OETCS/WP4/D4.1V03.01

openETCS

ITEA2
INFORMATION TECHNOLOGY FOR EUROPEAN ADVANCEMENT

ITEA2 Project
Call 6 11025
2012 – 2015

Work Package 4: “Validation & Verification Strategy”

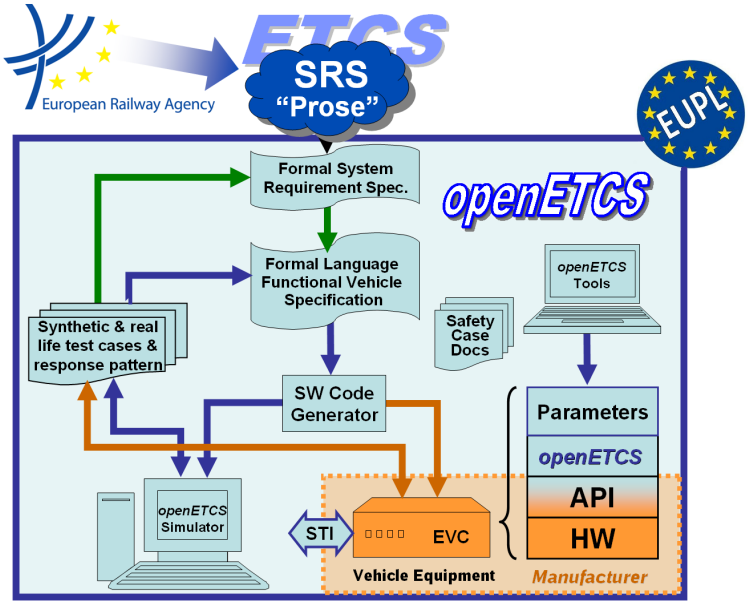
openETCS Validation & Verification Plan

Version 03.01

Hardi Hungar (Ed.)

13 Nov, 2015

Contributions by:
Frederic Badeau (Systerel), Marc Behrens (DLR),
Cecile Braunstein (U Bremen), Mirko Caspar (DLR),
Cyril Cornu (All4Tec),
Christophe Gaston (CEA), Jens Gerlach (Fraunhofer),
Ainhoa Gracia (SQS), Hardi Hungar (DLR),
Stephan Jagusch (AEbt), Alexander Nitsch (U Rostock),
Jan Peleska (U Bremen), Marielle Petit-Doche (Systerel),
Virgile Prevosto (CEA), Stefan Rieger (TWT),
Izaskun de la Torre (SQS), Jan Welte(TU-BS)



Funded by:

Federal Ministry of Education and Research
Région de Bruxelles-Capitale
GOBIERNO DE ESPAÑA
MINISTERIO DE INDUSTRIA, ENERGÍA Y TURISMO

This page is intentionally left blank

Work Package 4: "Validation & Verification Strategy" **OETCS/WP4/D4.1V03.01**
13 Nov, 2015

openETCS Validation & Verification Plan

Version 03.01

Document approbation

Lead author:	Technical assessor:	Quality assessor:	Project lead:
location / date	location / date	location / date	location / date
signature Hardi Hungar (DLR)	signature Marc Behrens (DLR)	signature Jens Gerlach (Fraunhofer FOKUS)	signature Klaus-Rüdiger Hase (DB Netz)

Hardi Hungar (Ed.)
Contributions by:
Frederic Badeau (Systerel), Marc Behrens (DLR),
Cecile Braunstein (U Bremen), Mirko Caspar (DLR),
Cyril Cornu (All4Tec),
Christophe Gaston (CEA), Jens Gerlach (Fraunhofer),
Ainhoa Gracia (SQS), Hardi Hungar (DLR),
Stephan Jagusch (AEbt), Alexander Nitsch (U Rostock),
Jan Peleska (U Bremen), Marielle Petit-Doche (Systerel),
Virgile Prevosto (CEA), Stefan Rieger (TWT),
Izaskun de la Torre (SQS), Jan Welte(TU-BS)

DLR
Lilienthalplatz 7
38108 Brunswick, Germany
eMail:hardi.hungar@dlr.de

Deliverable

Prepared for openETCS@ITEA2 Project

Abstract: This document describes strategy and plan of the verification and validation in the development of the software of the EVC (European Vital Computer) in the openETCS approach. It revises the previous versions (V01, V01.01) of this document. This document refers to the process for openETCS as defined in [1]. It comprises the current versions of the artifacts "0-03 Verification Plan" and "0-04 Validation Plan" as defined there.

The structure of the document follows the distinction from [1] between the current *openETCS project*, funded as part of the EUREKA cluster programme ITEA 2, and the *openETCS activity* as a whole, which encompasses the project.

In its three main parts, the current document addresses:

- verification and validation for the full development (openETCS activity)
- verification and validation for the current ITEA 2 project
- details about verification & validation tools and methods

The current document shall provide an important part of the basis for deliverable D4.4, the Final Report of the project on Verification & Validation.

Some of the revision work needed to be done for this version is indicated in the form of comments like this one. Most of the comments are missing themselves, though.

Table of Contents

Figures and Tables v

Document Control vi

I General Definitions 1

1 Purpose and Structure of the Document 2

2 Definitions 3

2.1 Verification 3
2.2 Validation 3
2.3 Review 4

II Verification & Validation for a Full Development 5

3 Verification & Validation in the openETCS Process 6

4 Verification & Validation Strategy for a Full Development 7

5 Verification Plan for a Full Development 8

5.0 Template for Describing a Verification or Validation Activity 8
5.1 Verification of Planning 8
5.1.1 Verification of the Project Plan 8
5.1.2 Verification of the Quality Assurance Plan 9
5.1.3 Verification of the Configuration Management Plan 9
5.1.4 Verification of the Verification Plan 9
5.1.5 Verification of the Validation Plan 9
5.1.6 Planning Verification Report 9
5.2 Verification of the System Design 9
5.3 Verification of the Sub-System Architecture Design 10
5.3.1 Verification of the Sub-System Architecture Design 10
5.3.2 Verification of the Acceptance Plan 10
5.3.3 Sub-System Architecture Design Verification Report 10
5.4 Verification of the SW Specification 10
5.5 Verification of the SW Design 10
5.6 Verification of the SW Component Implementation and Test 10
5.7 Verification of the SW Integration 10
5.8 Verification of the SW Validation 10

6 Validation Plan for a Full Development 11

III Verification & Validation Plan for the Project openETCS 12

7 Verification & Validation Strategy for the Project openETCS 13

8 Verification Plan for the Project openETCS 14

8.0 Template for Describing a Verification Activity of the Project 14

9 Validation Plan for the Project openETCS 15

IV Methods and Tools for Verification and Validation 16

References 17

Figures and Tables

Figures

Tables

Document Control

Document information	
Work Package	WP4
Deliverable ID or doc. ref.	D4.1
Document title	openETCS Validation & Verification Plan
Document version	03.01
Document authors (org.)	Frederic Badeau (Systerel), Marc Behrens (DLR), Cecile Braunstein (U Bremen), Mirko Caspar (DLR), Cyril Cornu (All4Tec), Christophe Gaston (CEA), Jens Gerlach (Fraunhofer), Ainhoa Gracia (SQS), Hardi Hungar (DLR), Stephan Jagusch (AEbt), Alexander Nitsch (U Rostock), Jan Peleska (U Bremen), Marielle Petit-Doche (Systerel), Virgile Prevosto (CEA), Stefan Rieger (TWT), Izaskun de la Torre (SQS), Jan Welte(TU-BS)

Review information	
Last version reviewed	–
Main reviewers	–

Approbation			
	Name	Role	Date
Written by	Hardi Hungar	WP4-T4.1 Task Leader	November 2015
Approved by	Marc Behrens	WP4 Leader	*tbd*

Document evolution			
Version	Date	Author(s)	Comment
03.01	13/11/2015	H. Hungar	Revision of document structure and content (partially) based on V01.01 and on D4.2.1, D4.2.2
03.02			
03.03			
03.	dd.mm.2015	M. Behrens	Review and approval

Part I

General Definitions

1 Purpose and Structure of the Document

This document describes strategy and plan of the verification and validation in openETCS. It revises the previous versions (V01, V01.01) of this document. This document refers to the process for openETCS as defined in [1]. It comprises the current versions of the artifacts "0-03 Verification Plan" and "0-04 Validation Plan" as defined there.

The structure of the document follows the distinction from [1] between the current *openETCS project*, funded as part of the EUREKA cluster programme ITEA 2, and the *openETCS activity* as a whole, which encompasses the project.

1. The first part defines (in this section) the role and purpose of the document. It provides basic definitions of verification and validation and provides a generic description of how to perform them.
2. The second part defines verification and validation for the full development. In its final version, it shall be a CENELEC-compliant plan for verification and validation of the openETCS EVC software.
3. The third part plans those activities which will actually be performed within the current project. These are related to the overall plan from the second part. They focus on the particular instantiation of the development in the project. They define concrete verification activities for the artifacts which are produced.
4. A fourth part collects descriptions of methods and tools. Most of these are already available from project partners or third parties. Some of them are subject to adaptations or even further development within openETCS. The second and third part refer to relevant methods and tools from the fourth part—ones which are used or could be used for verification or validation. Vice versa, the descriptions specify for which activity they can be used.

In both the second and third part, the verification & validation strategy, the verification plan and the validation plan are defined. Verification and validation share some of their methods and tools, and in some case are applied to the same design artifacts. Therefore, the plans for both are included in this document. Nevertheless, these activities are intended to be and remain independent.

Verification and validation play an important role in the safety case. This document identifies the V&V activities which do contribute and refers to the safety plan for further details on the additional requirements to be met and a precise statement of what has to be established.

2 Definitions

2.1 Verification

According to [2, 3.1.48], verification is an activity to check whether the output of a development phase meets the requirements. This concerns the following aspects.

Formalities: [2, 5.3.2.7 to 10]

1. The unambiguous identifiability of the artifacts which make up the objects of verification, their versions and their relationship with other artifacts by the consistent use of unique reference numbers.
2. Consistency in the usage of terms, names, descriptions
3. Formal completeness in addressing all applicable requirements laid down in the process plan.

Traceability: [2, 6.5.4.14 to 17] Most of the artifacts subjected to verification must provide detailed tracing information which establishes a relation between the constituting items of each object of verification and other artifacts, in particular the input artifacts of the design step whose output is verified.

Completeness, Correctness and Consistency: These properties refer to the specific content of the artifact.

The openETCS process [1] requires verification to be done in each of the phases. Verification has to check aspects of formal nature and those that refer to the content of the artifacts.

A typical example of verification is the check of a design refinement. The refined design must cover all required aspects, refine all requirements coming from the previous design step, and provide adequate tracing information. The requirements must all be correctly refined, and the refined design must in itself be consistent. Though correctness and consistently are not independent, it is usually beneficial to address both aspects explicitly. Furthermore, the refining artifacts must be readable and clearly structured.

Tracing information must be provided by the Quality Assurance Plan (0-01), the Verification Plan (0-03), the Validation Plan (0-04), the SW Requirement Specification (3-16), the Overall SW Test Specification (3-17), the SW Architecture and Design Specification (4-19), the SW Interface Specification (4-20), the SW Integration Test Specification (4-21), the SW Component Test Specification (4-22), the SW Components (5-24), the SW Component Test Report (5-25), the SW Integration Test Report (6-27), the Overall SW Test Report (7-29), the Validation Report (7-30).

2.2 Validation

Validation is name for the activity by which the compliance of an artifact with the user requirements is checked. Here, this means that the developed SW of the EVC is fit for its purpose: correct, safe, operational and fills its role in the EVC architecture.

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 neither replace verification nor validation steps. They may be worthwhile as means for early defect detection, and may also be integrated into verification activities, but they are not made parts of the current verification and validation plans.

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

2.3 Review

Most verification and validation activities consist in reviewing artifacts produced in the development process.

A *review* is

- a systematic analysis
- of a document or set of documents, the object of the review,
- performed by suitably trained personnel
- to determine the satisfaction of a specified set of properties
- potentially taking into account evidential material
- producing a documentation in a defined, structured format.

The *object of the review* may be text or structured text, can contain graphics, include formal notations of logical or mathematical nature, may be formal or semi-formal descriptions, programs or engineering descriptions.

The *documentation* classifies results as positive, negative or inconclusive with detailed references which items of the objects of the review and the properties to be checked these verdicts concern. Verdicts may be required to be substantiated by explanations.

Part II

Verification & Validation for a Full Development

3 Verification & Validation in the openETCS Process

4 Verification & Validation Strategy for a Full Development

5 Verification Plan for a Full Development

Verification has to be done in each of the phases of the development. This chapter is structured according to the verification reports to be produced.

5.0 Template for Describing a Verification or Validation Activity

General description of the activity	
Verification object	
Responsible role	
Reference material	
Objective	
Evidential material	
Method(s)	
Documentation	

Detailed description of the objective	
Formalities	
Traceability	
Completeness	
Correctness	
Consistency	

Detailed description of the documentation [Option 1]	
Template	

Detailed description of the documentation [Option 2]	
Managerial items	
[Output item/section *nn*]	

Detailed description of the activity	
Activity steps	
Step *nn*	

5.1 Verification of Planning

These verifications are done by reviewing the documents. The results are collected in the Planning Verification Report (0-05).

5.1.1 Verification of the Project Plan

General description of the activity	
Verification object	Project Plan (0-00)
Responsible role	VER
Reference material	EN 50126, EN 50128, FPP, openETCS Process (D2.3a)
Objective	Establish that the project plan defines a viable management structure, where all activities are assigned properly and adequately.
Evidential material	–
Method	Review
Documentation	Text document

Detailed description of the objective	
Formalities	*tbd*
Traceability	–
Completeness	*tbd*
Correctness	*tbd*
Consistency	*tbd*

Detailed description of the documentation	
Managerial items	**Input:** [Project plan version], **Output:** [Project plan verification report version], **Responsible person:** [person of role VER], **Contributors:** [list], **Date:** [date]
[Output item/section *nn*]	

Detailed description of the activity	
Activity steps	*tbd*
Step *nn*	*tbd*

5.1.2 Verification of the Quality Assurance Plan

5.1.3 Verification of the Configuration Management Plan

5.1.4 Verification of the Verification Plan

5.1.5 Verification of the Validation Plan

5.1.6 Planning Verification Report

Sec. 5.1.6 gives an overview of the components of the Planning Verification Report. Its main parts are described in the subsections of Sec. 5.1 preceding this subsection (Sec. 5.1.6).

5.2 Verification of the System Design

5.3 Verification of the Sub-System Architecture Design

5.3.1 Verification of the Sub-System Architecture Design

5.3.2 Verification of the Acceptance Plan

5.3.3 Sub-System Architecture Design Verification Report

5.4 Verification of the SW Specification

5.5 Verification of the SW Design

5.6 Verification of the SW Component Implementation and Test

5.7 Verification of the SW Integration

5.8 Verification of the SW Validation

6 Validation Plan for a Full Development

Part III

Verification & Validation Plan for the Project openETCS

7 Verification & Validation Strategy for the Project openETCS

8 Verification Plan for the Project openETCS

This chapter (Sec. 8) should have been based on a tailoring of the process for the full development [1] to the project activities. In absence of such a tailoring, which would relate the project activities and their outcome appropriately the full development, this is done here for each planned verification activity. I.e., the artifacts of the project which are subjected to verification are related to the artifacts defined in [1]. And for each verification activity of the project, the corresponding verification activity (or activities) from the Verification Plan for the full development (Sec. 5) is adapted here to the scope and form of the respective project result.

8.0 Template for Describing a Verification Activity of the Project

The template from Sec. 5.0 is to be extended by a description of the verification object in relation to the corresponding artifact(s) in the full development.

9 Validation Plan for the Project openETCS

Part IV

Methods and Tools for Verification and Validation

References

[1] Hardi Hungar. Definition of the openETCS development process. Technical Report D2.3a-V02, OpenETCS, September 2015.

[2] Railway applications – Communication, signalling and processing systems – software for railway control and protection systems. Norm EN 50128:2011, CENELEC, Brussels, Belgium, 2011.

[3] Edmund M. Clarke and Bernd-Holger Schlingloff. Model checking. In Robinson and Voronkov [75], pages 1637–1790.

[4] Dillon Pariente and Emmanuel Ledinot. *Formal Verification of Industrial C Code using Frama-C: A Case Study*, pages 205–219. Volume 6528 of Beckert and Marché [72], 2010.

[5] J. Peleska, J. Feuser, and A. E. Haxthausen. *Railway Safety, Reliability and Security: Technologies and Systems Engineering*, chapter The Model-Driven openETCS Paradigm for Secure, Safe and Certifiable Train Control Systems, pages 22–52. In Flammini [74], 2012.

[6] Jan Peleska, Elena Vorobev, and Florian Lapschies. Automated test case generation with smt-solving and abstract interpretation. In Bobaru et al. [73], pages 298–312.

[7] Jean-Raymond Abrial. *Modeling in Event-B - System and Software Engineering*. Cambridge University Press, 2010.

[8] Jean-Raymond Abrial. *The B-book - assigning programs to meanings*. Cambridge University Press, 2005.

[9] Rajeev Alur and David L. Dill. A theory of timed automata. *Theor. Comput. Sci.*, 126(2):183–235, April 1994.

[10] B. Bannour. *Symbolic analysis of scenario based timed models for component based systems: Compositionality results for testing*. PhD thesis, CEA LIST / École Centrale Paris, 2012. `http://www.cti.ecp.fr/~bannourb/PhDthesis.pdf`.

[11] B. Bannour, J.P. Escobedo, C. Gaston, and P. Le Gall. Off-line test case generation for timed symbolic model-based conformance testing. In *Proceedings of the 23rd International Conference on Testing Software and Systems (ICTSS)*. Springer LNCS, 2012.

[12] B. Bannour, C. Gaston, and D. Servat. Eliciting unitary constraints from timed Sequence Diagram with symbolic techniques: application to testing. In *Proceedings of the 18th Asian-Pacific Software Engineering Conference (APSEC)*. IEEE Computer Society, 2011.

[13] Sylvain Baro and Jan Welte. Requirements for openETCS. Technical Report D2.6, OpenETCS, June 2013.

[14] Patrick Baudin, Loïc Correnson, and Zaynah Dargaye. *WP Plugin*. CEA LIST, 0.7 for fluorine-20130601 edition, 2013. available at `http://frama-c.com/download/frama-c-wp-manual.pdf`.

[15] Patrick Baudin, Jean-Christophe Filliâtre, Claude Marché, Benjamin Monate, Yannick Moy, and Virgile Prevosto. *ACSL: ANSI/ISO C Specification Language*, 1.7 edition, April 2013. available at `http://frama-c.com/download/acsl.pdf`.

[16] M. Behrens, H. Hungar, A. Cavalli, J. Gerlach, H. Manz, and C. Cornu. openETCS validation and verification strategy work package: Description of work, May 2013.

[17] Gerd Behrmann, Alexandre David, and Kim G. Larsen. A tutorial on UPPAAL. In Marco Bernardo and Flavio Corradini, editors, *Formal Methods for the Design of Real-Time Systems: 4th International School on Formal Methods for the Design of Computer, Communication, and Software Systems, SFM-RT 2004*, number 3185 in LNCS, pages 200–236. Springer–Verlag, September 2004.

[18] Armin Biere, Alessandro Cimatti, Edmund Clarke, and Yunshan Zhu. Symbolic model checking without BDDs. In *Proceedings of the 5th International Conference on Tools and Algorithms for Construction and Analysis of Systems*, TACAS'99, pages 193—207. Springer-Verlag, 1999.

[19] A. Bouajjani, C. Dragoi, C. Enea, and M. Sighireanu. On inter-procedural analysis of programs with lists and data. In *Proceedings of Programming Languages Design and Implementation (PLDI)*, 2011.

[20] A Bradley. SAT-based model checking without unrolling. *\ldots , Model Checking, and Abstract Interpretation*, 2011.

[21] 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.

[22] CENELEC, European Committee for Electrotechnical Standardization. EN 50128: Railway applications - Communication, signalling and processing systems - Software for railway control and protection systems, June 2011.

[23] L.-A. Clarke. A system to generate test data and symbolically execute programs. *IEEE Transactions on software engineering*, 2(3):215–222, September 1976.

[24] Comission Decision. CCS TSI for HS and CR transeuropean rail. Technical Report 2012/88/EU, EU, January 2012.

[25] P. Cousot and R. Cousot. Static determination of dynamic properties of programs. In *Proc. 2nd Int. Symp. on Programming*, pages 106–130, Paris, 1976. Dunot.

[26] Pascal Cuoq, Florent Kirchner, Nikolai Kosmatov, Virgile Prevosto, Julien Signoles, and Boris Yakobowski. Frama-C: a Software Analysis Perspective. In *Proceedings of Software Engineering and Formal Methods (SEFM)*, volume 7504 of *LNCS*, pages 233–247. Springer, 2012.

[27] Pascal Cuoq, Boris Yakobowski, and Virgile Prevosto. *Frama-C's value analysis plug-in*. CEA LIST, fluorine-20130601 edition, June 2013. available at `http://frama-c.com/download/frama-c-value-analysis.pdf`.

[28] Alexandre David, M.Oliver Möller, and Wang Yi. Formal verification of uml statecharts with real-time extensions. In Ralf-Detlef Kutsche and Herbert Weber, editors, *Fundamental Approaches to Software Engineering*, volume 2306 of *Lecture Notes in Computer Science*, pages 218–232. Springer, 2002.

[29] Karsten Diethers and Michaela Huhn. Vooduu: Verification of object-oriented designs using uppaal. In Kurt Jensen and Andreas Podelski, editors, *Tools and Algorithms for the Construction and Analysis of Systems*, volume 2988 of *Lecture Notes in Computer Science*, pages 139–143. Springer, 2004.

[30] J.P. Escobedo, C. Gaston, and P. Le Gall. Timed Conformance Testing for Orchestrated Service Discovery. In *Proceedings of the 8th International Symposium on Formal Aspects of Component Software (FACS)*. Springer LNCS, 2011.

[31] M. E. Fagan. Design and code inspections to reduce errors in program development. *IBM Syst. J.*, 38(2-3):258–287, 1999. Reprint from 1976.

[32] A. Faivre, C. Gaston, and P. Le Gall. Symbolic Model Based Testing for Component Oriented Systems. In *Proceedings of the 19th International Conference on Testing Communicating Systems (TestCom/FATES)*. Springer LNCS, 2007.

[33] Harald Fecher, Jens Schönborn, Marcel Kyas, and Willem-Paul de Roever. 29 new unclarities in the semantics of UML 2.0 state machines. In Kung-Kiu Lau and Richard Banach, editors, *Formal Methods and Software Engineering*, number 3785 in Lecture Notes in Computer Science, pages 52–65. Springer Berlin Heidelberg, January 2005.

[34] C. Gaston, M. Aiguier, and P. Le Gall. Algebraic Treatment of Feature-oriented Systems. In *Language Constructs for Describing Features*. Springer LNCS, 2000.

[35] C. Gaston, P. Le Gall, N. Rapin, and A. Touil. Symbolic execution techniques for test purpose definition. In *Proceedings of the 18th International Conference on Testing Communicating Systems (TestCom)*. Springer LNCS, 2006.

[36] Jens Gerlach, Virgile Prevosto, Jochen Burghardt, Kerstin Hartig, Kim Völlinger, and Hans Pohl. Formal Specification and Automated Verification of Railway Software with Frama-C. In *IEEE International Conference on Industrial Informatics (INDIN)*. IEEE Xplore, July 2013.

[37] C.A.R. Hoare and Niklaus Wirth. An axiomatic definition of the programming language Pascal. *Acta Informatica*, 2:335 – 355, 1973.

[38] Klaus-Rüdiger Hase and Peter Mahlmann. Project outline full project proposal annex openETCS. Technical Report v4.0, openETCS, 2014.

[39] Anne Elisabeth Haxthausen, Jan Peleska, and Sebastian Kinder. A formal approach for the construction and verification of railway control systems. *Formal Asp. Comput.*, 23(2):191–219, 2011.

[40] C. A. R. Hoare. An axiomatic basis for computer programming. *Communications of the ACM*, 12(10):576–580 and 583, October 1969.

[41] Hardi Hungar. openETCS valdiation & verification plan. Technical Report D4.1.1.02, openETCS, July 2014.

[42] Hardi Hungar. Report on v&v plan and methodology. Technical Report D4.1, openETCS, July 2013.

[43] Michael Jastram and Marielle Petit-Doche. Report on the final choice of the primary toolchain. Technical Report 02, openETCS, November 2014.

[44] Rick Kazman, Gregory Abowd, Len Bass, and Paul Clements. Scenario-based analysis of software architecture. *IEEE Softw.*, 13(6):47–55, 1996.

[45] J.-C. King. A new approach to program testing. *Proceedings of the international conference on Reliable software, Los Angeles, California*, 21-23:228–233, April 1975.

[46] Alexander Knapp, Stephan Merz, and Christopher Rauh. Model checking - timed uml state machines and collaborations. In *Proceedings of the 7th International Symposium on Formal Techniques in Real-Time and Fault-Tolerant Systems: Co-sponsored by IFIP WG 2.2*, FTRTFT '02, pages 395–416, London, UK, UK, 2002. Springer.

[47] M. Krichen and S. Tripakis. Black-box time systems. In *Proc. of Int. SPIN Workshop Model Checking of Software*. Springer, 2004.

[48] Helge Löding and Jan Peleska. Timed moore automata: test data generation and model checking. In *Proc. 3rd International Conference on Software Testing, Verification and Validation (ICST'10)*. IEEE Computer Society, 2010.

[49] K.L. McMillan. Interpolation and sat-based model checking. In Jr. Hunt, WarrenA. and Fabio Somenzi, editors, *Computer Aided Verification*, volume 2725 of *Lecture Notes in Computer Science*, pages 1–13. Springer Berlin Heidelberg, 2003.

[50] Jan Peleska, Artur Honisch, Florian Lapschies, Helge Löding, Hermann Schmid, Peer Smuda, Elena Vorobev, and Cornelia Zahlten. A real-world benchmark model for testing concurrent real-time systems in the automotive domain. In Burkhart Wolff and Fatiha Zaidi, editors, *Testing Software and Systems. Proceedings of the 23rd IFIP WG 6.1 International Conference, ICTSS 2011*, volume 7019 of *LNCS*, pages 146–161, Heidelberg Dordrecht London New York, November 2011. IFIP WG 6.1, Springer.

[51] 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.

[52] Marielle Petit-Doche and Matthias Güdemann. openETCS process. Technical Report D2.3, OpenETCS, June 2013.

[53] A. Pnueli, O. Shtrichman, and M. Siegel. The code validation tool CVT: Automatic verification of a compilation process. *International Journal on Software Tools for Technology Transfer*, 2(2):192–201, 1998.

[54] Merlin Pokam and Norbert Schäfer. Report on CENELEC standards. Technical Report D2.2, OpenETCS, June 2013.

[55] C.-V. Ramamoorthy, S.-F. Ho, and W.-T. Chen. On the automated generation of program test data. *IEEE Transactions on software engineering*, 2(4):293–300, September 1976.

[56] N. Rapin, C. Gaston, A. Lapitre, and J.-P. Gallois. Behavioral Unfolding of Formal Specifications Based on Communicating automata. In *Proceedings of the 1st workshop on Automated Technology for Verification and Analysis (ATVA)*, 2003.

[57] RTCA SC-205. *Software Considerations in Airborne Systems and Equipment Certification (DO-178C)*. Radio Technical Commission for Aeronautics (RTCA Inc.), Washington/DC, Dec 2011.

[58] RTCA SC-205. *Formal Methods Supplement to DO-178C and DO-278A (DO-333)*. Radio Technical Commission for Aeronautics (RTCA Inc.), Washington/DC, Dec 2011.

[59] Carnegie Mellon University Software Engineering Institute. Architecture tradeoff analysis method. http://www.sei.cmu.edu/architecture/tools/evaluate/atam.cfm.

[60] Jean Souyris and David Delmas. Experimental Assessment of Astrée on Safety-Critical Avionics Software. In *Proc. Int. Conf. Computer Safety, Reliability, and Security, SAFECOMP 2007*, volume 4680 of *LNCS*. Springer, September 2007.

[61] Nicolas Stouls and Virgile Prevosto. *Aoraï Plugin Tutorial*. INSA Lyon and CEA LIST, 2013. Available at `http://frama-c.com/download/frama-c-aorai-manual.pdf`.

[62] I. de la Torre. Project quality assurance plan. openETCS Deliverables D1.3.1.

[63] J. Tretmans. Conformance Testing with Labelled Transition Systems: Implementation Relations and Test Generation. *Computer Networks and ISDN Systems*, 29:49–79, 1996.

[64] UNISIG. SUBSET-034 3.0.0 - Train interface FIS. Technical Report 3.0.0, ERA.

[65] UNISIG. SUBSET-076 - Test related ERTMS documentation (this version is related to version 2.3.y of SUBSET-026). Technical Report 2.3.y, ERA.

[66] UNISIG. SUBSET-088 2.3.0 - ETCS Application Levels 1 & 2 - Safety Analysis. Technical Report 2.3.0, ERA.

[67] UNISIG. SUBSET-091 3.2.0 - Safety Requirements for the Technical Interoperability of ETCS in Levels 1 & 2. Technical Report 3.2.0, ERA.

[68] UNISIG. SUBSET-026 - System Requirements Specification. Technical Report 3.3.0, ERA, March 2012.

[69] Verified Systems International GmbH. Verified :: Products. http://www.verified.de/en/products.

[70] Jan Welte. Safety plan. Technical Report O 4.4.1, openETCS, October 2013.

[71] Jan Welte and Hansjörg Manz. Report on existing methodologies. Technical Report D2.1, OpenETCS, June 2013.

[72] Bernhard Beckert and Claude Marché, editors. *Formal Verification of Object-Oriented Software - International Conference, FoVeOOS 2010, Paris, France, June 28-30, 2010, Revised Selected Papers*, volume 6528 of *Lecture Notes in Computer Science*. Springer, 2010.

[73] Mihaela Gheorghiu Bobaru, Klaus Havelund, Gerard J. Holzmann, and Rajeev Joshi, editors. *NASA Formal Methods - Third International Symposium, NFM 2011, Pasadena, CA, USA, April 18-20, 2011. Proceedings*, volume 6617 of *Lecture Notes in Computer Science*. Springer, 2011.

[74] Francesco Flammini, editor. *Railway Safety, Reliability and Security: Technologies and Systems Engineering*. Information Science Reference, 2012.

[75] Alan Robinson and Andrei Voronkov, editors. *Handbook of Automated Reasoning*. Elsevier, 2001.

[76] Theorem Prover. `http://alt-ergo.lri.fr`.

[77] Atelier B. `http://www.atelierb.eu/`.

[78] Coq Prover. `http://coq.inria.fr`.

[79] CPN Tools. `http://cpn-tools.org`.

[80] Frama-C: Source Code Analysis Suite. `http://frama-c.com/`.

[81] List of free formal verification tools. `http://gulliver.eu.org/free_software_for_formal_verification`.

[82] Isabelle Theorem Prover. `http://www.cl.cam.ac.uk/research/hvg/isabelle`.

[83] NuSMV Model Checker. `http://nusmv.fbk.eu`.

[84] Event-B. `http://www.event-b.org/`.

[85] Microsoft's Verifier for Concurrent C. `http://research.microsoft.com/en-us/projects/vcc`.

[86] Prover Platform. `http://why3.lri.fr`.

[87] Railway applications – communication, signalling and processing systems – safety related electronic systems for signalling. Norm EN 50129:2010, CENELEC, Brussels, Belgium, 2011.

[88] Railway applications – Communication, signalling and processing systems – Safety-related communication in transmission systems. Norm EN 500159:2010, CENELEC, Brussels, Belgium, 2010.

[89] Railway applications – the specification and demonstration of Reliability, Availability, Maintainability and Safety (RAMS). Norm EN 50126-1:1999, CENELEC, Brussels, Belgium, 1999.

[90] Railway applications. the specification and demonstration of reliability, availability, maintainability and safety (RAMS). guide to the application of EN 50126-1 for rolling stock RAM. Norm PD CLC/TR 50126-2:2008, CENELEC, Brussels, Belgium, 2008.

[91] Railway applications. the specification and demonstration of reliability, availability, maintainability and safety (RAMS). guide to the application of EN 50126-1 for safety. Norm PD CLC/TR 50126-2:2007, CENELEC, Brussels, Belgium, 2007.