OETCS/WP4/D4.1aV02

openETCS

ITEA2 Project
2012 – 2015

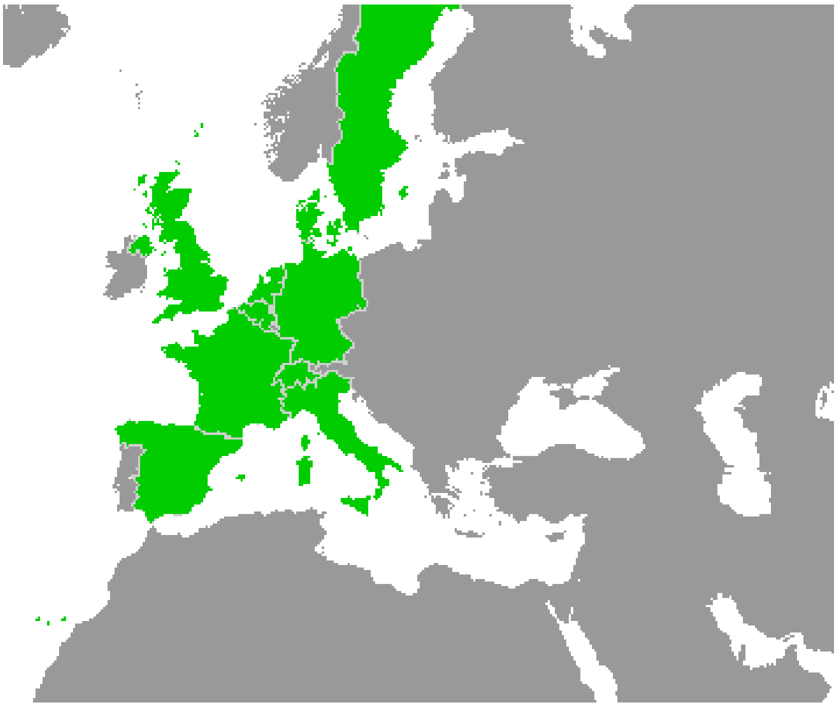
Work-Package 4: “Verification and Validation”

Preliminary Evaluation Criteria on Verification and Validation

Version O2

Hardi Hungar, Marc Behrens

May 2013



This page is intentionally left blank

Preliminary Evaluation Criteria on Verification and Validation

Version O2

Hardi Hungar, Marc Behrens

DLR
Germany

Description of work

Prepared for ITEA2 openETCS consortium
Europa

Abstract: Evaluation criteria for tools and methods to be selected for use in V&V activities are derived from a decription of their purpose (i.e., the activities to be performed with the help of the tools). This description is complemented by `WP41a-PreliminaryEvaluationCriteriaOnVAndVTables_V02_20130522.xml`.

Table of Contents

Figures and Tables **Figures**

Tables

1 Verification and Validation Activities

1.1 Definitions

Verification

Verification is an activity which has to be performed at each step of the design. It has to be verified that the design step achieved its goals. This consists at least of two parts:

- that the artifacts produced in the step are of the right type and contain allthe information they should. E.g., that the SSRS identifies all components addressed in SS 026, specifies their interfaces in sufficient detail and has allocated the functions to the components (this should just serve an example and is based on a guess what the SSRS should do)
- that the artifact correctly implements the input requirements of the design step. These typically include the main output artifacts of the previous step. "Correctly implements" includes requirement coverage (tracing). This can and should be supported by some tools. Adequacy of such tools depends on things like format compatibility, degree of automation, functionality (e.g., ability to handle m-to-n relations). Depending on the design step (and the nature of the artifacts) different forms of verification will complement requirement coverage, with different levels of support. The step from SS 026 to the SSRS will mainly consist of manual activities besides things like coverage checks. Verifying a formal (executable) model against the SSRS can be supported by animation or simulation to e.g. execute test cases which have been designed to check compliance with the SSRS. Even formal proof tools may be employed to check or establish properties. Model-to-code steps offer far more options (and needs) for tool support. And tools or tool sets for unit test will support dynamic testing for requirement or code coverage. This may include test generation, test execution with report generation, test result evaluation and so on. Also, code generator verification (or qualification) may play a role, here. Integration steps mandate still other testing (or verification) techniques.

Summarizing, one may say that verification subsumes highly diverse activities, and may be realized in very many different forms.

Validation

Validation is name for the activity by which the compliance of the end result with the initial requirements is shown. In the case of openETCS, this means that the demonstrator (or parts of it) are checked against the SS 026 or one of its close descendants (i.e., SSRS). This will consist of testing the equipment according to a test plan derived form the requirements and detailed into concrete test cases at some later stage. Tool support for validation will thus mainly concern test execution and evaluation, perhaps supplemented by test derivation or test management. Ambitous techniques like formal proof are most likely not applicable here.

Thus, the tool support for validation will not differ substantially from that for similar verification activities.

One might also consider "early" validation activities, e.g. "validating" an executable model against requirements from the SS 026. These are not mandated by the standards and can per se not replace design step verification. They may nevertheless be worthwhile as means for early defect detection.

Further (mostly complementary) information on V&V can be found in the report on the CENELEC standards (D2.2).

2 Evaluation Criteria

2.1 An Incomplete List of V&V Activities

The following activities, which can all be performed or supported by tools, might be relevant to openETCS.

Tracing: Relating requirement items to implementing items

Animation: Executing a model

Simulation: code/model, perhaps with some environment representation

Formal proof of properties: e.g. establishing some invariant

Proof checking: Verifying that a proof is correct (independently)

Model checking: Deciding properties of formal objects (specs, models, code)

Test case generation: From a formal object

Test sequence generation: Arranging test cases in a suitable way

Test execution: with subactivities test evaluation, report generation, perhaps test management, regression testing

Table 1 lists some verification techniques which may be used in openETCS to perform mandatory (according to the EN 50128) verification steps. It is taken from the accompanying document `WP41a-PreliminaryEvaluationCriteriaOnVAndVTables_V02_20130522.xml`.
The following abbreviations for design artifact are used in the table.

`C`: code.

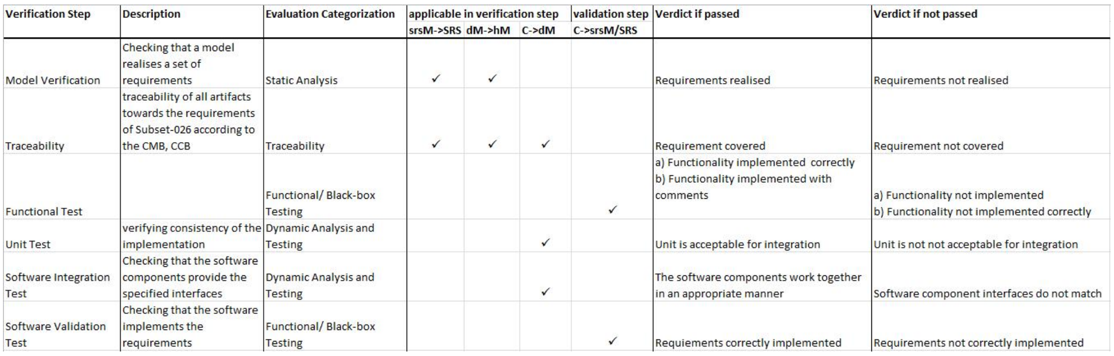
`dM`: A detailed model (from which code is derived or generated).

`hM`: A higher-level model (e.g., design specification model.

`srsM`: A model of the requirements (SRS).

`SRS`: The system requirements specification.

Table 1. Verification Steps and Techniques in openETCS



Verification Step	Description	Evaluation Categorization	applicable in verification step			validation step	Verdict if passed	Verdict if not passed
			srsM->SRS	dM->hM	C->dM	C->srsM/SRS		
Model Verification	Checking that a model realises a set of requirements	Static Analysis	✓	✓			Requirements realised	Requirements not realised
Traceability	traceability of all artifacts towards the requirements of Subset-026 according to the CMB, CCB	Traceability	✓	✓	✓		Requirement covered	Requirement not covered
Functional Test		Functional/ Black-box Testing				✓	a) Functionality implemented correctly b) Functionality implemented with comments	a) Functionality not implemented b) Functionality not implemented correctly
Unit Test	verifying consistency of the implementation	Dynamic Analysis and Testing			✓		Unit is acceptable for integration	Unit is not not acceptable for integration
Software Integration Test	Checking that the software components provide the specified interfaces	Dynamic Analysis and Testing			✓		The software components work together in an appropriate manner	Software component interfaces do not match
Software Validation Test	Checking that the software implements the requirements	Functional/ Black-box Testing				✓	Requiements correctly implemented	Requirements not correctly implemented

2.2 An Incomplete List of V&V Tool Evaluation Criteria

As broad as the range of V&V activities and tools supporting them is the set of evaluation criteria:

Effectiveness: A tool must be able to perform a useful function

Efficiency: Very important for automatic routines like test case generation, model checking: They should not take forever or use infinite memory

Vertical workflow integration: input and output formats should match along a chain of dependent steps. Very specific input requirements can make a tool useless. More general, we should strive for a complete, rather well integrated tool chain

Horizontal workflow integration: Test management and test execution should go hand in hand.

Qualifiabilty: T2 or T3 qualification is required for some usages. Depends also on the process, as the failure of tool not being qualifiable in itself can be remedied by introducing a complemnting tool the checks the first's output (Example: Complementing code generation by verifying the equivalence of input and output)

FLOSS: of course (openETCS)

3 Summary

This document lists general criteria for the evaluation methods and tools for verification and validation within openETCS. Detailing the criteria needs more information on the design steps, which artifacts are produced by them and by which methods. Concrete criteria are at least as diverse as the different verification steps, and these depend highly on the objects and how they are produced.