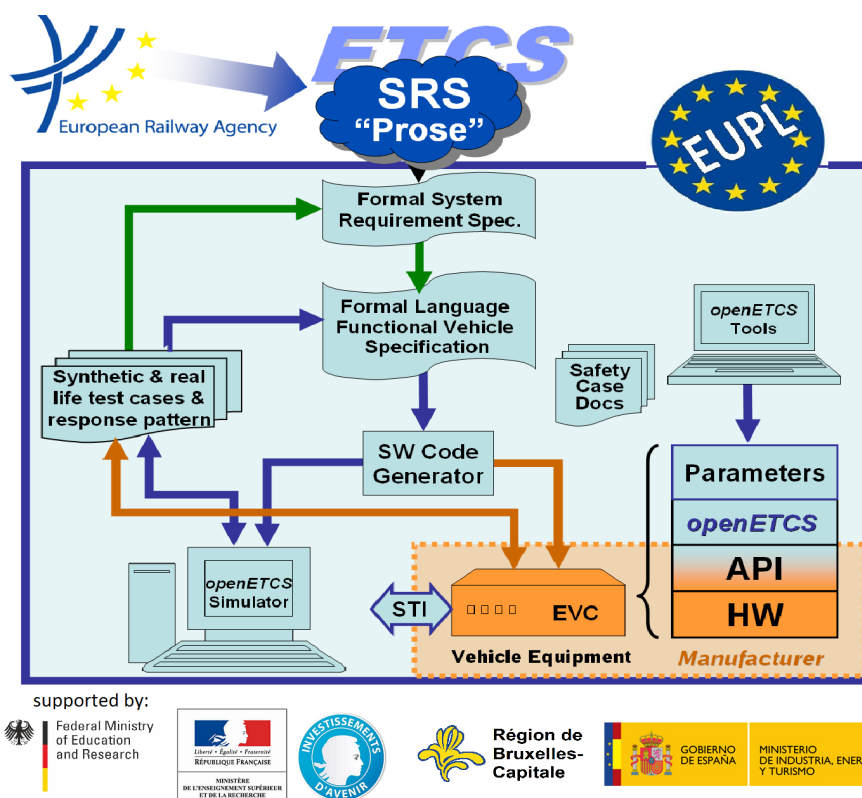Work-Package 7: "Secondary tools"

Report on all aspects of secondary tooling

Results of T7.2

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

December 2013

supported by:

This page is intentionally left blank

Work-Package 7: "Secondary tools"   **OETCS/WP7/D7.2 – 00/01**
                                                               December 2013

Report on all aspects of secondary tooling
Results of T7.2

Marielle Petit-Doche

Systerel

all participants of the benchmark

WP7 partners

all participants of VnV and Safety process

WP4 partners

Deliverable

Prepared for   openETCS@ITEA2 Project

Abstract: This document gives results of the evaluation and selection of the tools and methods to complete the secondary toolchain and to support verification and validation activities, safety activities, model transformation and data management for the whole project.

Table of Contents

Figures and Tables

Figures

Tables

Document information	
Work Package	WP7
Deliverable ID or doc. ref.	D7.2
Document title	Report on all aspects of secondary tooling
Document version	00.01
Document authors (org.)	Marielle Petit-Doche (Systerel)

Review information	
Last version reviewed	00.01
Main reviewers	

Approbation			
	Name	Role	Date
Written by	Marielle Petit-Doche	WP7-T7.2 Sub-Task Leader	
Approved by	Michael Jastram	WP7 leader	

Document evolution			
Version	Date	Author(s)	Justification
00.01	19/12/2013	M. Petit-Doche	Document creation

1 Introduction

The aim of this document is to report the results of the evaluation of means and tools for the secondary toolchain, 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:

- data, function and requirement management (SSRS, WP3 and WP4), in chapter **??**;

- verification and validation (WP4), in chapter **??**;

- safety activities support (WP4), in chapter **??**;

- model transformation and code generation (WP3 and WP4), in chapter **??**.

1.1 Approach

(mj) Content to be discussed in https://github.com/openETCS/toolchain/issues/236

2   Data and Requirements Management

This section is dedicated to tools and means to support management of data, functions requirements and other artifacts along the openETCS process.

In total, seven tools have been proposed. Out of these, only one has been evaluated in detail (ProR). What follows is a qualitative description of the seven tools. A quantitative evaluation of ProR is included as well.

2.1   Candidates

The list of initial candidates is:

Scade Suite. Scade includes Reqtify as the requirements traceability solution. It allows to create traceability directly to Word, thereby making traceability to Subset-26 easy. However, there is no clear solution for authoring additional requirements (except using Word). Further, it is not clear how traceability to model artifacts should be realized. Last, this is a closed source solution and therefore only a last resort.

Rodin and Pluggin.

Matelo.

Goal Structuring Notation (GSN).

Eclipse ProR.

Eclipse EMF Store.

Eclipse EMF Client Platform.

During the evaluation phase, a number of challenges were identified that were not clearly defined before. The list of challengers discussed during the evaluation is:

- Ecore model + XML files
- UML library
- ReqCity

2.1.1   ProR Evaluation

Name Eclipse ProR (part of the Eclipse Requirements Modeling Framework (RMF))

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

Licence Eclipse Public License

Abstract

ProR is a tool for requirements engineering that supports the ReqIF 1.0.1 Standard natively. It is part of the Eclipse RMF project and is built for extensibility. ProR is the result of the European Union FP7 Research Project Deploy, where it is used to provide traceability between requirements and formal models.

Vision: Our vision is to provide a tool that supports interoperability by operating directly of the ReqIF data standard, and that supports integration by taking advantage of the extendability of the Eclipse platform. ProR is targeted to academic and industrial users alike.

Publications

Wikipedia: Requirements Modeling Framework. The Requirements Modeling Framework (RMF) is an Open-Source-Framework for working with requirements based on the ReqIF standard. RMF consists of a core allowing reading, writing and manipulating ReqIF data, and a user interface allowing to inspect and edit request data.

RMF is the first and, currently, the only open source reference implementation of the ReqIF standards. Noteworthy is the fact that RMF has already been deployed in the ProStep ReqIF Implementor Forum in order to ensure the interoperability of the commercial implementation. Since 2011 there have been reports in the German and in the international press about RMF. (more)

The Eclipse Requirements Modeling Framework. In Managing Requirements Knowledge, Springer, 2013. Michael Jastram.

This chapter presents the the Requirements Modeling Framework (RMF), an Eclipse-based open source platform for requirements engineering. The core of RMF is based on the emerging Requirements Interchange Format (ReqIF), which is an OMG standard. The project uses ReqIF as the central data model. At the time of this writing, RMF was the only open source implementation of the ReqIF data model.

By being based on an open standard that is currently gaining industry support, RMF can act as an interface to existing requirements management tools. Further, by based on the Eclipse platform, integration with existing Eclipse-based offerings is possible.

In this chapter, we will describe the architecture of the RMF project, as well as the underlying ReqIF standard. Further, we give an overview of the GUI, which is called ProR. A key strength of RMF and ProR is the extensibility, and we present the integration ProR with Rodin, which allows traceability between natural language requirements and Event-B formal models.

Openness in Systems Engineering with Eclipse ProStep Symposium, 2013. Michael Jastram.

Eclipse is an open source framework for building platform-independent GUI applications. It is managed by the Eclipse Foundation (a non-profit organization), which ensures that official Eclipse projects are interoperable and follow certain intellectual property guidelines. Open Source in general allows organizations to remedy the risk of being dependent on one single vendor. This includes the risk of the feature set provided: users can add missing features themselves or commission their inclusion to any competent party, rather than having to rely on the vendor to implement it. It further includes the risk of maintenance and long-term support. Eclipse in particular provides a solid, mature and open platform for desktop applications with a rich ecosystem. Many Eclipse offerings are ready to be used "as is", thereby offering great cost savings.

In this talk, we demonstrate how Eclipse can be used as an integration platform for systems engineering. We focus on RMF (Requirements Modeling Framework) as a case study on how the Eclipse ecosystem can be leveraged in a business environment. RMF is a clean-room implementation of the open ReqIF standard, which is currently being adopted by various tool vendors: The currently ongoing ReqIF Implementor Forum , which is organized by ProSTEP iViP, will ensure that the various ReqIF implementations will properly function together. We will look at both the technical and business implications.

From a business point of view, this approach promises cost savings and prevents vendor lock-in. To understand the value, we will look at the openETCS project , which is an ITEA2 EU-funded project. The purpose of this project is the development of an integrated modeling, development, validation and testing framework for leveraging the cost-efficient and reliable implementation of the European Train Control System (ETCS), based on open source technologies. While the technology choice has not yet been fi¬nalized, Eclipse is a strong candidate for realizing this project, and it being open source is a core requirement. We will present the implications of such an open platform from a business point of view for the parties involved, which are customers (e.g. Deutsche Bahn), equipment manufacturers (e.g. Siemens) service providers (e.g. Formal Mind) and, of course, the EU and its citizens.

ReqIF-OLUTION: Mit Eclipse und ReqIF zur Open-Source ALM-Werkzeugkette ObjektSpektrum, 3, 2013. Michael Jastram.

Der Austausch von Anforderungen war bisher entweder mit Datenverlust oder verlustfrei nur über proprietäre Wege zu realisieren. Aber letztes Jahr wurde mit ReqIF ein internationaler Standard von der OMG verabschiedet, der dieses Problem löst. Dies hat eine Lawine von Aktivitäten ausgelöst, einschließlich der Entwicklung einer OpenSource Referenzimplementierung (Eclipse RMF). In diesem Artikel zeigen wir, wie mit ReqIF, Eclipse und RMF mit wenig Aufwand eine ALM-Werkzeugkette realisiert werden kann, und wo die Reise hingeht.

The ProR Approach: Traceability of Requirements and System Descriptions. CreateSpace, 2012. Michael Jastram.

Creating a system description of high quality is still a challenging problem in the field of requirements engineering. Creating a formal system description addresses some issues. However, the relationship of the formal model to the user requirements is rarely clear, or documented satisfactorily.

This work presents the ProR approach, an approach for the creation of a consistent system description from an initial set of requirements. The resulting system description is a mixture of formal and informal artefacts. Formal and informal reasoning is employed to aid in the process. To achieve this, the artefacts must be connected by traces to support formal and informal reasoning, so that conclusions about the system description can be drawn.

The ProR approach enables the incremental creation of the system description, alternating between modelling (both formal and informal) and validation. During this process, the necessary traceability for reasoning about the system description is established. The formal model employs refinement for further structuring of large and complex system descriptions. The development of the ProR approach is the first contribution of this work.

This work also presents ProR, a tool platform for requirements engineering, that supports the ProR approach. ProR has been integrated with Rodin, a tool for Event-B modelling, to provide a number of features that allow the ProR approach to scale.

The core features of ProR are independent from the ProR approach. The data model of ProR builds on the international ReqIF standard, which provides interoperability with industrial tools for requirements engineering. The development of ProR created enough interest to justify the creation of the Requirements Modeling Framework (RMF), a new Eclipse Foundation

project, which is the open source host for ProR. RMF attracted an active community, and ProR development continues. The development of ProR is the second contribution of this work.

This work is accompanied by a case study of a traffic light system, which demonstrates the application of both the ProR approach and ProR.

A Systems Engineering Tool Chain Based on Eclipse and Rodin In Forms/Format, 2012. Michael Jastram.

Formal methods are experiencing a renaissance, especially in the development of safety-critical systems. An indicator for this is the fact that more and more standards either recommend or prescribe the use of formal methods.

Using formal methods on an industrial scale requires their integration into the system engineering process. This paper is exploring how an integrated tool chain that supports formal methods may look like. It thereby focusses on our experience with tool chains that are based on the open source Eclipse platform in general, and the Rodin formal modeling environment in particular.

Open Source allows organisations to remedy the risk of being dependent on one single vendor. This includes the risk of the feature set provided: users can add missing features themselves or commission their inclusion to any competent party, rather than having to rely on the vendor to implement it. It further includes the risk of maintenance and long-term support.

We see industrial interest in open source for systems engineering in general, and Eclipse in particular. Eclipse is attractive, because its license is business-friendly. Further, its modular architecture makes it easy to seamlessly integrate the various Eclipse-based tools for systems engineering.

This paper focuses on an ecosystem that is accumulated around two Eclipse-based platforms, First, the Rodin platform is an open source modeling environment for the Event-B formalism. Second, the Requirements Modeling Framework (RMF) is a platform for working with natural language requirements, supporting the international ReqIF standard.

Requirements Modeling Framework Eclipse Magazin, 6.11, 2011. Michael Jastram, Andreas Graf.

Im August 2011 hat das Requirements Modeling Framework (RMF) als ein neues Eclipse-Projekt das Licht der Welt erblickt. RMF besteht aus einem Kern, der Daten im Requirements Exchange Format (RIF/ReqIf) verarbeiten kann, und einem GUI namens ProR, mit dem sich diese Daten komfortabel bearbeiten lassen. ProR stellt einen Extension Point zur Verfügung, der die Integration mit anderen Werkzeugen ermöglicht.

Requirement Traceability in Topcased with the Requirements Interchange Format (RIF/ReqIF) First Topcased Days Toulouse, 2011. Michael Jastram, Andreas Graf.

One important step of the systems engineering process is requirements engineering. Parallel to the development of Topcased, which includes tooling for requirements engineering, a new standard for requirements exchange is emerging at the OMG under the name "ReqIF" (formally called RIF). In our talk we introduce the activities of two research projects and their tool developments, VERDE (Yakindu Requirements) and Deploy (ProR) and discuss possible synergies with Topcased.

Requirements, Traceability and DSLs in Eclipse with the Requirements Interchange Format (RIF/ReqIF) Technical Report, Dagstuhl-Workshop MBEES 2011: Modellbasierte Entwicklung eingebetteter Systeme, 2011.

Requirements engineering (RE) is a crucial aspect in systems development and is the area of ongoing research and process improvement. However, unlike in modelling, there has been no established standard that activities could converge on. In recent years, the emerging Requirements Interchange Format (RIF/ReqIF) gained more and more visibility in industry, and research projects start to investigate these standards. To avoid redundant efforts in implementing the standard, the VERDE and Deploy projects cooperate to provide a stable common basis for RIF/ReqIF that could be leveraged by other research projects too. In this paper, we present an Eclipse-based extensible implementation of a RIF/ReqIF-based requirements editing platform. In addition, we are concerned with two related aspects of RE that take advantage of the common platform. First, how can the quality of requirements be improved by replacing or complementing natural language requirements with formal approaches such as domain specific languages or models. Second, how can we establish robust traceability that links requirements and model constructs and other artefacts of the development process. We present two approaches to traceability and two approaches to modelling. We believe that our research represents a significant contribution to the existing tooling landscape, as it is the first clean-room implementation of the RIF/ReqIF standard. We believe that it will help reduce gaps in often heterogeneous tool chains and inspire new conceptual work and new tools.

Added value for OpenETCS project

ProR fulfills the formal requirements to be part of the openETCS toolchain: It is based on Eclipse and EMF, and it is licensed under an appropriate open source license. It is mature enough to be used for managing natural language requirements.

Integration in OpenETCS process and toolchain

As of this writing, ProR is already integrated into the openETCS tool. As it is based on EMF, data integration should be reasonably straight forward.

2.1.2   EventB, Rodin and pluggins

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 **?** contains a short overview of the Rodin tool

- The book **?** explains the usage of Rodin and serves as a gentle introduction into Event-B modeling in Rodin

- The book **?** contains an extensive presentation of Event-B an several modeling examples for different system

- The scientific journal article **?** contains an in-depth look at the integration of Event-B into the Rodin platform

A quantitative evaluation is available in `https://github.com/openETCS/toolchain/blob/master/T7.2/O7.2.1_Safety/O7-2-1_Safety.pdf`

Added value for OpenETCS project

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.

Integration in OpenETCS process and toolchain

The Rodin platform is fully based on Eclipse.

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.

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.

2.2 Open issues

2.2.1 Traceability

TODO

2.2.2 How to Deal with Subset-26

TODO

2.3 Selected means and tools

Comment. To complete after decision meeting with a section for each tool with the following contents:

- *description of the means or tools, references and links*
- *added value for openETCS*
- *for which tasks and how (input/output/actions) is the mean or tools used.*

3 Verification and Validation

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

3.1 Candidates

The list of initial candidates is:

- 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

- SPIN

3.1.1 SystemC

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

A quantitative evaluation is available in `https://github.com/openETCS/toolchain/blob/master/T7.2/O7.2.1_VnV/O7-2-1_VnV.pdf`

Added value for OpenETCS project

Comment. To complete: Stefan Rieger ?

Integration in OpenETCS process and toolchain

Comment. To complete: Stefan Rieger ?

3.1.2 UPPAAL

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`

A quantitative evaluation is available in `https://github.com/openETCS/toolchain/blob/master/T7.2/O7.2.1_VnV/O7-2-1_VnV.pdf`

Added value for OpenETCS project

Comment. To complete: Stefan Rieger ?

Integration in OpenETCS process and toolchain

Comment. To complete: Stefan Rieger ?

3.1.3 CPN Tools

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

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

A quantitative evaluation is available in `https://github.com/openETCS/toolchain/blob/master/T7.2/07.2.1_VnV/07-2-1_VnV.pdf`

Added value for OpenETCS project

Comment. To complete: Stefan Rieger , Jan Welte ?

Integration in OpenETCS process and toolchain

Comment. To complete: Stefan Rieger , Jan Welte ?

3.1.4 RT-Tester

Name RT-Tester

Web site http://www.verified.de/en/products/rt-tester

Licence • Generator: closed

- SMT Solver: open
- GUI (eclipse plug-in): open

Abstract

The RT-Tester test automation tool, made by Verified, performs automatic test generation, test execution and real-time test evaluation. It supports different testing approach such as unit testing, software integration testing for component, hardware/software integration testing and system integration testing. The tool generates automatically behavioral tests that covers the classical criteria such as reachable state, branch and MC/DC. The RT-Tester also implements the so-called model-based testing approach: starting from a test model design with UML/SysML, the RT-tester fully automatically generates test cases. Moreover the test model may be directly linked to the requirements, thus the requirement coverage may also be ensured.

The RT-tester provides the following features :

- Automated Test Case Generation

- Automated Test Data and Test Oracles Generation

- Automated Test Procedure Generation

- Automated Requirement Tracing

- Test Management system

- Test Report Generation

Note that the RT-tester model based testing has be qualified according to ISO 26262.

Publications

- Jan Peleska, Elena Vorobev, and Florian Lapschies. Automated test case generation with smt-solving and abstract interpretation. In Mihaela Bobaru, Klaus Havelund, GerardJ. Holzmann, and Rajeev Joshi, editors, NASA Formal Methods, volume 6617 of Lecture Notes in Computer Science, pages 298–312. Springer Berlin Heidelberg, 2011.

- Jan Peleska, Elena Vorobev, Florian Lapschies, and Cornelia Zahlten. Automated model-based testing with RT-Tester. Technical report, Universität Bremen, 2011.

- Jörg Brauer, Jan Peleska, and Uwe Schulze. Efficient and trustworthy tool qualification for model- based testing tools. In Brian Nielsen and Carsten Weise, editors, Testing Software and Systems, volume 7641 of Lecture Notes in Computer Science, pages 8–23. Springer Berlin Heidelberg, 2012.

- Verified Systems International GmbH. Verified :: Products. http://www.verified.de/en/products.

Added value for OpenETCS project

RT-Tester will add an automated test case and test data generator that interprets test models directly describes in SysML. The test models may specify concurrent sub-components of SUT (System Under Test) and TE (Test Environment), and timing conditions using dense time (i. e., continuous physical time) and an arbitrary number of timers.

Another added value is that the tool also provides a simulation environment that can directly generate C code from the SysML model and run the tests. The code generator may also be adapted to other purposes.

Integration in OpenETCS process and toolchain

- GUI via an eclipse plug-in already exist,

- read SysML model

- Can run SCADE generated code

3.2 Selected means and tools

Comment. To complete after decision meeting with a section for each tool with the following contents:

- *description of the means or tools, references and links*
- *added value for openETCS*
- *for which tasks and how (input/output/actions) is the mean or tools used.*

4 Safety support

This section is dedicated to tools and means to support safety analyses.

4.1 Candidates

The list of initial candidates is:

- Rodin and Pluggins

- CPN tools

- Goal Structuring Notation (GSN)

- Safety Architect

4.1.1 EventB, Rodin and pluggins

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 **?** contains a short overview of the Rodin tool

- The book **?** explains the usage of Rodin and serves as a gentle introduction into Event-B modeling in Rodin

- The book **?** contains an extensive presentation of Event-B an several modeling examples for different system

- The scientific journal article **?** contains an in-depth look at the integration of Event-B into the Rodin platform

A quantitative evaluation is available in `https://github.com/openETCS/toolchain/blob/master/T7.2/07.2.1_Safety/07-2-1_Safety.pdf`

Added value for OpenETCS project

Rodin and Event-B do not directly support a hazard or risk analysis. Their goal is to strengthen the confidence in the correctness of an external safety analysis, by providing means to represent safety requirements (in particular functional requirements) in a formal model and to verify them there or to validate the intended behavior wrt. safety by simulating and observing the model.

Sub-system requirements can be specified and verified, if the formal model contains a representation of the sub-systems. While this can be achieved by refinement, it should be kept in mind that Event-B aims at system-level modeling and analysis, and therefore there could be better alternatives to analyze a very detailed model on implementation level.

The main application of Rodin is to formalize and verify the safety requirements where applicable. This supports the verification of the correctness of the arguments in the safety case, therefore strengthening the confidence in these arguments, but also to provide insight into probably lacking aspects of the safety case.

The Event-B approach is based on iterative refinements form the most abstract model to the desired level of detail. It is therefore a to-down approach, a bottom-up approach does not make sense using Event-B.

And database connection would require the development of additional plug-ins, but would be possible.

VnV of safety requirements is achieved by formal proof and simulation to validate correct functionality.

Traceability is achieved by the connection to ProR.

Generation of some documentation is already supported, as Latex documents can be generated from models. For more extensive documentation, e.g., links with safety requirements, some additional functionality would have to be developed.

Integration in OpenETCS process and toolchain

The Rodin platform is fully based on Eclipse.

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.

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.

4.2 Selected means and tools

Comment. To complete after decision meeting with a section for each tool with the following contents:

- *description of the means or tools, references and links*
- *added value for openETCS*
- *for which tasks and how (input/output/actions) is the mean or tools used.*

5   Model transformation and Code generation

This section is dedicated to tools and means for model transformation and code generation.

5.1   Candidates

The list of initial candidates is:

- Scade Suite

- Rodin and Pluggins

- Acceleo

- ATL

- QVTO and SmartQVT

- Xtend

5.1.1   Acceleo

Name Acceleo

Web site http://www.eclipse.org/acceleo/

Licence Eclipse

Abstract

Short abstract on the approach and tool (10 lines max) Acceleo is an implementation of the Object Management Group (OMG) MOF Model to Text Language (MTL) standard. Based on a special template language model to text transformations can be defined. It is fully integrated with Eclipse and also part of Polarsys.

Publications

Short list of publications on the approach (5 max) Most imformation is available on the homepage http://www.eclipse.org/acceleo/

Added value for OpenETCS project

Comment. To complete: Stefan Rieger ?

Integration in OpenETCS process and toolchain

Comment. To complete: Stefan Rieger ?

5.2   Selected means and tools

Comment.   To complete after decision meeting with a section for each tool with the following contents:

- *description of the means or tools, references and links*
- *added value for openETCS*
- *for which tasks and how (input/output/actions) is the mean or tools used.*

6   Conclusion

Comment.   MPD : Todo

To complete after Munich meeting in January 2014.