Output Artifacts

Which artifacts are used as output of the mean or tool ?

	Author	Assessor 1	Assessor 2	Total
Informal description	0			
Structured description	2			
Spread sheet	0			
XML files	2			
EFS model	0			
DSL	0			
Others (give details)	3			

Author: Rodin produces documentation output of its models in Latex format and formal proof trees are saved in XML files. In general, its main output is a **formal description** of the predicates and proofs of the formal invariants for the system level functional model.

C.5.4 Requirement Management

This section is link to reauirement definition and management activities.

Are these criteria covered by the tool or mean ?

	Author	Assessor 1	Assessor 2	Total
Editing of Textual Requirements	3			
Represent Relations between Req (Textual-Based)	3			
Represent Relations between Req (Modelling-Based)	3			
Glossary and Abbreviation handling (Linked to Req))	2			
Traceability of Textual Requirements to Modelling	3			
Import/Export of Industrial Standard Data (e.g., REQIF)	2			
Documentation generation	1			
Search and Filtering functions	1			
Others (give details)				

Author: Rodin supports requirements managing with the ProR plug-in. Via its EMF model, its artifacts can be linked to elements of a ReqIf file via drag and drop. This allows for good traceability, as every change can be marked for later validation.

Textual requirements tracing is also possible by using comments in the Event-B model. Search and filtering is partially supported by the Event-B editor, a better integration, more focused on requirements traceability could be developed with some effort.

C.6 Other comments

Author: In general, the Rodin platform can give more confidence in the completeness and correctness of the requirements on the system level. The formalization allows to identify contradictions and missing elements in the specification.

Appendix D: Matelo

No results of evaluation.

Appendix E: Goal Structuring Notation

No results of evaluation.

Appendix F: EMF Store

No results of evaluation.

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

Appendix G: EMF Client Platform

No results of evaluation.

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

Appendix: References

- [1] Eclipse Requirements Modeling Framework. <http://eclipse.org/rmf>.
- [2] Jean-Raymond Abrial. *Modeling in Event-B: System and Software Engineering*. Cambridge University Press, New York, NY, USA, 1st edition, 2010.
- [3] Jean-Raymond Abrial, Michael Butler, Stefan Hallerstede, Thai Son Hoang, Farhad Mehta, and Laurent Voisin. Rodin: an open toolset for modelling and reasoning in Event-B. *STTT*, 12(6):447–466, 2010.
- [4] Sylvain Baro and Jan Welte. Requirements for openETCS. Technical Report D2.6, OpenETCS, 2013.
- [5] Michael Jastram (Ed.). Rodin user’s handbook. <http://handbook.event-b.org>, 2012.
- [6] Stefan Hallerstede, Michael Jastram, and Lukas Ladenberger. A method and tool for tracing requirements into specifications. page 36 p, 2013. Science of Computer Programming.
- [7] Michael Jastram. A Systems Engineering Tool Chain Based on Eclipse and Rodin. In *Forms/Format*, 2012. http://www.stups.uni-duesseldorf.de/w/Special:Publication/jastram_forms_2012.
- [8] Michael Jastram. *Managing Requirements Knowledge*, chapter 16: The Eclipse Requirements Modeling Framework, pages 353–372. Springer, March 2013. http://www.stups.uni-duesseldorf.de/w/Special:Publication/RMF_Mark_Book_Jastram_2013.
- [9] Michael Jastram and Andreas Graf. Requirement traceability in Topcased with the requirements interchange format (RIF/ReqIF). *First Topcased Days Toulouse*, 2011. <http://www.stups.uni-duesseldorf.de/w/Special:Publication/topcase-JaGr2011>.
- [10] Matthias GÜdemann. openETCS Event-B Benchmark. https://github.com/openETCS/model-evaluation/blob/master/model/Event_B_Systemrel/rodin-projects-github.pdf?raw=true, 2013.
- [11] Michael Jastram. openETCS Benchmark: Using Eclipse ProR for openETCS. https://github.com/openETCS/model-evaluation/blob/master/management/ProR_FormalMind/pror_evaluation.pdf, 2013.
- [12] OMG. Requirements Interchange Format (ReqIF) 1.0.1. <http://www.omg.org/spec/ReqIF/>, 2011.
- [13] Marielle Petit-Doche and WP7 Participants. D7.1: Report on the final choice of the primary toolchain. Primary Toolchain OETCS/WP7/D7.1, openETCS, July 2013.
- [14] Systemel. http://sourceforge.net/projects/rodin-b-sharp/files/Doc_Rodin_General/Rodin_Leaflets/Leaflet_Rodin_E.pdf/download, 2012.