OETCS/WP7/O7.1.9 – 01/00

openETCS

ITEA2 Project
Call 6 11025
2012 – 2015

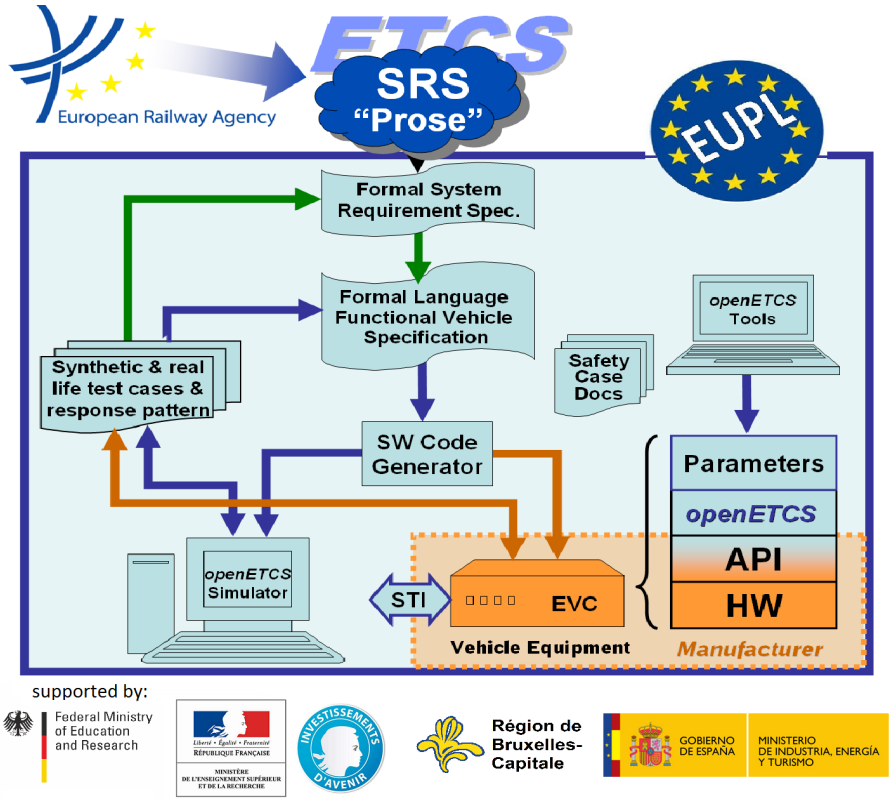
Work-Package 7: “Primary tool chain”

Evaluation of the tool platform against the WP2 requirements

List of criteria on tool platform and results on the benchmark

Cécile Braunstein

June 2013



This page is intentionally left blank

Work-Package 7: “Primary tool chain” **OETCS/WP7/O7.1.9 – 01/00**
June 2013

Evaluation of the tool platform against the WP2 requirements

List of criteria on tool platform and results on the benchmark

Cécile Braunstein

University Bremen

Prepared for ITEA2 openETCS consortium
Europa

Abstract: This document gives elements to evaluate the tool platform according WP2 requirements and presents the results of the evaluation of each tool platform.

Table of Contents

Figures and Tables

Figures

Tables

Document information	
Work Package	WP7
Deliverable ID or doc. ref.	O7.1.9
Document title	Evaluation of the tool platform against the WP2 requirements
Document version	02.00
Document authors (org.)	Cécile Braunstein (Uni. Bremen)

Review information	
Last version reviewed	00.01
Main reviewers	Marielle Petit-Doche

Approbation			
	Name	Role	Date
Written by	Cécile Braunstein	WP7 Contributor	01.06.2013
Approved by	Marielle Petit-Doche	WP7-T7.1 Sub-task leader	
Approved by	Michael Jastram	WP7 leader	

Document evolution			
Version	Date	Author(s)	Justification
00.00	28/05/2013	C. Braunstein	Document creation
00.01	05/06/2013	C. Braunstein	Marielle review incorporated
01.00	10/06/2013	C. Braunstein	Marielle review incorporated
02.00	19/07/2013	C. Braunstein	WP7 July meeting Jonas Helming Eclipse description

1 Introduction

This document is the results report of the evaluation of the tool platform. This work is the first part of the tool platform selection as described in the WP7 description of work [1]. This task aims at evaluating tool platform integration capabilities, regardless the primary and second tools that will be chosen for the tool chain implementation.

The tool platform should provide mechanisms to integrate various tools. The tool platform is not the primary nor secondary tools, nor the tool chain. It is the support for the tool chain implementation, it shall help to integrate the tools into a seamless tool chain. The evaluation will focus on the integration capabilities of the tool platform.

This document will set up a way to evaluate tool platform and to determine the set of best candidate according to the WP2 requirements [2]. This task will be followed by finding the best tool platform candidates, e.g.finding the best the adequacy between primary and secondary tools and the tool platform.

1.1 The tool integration problem

The tool platform raises the question of tool integration. Wassermann [3] defines the 5 problems of tool integration that must be addressed:

1. Platform integration: How to deal with heterogeneous OS ?
2. Data integration: How to share data and how to manage their relationships ?
3. Presentation integration: How to unify the user-interface, the “look and feel” ?
4. Control integration: How to enable service sharing between tools and how to notify one another of events ?
5. Process integration: How to support engineering process within the tool platform ?

More recently Asplund and al. in [4] add some metrics to evaluate the tool chain integration. Following these propositions **the tool platform evaluation will focus on these integration capabilities**.

Moreover, one of the openETCS’s goal is to develop a tool chain to generate a software meeting the CENELEC EN 50128:2011 requirements and certifiable SIL4. **The tool platform shall help to support the EN 50128 standard**.

The system integraty level of the product depends on the process defined by the tool chain. As shown by Asplund and al. in [5], sometimes reasoning tool by tool is not suffisient to analyze the resulted product. This implies that the tool chain should be considered as a whole and not only individual tools. Moreover, Asplund and al. as well as Slotoch and al. in [6] shows that taking a holistic approach and use some rearrangement or extension of the tool chain allow to avoid the qualification of all tools and mitigate the certification process. The tool chain doest not have to be certified but considering the tool chain as whole will help to identify which part should be qualified in order to obtain a certifiable software. The evaluation of **The tool platform with regards to the tool chain analysis capabilities** will be made.

1.2 Structure of the document

Chapter 2 presents the template of the evaluation criteria. The initial candidates where the following The appendix show the evaluation of each tool platform :

- Eclipse
- SCADE
- TopCased/Polarsys
- MONO
- RTP-Cesar

Nevertheless the only tool that can be considered as a tool platform is Eclipse, thus only this tool has been evaluated.

1.3 List of terms

Notation	Description
GUI	Graphical User Interface.
IDE	Integrated Development Environment.
OS	Operating System.
SIL	System Integrity Level.

1.4 Reference documents

[1] Michael Jastram, Marielle Petit-Doche, Jonas Helming, and Jan Peleska. openETCS toolchain WP7 descritpiton of work. Defin D01, openETCS, February 2013.

[2] Sylvain Baro and Jan Welte. Requirements for openETCS. Requirements D2.6, openETCS, April 2013.

[3] AnthonyI. Wasserman. Tool integration in software engineering environments. In Fred Long, editor, *Software Engineering Environments*, volume 467 of *Lecture Notes in Computer Science*, page 137–149. Springer Berlin Heidelberg, 1990.

[4] Fredrik Asplund, Matthias Biehl, Jad El-Khoury, and Martin Törngren. Tool integration beyond wasserman. In Camille Salinesi and Oscar Pastor, editors, *Advanced Information Systems Engineering Workshops*, number 83 in Lecture Notes in Business Information Processing, pages 270–281. Springer Berlin Heidelberg, January 2011.

[5] Fredrik Asplund, Jad El-khoury, and Martin Törngren. Qualifying software tools, a systems approach. *Computer Safety, Reliability, and \ldots*, page 340–351, 2012.

[6] Oscar Slotosch, Martin Wildmoser, Jan Philipps, Reinhard Jeschull, and Rafael Zalman. ISO 26262-tool chain analysis reduces tool qualification costs. *Automotive 2012*, 2012.

2 Template Description

Each tool platform candidates should fill the following sections

Name %%Name of the approach and the tool%%

Web site %%if available, how to find information%%

License %%Kind of license%%

2.1 Abstract

Short introduction to the tool

2.2 Open Source (D2.6-02-07)

Describes the license.

2.3 Long-Term Maintenance

maintenance infrastructure.

2.4 Portability (R-WP2/D2.6-02-076)

As the tools chain shall be portable to common OS, the tool platform too.

2.5 Tools Interoperability

How the tool platform support tool integration.

2.6 Modularity (D2.6-02-078.01)

How to perform modularity.

2.7 Framework Support

How the tool platform deals with framework integration.

Appendix A: Eclipse Platform

Name Eclipse and the Eclipse Modeling Framework (EMF)

Web site http://eclipse.org http://eclipse.org/emf

License EPL

A.1 Abstract

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

A.2 Open Source

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

A.3 Long-Term Maintenance

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

A.4 Portability

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

A.5 Tools Interoperability

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

A.6 Modularity

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

A.7 Framework Support

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

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

