OETCS/WP7/D7.3

openETCS

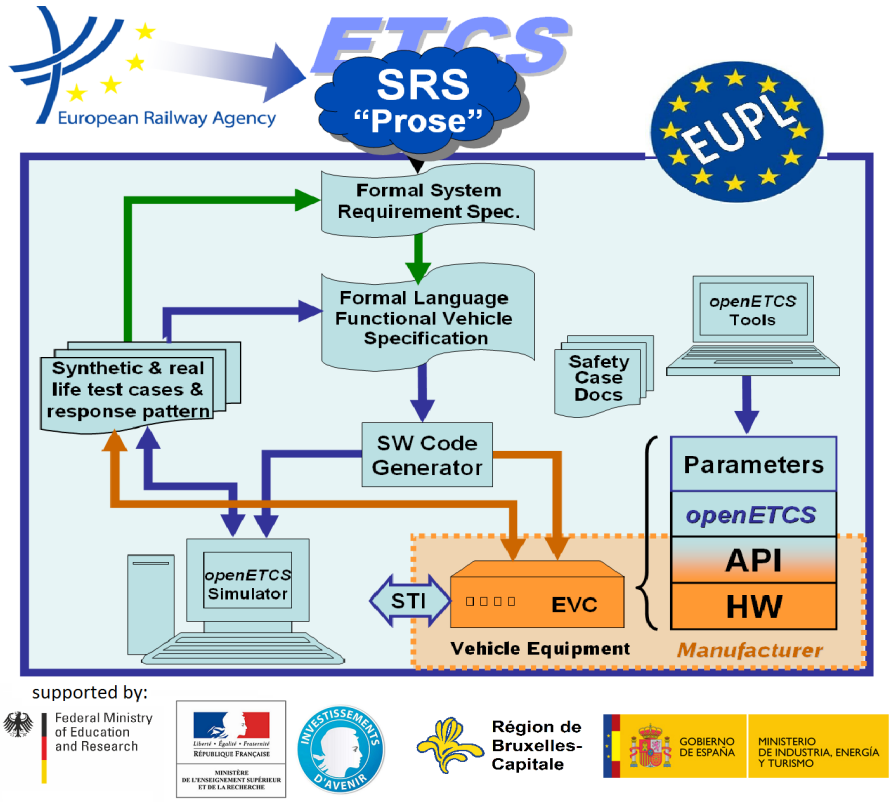
ITEA2 Project
Call 6 11025
2012 – 2015

Work-Package 7: “Tool chain”

Tool chain Qualification Process Description

Cecile Braunstein and Jan Peleska

January 2014



This page is intentionally left blank

Work-Package 7: “Tool chain”

OETCS/WP7/D7.3
January 2014

Tool chain Qualification Process Description

Cecile Braunstein and Jan Peleska
University Bremen

Qualification process description

Prepared for ITEA2 openETCS consortium
Europa

Abstract: This document presents different ideas of a tool chain qualification. It describes a process for the openETCS tool chain qualification.

Table of Contents

Document Information

Document information	
Work Package	WP7
Deliverable ID or doc. ref.	D7.3
Document title	Tool chain Qualification Process Description
Document version	01.00
Document authors (org.)	Cécile Braunstein and Jan Peleska (Uni.Bremen)

Review information	
Last version reviewed	
Main reviewers	

Approbation			
	Name	Role	Date
Written by	Cécile Braunstein	WP7-T7.3 Sub-Task Leader	12.01.2014
Approved by			

Document evolution			
Version	Date	Author(s)	Justification
00.00	12.01.2014	C. Braunstein	Document creation
01.00	19.01.2014	C. Braunstein	Jan Peleska suggestions

1 introduction to tool chain qualification

1.1 Tool Qualification

The CENELEC EN 50128 standard [11] defines the tool qualification as follows :
"The objective is to provide evidence that potential failures of tools do not adversely affect the integrated tool-set output in a safety related manner that is undetected by technical and/or organizational measures outside the tool. To this end, software tools are categorized into three classes namely, T1, T2 & T3 respectively."

We recall here the different class definitions:

- Tool class T1: No generated output can be used directly or indirectly to the executable code;
- Tool class T2: Verification tools, the tool may fail to detect errors or defects;
- Tool class T3: Generated output directly or indirectly as part of the executable code.

The deliverable D2.2 [8] summarizes the requirements for the tool needed by the different tool classes. The report highlights that the effort differs depending on the tool class. Furthermore, for the most critical class T3, the evidence should be provided that the output is conform to the specification or that *any failure in the output are detected*.

The standard defined how to classify each tool individually (see [6, 7] as an example). But dealing with a tool chain, integrated within a tool platform, implies extra effort to ensure that the tool integration does not introduce new errors. For example mechanism such as artifacts versioning, time-stamping operations, etc ... should also be considered when qualifying the tool chain.

In summary, the effort of qualification depends on the tool class and the tool error detection capabilities. To reduce the cost of the tool chain qualification and regarding the fact that our development imply regular releases, a systematic tool chain analysis approach has to be defined.

1.2 Tool chain qualification state of the art

Some recent works have been done in the field of tool chain qualification from a variety of projects. The next section summarizes the most significant ones.

1.2.1 Slotosh and al. (project RECOMP)

[9] describes a model-based approach to tool qualification to comply with DO-330 and integration into the Eclipse development environment. The authors claim that the benefits of their method are the following :

- Clarity : remove ambiguities;
- Re-usability and Transparency : check for reuse in different tool chain;
- Completeness: the model covers all parts of the development process and tis traceable;

- Automation: Some part of the process may be automated.

Their method is explained in detail in [10]: the tool chain analysis is based on a domain specific tool chain model they have defined. This model is used to represent the tool chain structure as well as the tool confidence. Their Goal is to deduce the tool confidence level and to expose specific qualification requirements, furthermore, their idea is not only to check tool by tool but to have a more holistic approach and makes use of rearrangement and/or the extension of the tool chain to avoid the certification of all tools. This allows them to reduce the qualification effort by focusing only on the critical tools and make use of already available information. Moreover, some inconsistencies check such as, missing description, unused artifacts ..., may be automatically done by using the tool chain model. Finally, they provide support to automatize the document production.

[12] apply their tool and methods an industrial use case to determine the potential errors in the tool-chain.

1.2.2 Asplund and al. (projects iFEST, MBAT)

The authors investigate the question if there exists part of the environment related to tool integration that may fall outside the tool qualification defined by the a norm (ISO 26262 here [2]). And if so, how tool integration is affected by ensuring functional safety. One conclusion is that the tool integration may lead to increase the qualification effort.

They also state that the standards (EN 50128, DO-178C and ISO26262) are not sufficient to check safety of a tool chain, but some part of a tool chain may be taken into account to mitigate the qualification effort. They highlight 9 safety issues caused by tool integration that also allow to be more exact when identifying software that have to be qualified for certification purpose.

They advocate that to deal with the qualification of tool integration within a tool chain a system approach should be taken, we should not thing about individual tools anymore. Their proposed method is a “System Approach” for tool chain qualification following these steps ;

1. Pre-Qualification of development tool (requirements tools,design tools ...): provided by the vendors.
2. Pre-qualification at the tool-chain level: based on step 1 and reference work-flows, defined where are the safety critical part.
3. Qualification of the tool-chain: check differences of step 2 and the actual deployed tool chain.
4. Qualification at the tool level: based on the actual environment when deploying the tool chain.

This approach leads them to separate the parts required to software tool qualification and to identify safety issues related to tool integration.

In [1], they explore the step 2): identifying the required safety goals due to tool integration and obtain a description of a reference work-flow and tool-chain with annotation about the mitigating effort. They proposes to use the TIL language, a domain specific language for tool chain model. The model of the tool chain is used to perform a risk analysis and to annotate parts that need mitigating effort for the safety issues due to tool integration.

1.2.3 Biehl and al. (projects CESAR, iFEST, MBAT)

Biehl proposed a Domain Specific Language named TIL for Generating Tool Integration Solutions [5]. A tool chain is described in terms of a number of "Tool Adapters" and the relation between them.

- ToolAdapters : exposes data and functionality of a tool
- Channels
  - ControlChannels describe service calls
  - DataChannel describe data exchanges
  - TraceChannel describe creation of a trace links
- Sequencer : describe sequential control flow (sequence of services)
- User : describe and limit the possible interaction
- Repository: provide storage and version Management of tool data

This DSL allows early analysis of the tool chain. It may generate part of tool adapter code based on the source and target meta-model.

More recently, Biehl and al. define a standard language for modeling development process defines by OMG 2008. The language has been used in [4, 3] together with the TIL language to tailor a tool chain following a process model. The goal is to be able to model both the development process and the set of tools used. A process is defined as follows:

- Process : several Activities
- Activity : set of linked Tasks, WorkProducts, Roles
- A Role can perform a Task
- A WorkProduct can be anaged by a Tool
- A Task can use a Tool

Using together the process development language and the tool chain language, in [3], the authors measure the alignment of a tool chain with a product development process. The method proceeds as follows:

1. Inputs :
   - formalized description of the tool chain design
   - description of the process including the set of tools and their capabilities
2. Initial verification graph
3. Automatic mapping links to the verification graph (acc. to mapping rules)
4. Apply alignment rule on the verification graph
5. Apply metrics to determine the degree of alignment btw the tool-chain and the process

The metrics and the misalignment list provide feedback to refine the tool-chain design.

2 OpenETCS tool chain qualification process

2.1 Tool chain Analysis

All the methods mentioned above start with a complete definition of the tool chain. In OpenETCS, the development of the tool chain follow an AGILE method, hence for each (major) release we have to deal with an incomplete tool chain. In addition to the methods of the previous section, we need a qualification process that can adapt to the development speed, deal with incomplete tool chain and can re-use qualification information.

Moreover, as stated by Asplund and al., the tool chain provides some mechanism that has to be also ensured, reducing thereby the effort for each tool. These safety goals are related to the tool integration. In our context, most of the tool integration is made by integrated tools into a tool platform. From the previous cited paper, the tool platform should ensure the following safety-goals

- Coherent Time Stamp Information: common time stamps on development artifacts.
- Notification: the user should be notified when artifacts changed.
- Data integrity: avoid use of obsolete artifacts, the data used reflects the current state.
- Data Mining : all data used by safety analysis should be available and be verifiable.

2.2 OpenETCS tool chain qualification process

The OpenETCS tool chain is described as a SysML activity diagram. This activity diagram grows according to the new feature request and the need of openETCS participants. Each feature of the tool chain is represented as an activity node, each artifacts by a data store. Each feature realizes at least one use case and is implemented by at least one tool. Note that in the tool platform environment tool may also be implemented as plug-ins. The diagram also represents the order between the different features, it also shows which tasks maybe done in parallel and which ones are dependent of other tasks.

To mitigate the qualification process, we will consider each feature and not each tool since the combination of tool may, for example ensure the error detection capability of a feature. Furthermore, the tool chain is a collection of feature and not tools, this differs from Asplund and al. in the sense that some of the tool integration mechanism Automated Transformation of Data are part of the feature and are not falling out of the scope of the qualification.

Due to our development process, a “pre-qualification” of tools should be made when integrating a tool.

Tool integration process for qualification

- Define name and version
- Describe use cases

- Provide input/output artifacts format (associated with the version)
- Integrate the tool in the SysML model
- Provide tool manual and other available documentation (associated with the version)
- Link with an issue tracker

One possible implementation is to represent all these in formations directly in the SysML model.

The qualification process

1. Feature Analysis
   - This step should assign a class to each feature based on the use cases.
   - Define the potential errors
   - Identify counter measure and/or error detection
   - For T3 tools 2 alternatives: certified compiler/generator or object code checker and/or exhaustive tester
2. Tool platform analysis
   - Provide evidence of the safety-goals mentioned in the previous sub-section
3. Tool chain Analysis
   - Defines the work-flow
   - Identify the “hot spots” of the tool chain
   - Rearrange the tool chain if possible
   - Find new measures when needed with combining tools (redundancy with orthogonal codes . . .)
4. Tool chain qualification verification
   - check consistency of tool version with manuals . . .
   - Generate table to check if all possible errors has a detection or a correction mechanism
   - Generate the qualification report

References

[1] Fredrik Asplund, Matthias Biehl, and Frédéric Loiret. Towards the automated qualification of tool chain design. In *Computer Safety, Reliability, and Security*, volume 7613 of *Lecture Notes in Computer Science*, page 392–399. Springer, 2012.

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

[3] Matthias Biehl. Early automated verification of tool chain design. *Computational Science and Its Applications–ICCSA \ldots*, page 40–50, 2012.

[4] Matthias Biehl and M Törngren. Constructing tool chains based on SPEM process models. In *The Seventh International Conference on Software Engineering Advances (ICSEA2012)*, page 267–273, 2012.

[5] Biehl, Matthias, El-Khoury, Jad, Loiret, Frédéric, and Törngren, Martin. A domain specific language for generating tool integration solutions. In *In 4th Workshop on Model-Driven Tool & Process Integration*, 2011.

[6] 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.

[7] Wen-Ling Huang, Jan Peleska, and Uwe Schulze. Test automation support. Delivrable D34.1, COMPASS, January 2013.

[8] Merlin Pokam and Norbert Schäfer. Report on CENELEC standards. Requirements D2.2, openETCS, April 2013.

[9] Oscar Slotosch. Model-based tool qualification : The roadmap of eclipse towards tool qualification. *Springer*, 2012.

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

[11] European Standard. *Railway applications-Communication,signalling and processing system- Software for railway control and protecton system*. CENELEC EN 50128. DIN, October 2011.

[12] Martin Wildmoser, Jan Philipps, and Oscar Slotosch. Determining potential errors in tool chains: strategies to reach tool confidence according to ISO 26262. In *Proceedings of the 31st international conference on Computer Safety, Reliability, and Security*, SAFECOMP'12, page 317–327, Berlin, Heidelberg, 2012. Springer-Verlag.

