



**EULYNX Initiative**

### **EULYNX Partners:**

Bane NOR  
Société Nationale des Chemins de Fer Luxembourgeois (CFL)  
DB Netz AG (DB)  
S.A. Infrabel  
Liikennevirasto (FTA)  
Network Rail  
ProRail B.V.  
Rete Ferroviaria Italiana (RFI)  
SBB AG  
Société Nationale des Chemins de Fer Français (SNCF)  
SŽ-Infrastruktura, d.o.o. (SŽ)  
Trafikverket

## **Verification and validation plan**

Document number: Eu.Doc.31

Baseline: 1.0 (0.A)

EULYNX Baseline Set: 3



## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Release information	1
1.2	Impressum	2
1.3	Purpose	2
1.4	Applicable standards and regulations	2
1.5	Applicable documents	3
1.6	Terms and abbreviations	3
1.7	Definition of object types	3
<b>2</b>	<b>Verification and validation of the model</b>	<b>3</b>
2.1	Introduction for verification and validation of the model	3
2.2	Roles	5
2.3	Artefacts	5
2.3.1	Refinement of artefact MODEL SUBSET	5
2.3.2	Refinement of artefact MODEL TEST SPECIFICATION	6
2.3.3	New artefact TEST REPORT	6
2.4	Verification and validation process of MODEL SUBSET	6
2.4.1	Overview of Model Verification and Validation	7
2.4.2	Requirement Verification	8
2.4.3	Model Subset Verification	8
2.4.3.1	Black-Box verification	9
2.4.3.2	White-Box verification	9
2.4.4	Model Validation	10
2.4.4.1	Model Subset Validation	10

ID	Type	Requirements
Eu.VV.1	Head	<b>1 Introduction</b>
Eu.VV.8	Head	<b>1.1 Release information</b>
Eu.VV.5	Info	[Eu.Doc.31] Verification and validation plan Version: 1.0 (0.A) CENELEC Phase: 2 EULYNX Baseline Set: 3 Approval date: 29.11.2018
Eu.VV.9	Info	<b>Version history</b>
Eu.VV.10	Info	version number: 0.0 date: 26.10.2017 author: Charlotte Gäbel, Oliver Lemke review: changes: initial version
Eu.VV.77	Info	version number: 0.1 (0.A) date: 10.11.2017 author: Charlotte Gäbel, Oliver Lemke review: Mirko Blasic, Marie Killat changes: EUAR-136, EUAR-137, EUAR-138, EUAR-205
Eu.VV.107	Info	version number: 0.1 (1.A) date: 15.06.2018 author: Charlotte Gäbel review: Mirko Blasic changes: EUAR-208
Eu.VV.109	Info	version number: 0.2 (0.A) date: 24.10.2018 author: Charlotte Gäbel review: CCB review changes: EUMT-43, EUMT-44, EUMT-45

ID	Type	Requirements
Eu.VV. 120	Info	version number: 1.0 (0.A) date: 12.12.2018 author: Charlotte Gäbel review: CCB changes: EUMT-47, EUMT-48
Eu.VV. 11	Head	<b>1.2 Impressum</b>
Eu.VV. 12	Info	Publisher: <b>EULYNX Initiative</b>  <b>EULYNX Partners:</b> Bane NOR Société Nationale des Chemins de Fer Luxembourgeois (CFL) DB Netz AG (DB) S.A. Infrabel Liikennevirasto (FTA) Network Rail ProRail B.V. Rete Ferroviaria Italiana (RFI) SBB AG Société Nationale des Chemins de Fer Français (SNCF) SŽ-Infrastruktura, d.o.o. (SŽ) Trafikverket
Eu.VV. 94	Info	Responsible for this document: EULYNX Project Management Office <a href="http://www.eulynx.eu">www.eulynx.eu</a>
Eu.VV. 95	Info	Copyright EULYNX Partners All information included or disclosed in this document is licensed under the European Union Public Licence EUPL, Version 1.1.
Eu.VV. 13	Head	<b>1.3 Purpose</b>
Eu.VV. 14	Info	The purpose of this document is to define the process for verification and validation of EULYNX model based specifications.
Eu.VV. 97	Head	<b>1.4 Applicable standards and regulations</b>

ID	Type	Requirements
Eu.VV.98	Info	A list of applicable standards and regulations used in EULYNX is listed in the EULYNX Reference Document List [Eu.Doc.12].
Eu.VV.99	Head	<b>1.5 Applicable documents</b>
Eu.VV.100	Info	The current versions of documents used as input or related to this document are listed in the EULYNX Documentation Plan [Eu.Doc.11]. The relationships between the documents are displayed in the Appendix A1 Documentation plan and structure [Eu.Doc.11_A1].
Eu.VV.101	Head	<b>1.6 Terms and abbreviations</b>
Eu.VV.102	Info	The terms and abbreviations are listed in the EULYNX Glossary [Eu.Doc.9].
Eu.VV.103	Head	<b>1.7 Definition of object types</b>
Eu.VV.104	Info	The following definition for object types is applied in this document:
Eu.VV.105	Info	<ul style="list-style-type: none"> <li>• "Info" - This denotes additional information to help understand the specification. These objects do not specify any additional requirements.</li> </ul>
Eu.VV.106	Info	<ul style="list-style-type: none"> <li>• "Head" - This denotes chapter headings.</li> </ul>
Eu.VV.3	Head	<b>2 Verification and validation of the model</b>
Eu.VV.4	Head	<b>2.1 Introduction for verification and validation of the model</b>
Eu.VV.23	Info	System development for railway systems according to the CENELEC standards follows the V-model, which comprises verification and validation are important tasks.
Eu.VV.24	Info	During the specification phases of the V-model, verification and validation are important activities, applied to assure the quality of the specification itself.
Eu.VV.25	Info	This is especially true for model based specifications, because models of the intended system behaviour can be seen as abstract reference implementations of the future system. Following this notion, it is valid to define a "small V"-process within the specification phases of the underlying "big V"-CENELEC process.

ID	Type	Requirements
Eu.VV.27	Info	<p>The diagram illustrates the 'small V' process of the 'big V' - CENELEC process. It shows a large 'V' shape representing the development and testing phases, and a smaller 'v' shape representing the verification and validation phases. The development phases are: User Requirements (IM Expert Knowledge), Formalised Requirements (Sequence Diagrams), State Machine Implementation, State Machine Testing, and IM Acceptance. The verification and validation phases are: Verification of Requirements (SDs) by IMs, Verification of State Machines by Modellers, and Validation of State Machine Models by IMs. Arrows indicate the flow of information and the relationships between these phases.</p> <p>Fig. 1.: The "small V"-process of the "big V" - CENELEC process</p>
Eu.VV.112	Info	For the "big V" - CENELEC process the activities and artefacts are described in Eu.Doc.27 "System engineering process".
Eu.VV.113	Info	The "small V" is highlighted in the "big V" and pictures the relationships of verification and validation for the STATE MACHINE DIAGRAM [Eu.VV.35] development.
Eu.VV.110	Info	User requirements are described in Eu.Doc.30 "Modelling Standard" [Eu.ModStan.108]. The common artefact in EULYNX is the "FUNCTION LIST". It lists the required functions based on IM requirements and is the input document for the Formalised requirements.

ID	Type	Requirements
Eu.VV.111	Info	Formalised requirements are artefacts of the MODEL SUBSET, e.g. SEQUENCE DIAGRAM [Eu.VV.34]. Those artefacts realize the required functions and are input documents for the STATE MACHINE DIAGRAM [Eu.VV.35]. The STATE MACHINE DIAGRAMS are verified against the Formalised requirements (SEQUENCE DIAGRAMS).
Eu.VV.114	Info	STM acceptance is the validation against the user requirements by Infrastructure Manager (IM). Every IM is responsible for their own required user requirements. By testing the implemented function on the EXECUTABLE SIMULATION of the STM the IM validates whether the simulation behaviour is according to IM specific expectations. After the STM is accepted by every participating IM, the "small V"-process ends and the "big V" - CENELEC process goes forward.
Eu.VV.2	Head	<b>2.2 Roles</b>
Eu.VV.28	Info	This inner verification and validation process is part of the MBSE process, which is explained from the modelling perspective at the Modelling Standard [Eu.Doc.30], e.g. in ID [Eu.ModSt.340].
Eu.VV.29	Info	For verification and validation, the MBSE Process defines three roles:
Eu.VV.30	Info	<ul style="list-style-type: none"> <li>• CLUSTER ENGINEER</li> <li>• MODEL VERIFIER and</li> <li>• MODEL VALIDATOR.</li> </ul>
Eu.VV.31	Info	The definition of these roles is also valid in this document.
Eu.VV.15	Head	<b>2.3 Artefacts</b>
Eu.VV.32	Info	In the following chapters, the artefacts used during the verification and validation process are defined. This is either done by refining artefacts described in the System Engineering process [Eu.Doc.27] or by introducing new artefacts.
Eu.VV.17	Head	<b>2.3.1 Refinement of artefact MODEL SUBSET</b>
Eu.VV.33	Info	According to the System engineering process in ID [Eu.SEP.49] the MODEL SUBSET refers to a subset of the specification model. For the verification and validation process, the MODEL SUBSET represents this part of the model, which needs to be verified and validated. This MODEL SUBSET can be further refined into the following artefacts:
Eu.VV.34	Info	SEQUENCE DIAGRAM (SD) specifying the intended stimulus-response-behaviour of the MODEL SUBSET. These diagrams are used during the specification phase to capture the basic functional requirements in a semi-formal way. Based on these diagrams, the STATE MACHINE DIAGRAMS are derived.

ID	Type	Requirements
Eu.VV.35	Info	STATE MACHINE DIAGRAMS (STM), as a graphical representation of the MODEL SUBSET behaviour.
Eu.VV.36	Info	An EXECUTABLE SIMULATOR comprising the MODEL SUBSET behaviour and a GRAPHICAL USER INTERFACE (GUI) for triggering the stimuli and visualising the responses.
Eu.VV.18	Head	<b>2.3.2 Refinement of artefact MODEL TEST SPECIFICATION</b>
Eu.VV.37	Info	According to the System engineering process in ID [Eu.SEP.61] the MODEL TEST SPECIFICATION comprises the information suitable to sufficiently test the behaviour of the MODEL SUBSET. The MODEL TEST SPECIFICATION consists of one or more TEST CASES.
Eu.VV.38	Info	A TEST CASE comprises meta-information (Creator, date, subsystem covered, IM applicability) and a test script. The test script contains a list of steps to instruct the MODEL VERIFIER or MODEL VALIDATOR how to execute the TEST CASE. The test script should clearly indicate the stimuli to be performed during the test as well as the expected results to be observed.
Eu.VV.39	Info	For dynamic test (also known as unit test) the TEST CASE specifies a precondition which requires a state. This state is usually not the initial state of the sub-/system behaviour. A Test Suite combines several Test Cases to create the logical sequence of states. When designing a Test Suite the first Test Case is with the initial state and creates with its Postcondition the Precondition of the following Test Case, and so on.
Eu.VV.19	Head	<b>2.3.3 New artefact TEST REPORT</b>
Eu.VV.40	Info	The TEST REPORT documents the results of one test run. The TEST REPORT comprises status information on every TEST CASE included in the MODEL TEST SPECIFICATION.
Eu.VV.41	Info	Possible statuses for the TEST CASES are:
Eu.VV.42	Info	<ul style="list-style-type: none"> <li>• not executed – TEST CASES not included in the current test run</li> <li>• in progress – currently executed in the test run,</li> <li>• pass – test completed successfully in the test run or</li> <li>• fail – test failed in the test run.</li> </ul>
Eu.VV.16	Head	<b>2.4 Verification and validation process of MODEL SUBSET</b>
Eu.VV.43	Info	Verification and validation of the MODEL SUBSET is done by executing the EXECUTABLE SIMULATOR and using its GRAPHICAL USER INTERFACE to execute the TEST CASES.



ID	Type	Requirements
Eu.VV.20	Head	<b>2.4.1 Overview of Model Verification and Validation</b>
Eu.VV.44	Info	This chapter describes the workflow for performing verification and validation of the MODEL SUBSET. The workflow is split into two main phases: Verification (Eu.VV.21) and validation (Eu.VV.22) (see picture in ID Eu.VV.50 for an overview on both phases).
Eu.VV.45	Info	The iterative development of the MODEL SUBSET is accompanied by a synchronous, iterative verification. For every iterative development of the STATE MACHINE DIAGRAMS, a verification phase will be performed. This process finishes, when all requirements are implemented in the STATE MACHINE DIAGRAMS and are successfully verified.
Eu.VV.46	Info	The verification itself consists of two different sub-phases: Black-Box verification and White-Box verification, which are described in the chapter 2.4.3 Model Subset Verification.
Eu.VV.47	Info	The end criterion for the verification is reached, as soon as Black-Box verification and White-Box verification are successfully finished.
Eu.VV.48	Info	After the completion of the verification phase, the verified MODEL SUBSET is released and distributed to the IMs for validation. The validation activities are described in chapter 2.4.4 Model Validation.
Eu.VV.49	Info	In case of an unsuccessful validation, a modification of the MODEL SUBSET is necessary. After this the cycle of verification and validation starts again, until verification as well as validation is completed successfully.
Eu.VV.50	Info	

ID	Type	Requirements
		<pre> graph TD     Start([Start]) --&gt; Verify[Verify the MODEL SUBSET during development.]     Verify --&gt; EndCriteria{Is the end criterion for verification reached?}     EndCriteria -- NO --&gt; Verify     EndCriteria -- YES --&gt; Release[Release MODEL SUBSET for Delivery to IMs.]     Release --&gt; Validate[Validate the released MODEL SUBSET.]     Validate --&gt; ValidationSuccess{Is the Validation successful?}     ValidationSuccess -- NO --&gt; Verify     ValidationSuccess -- YES --&gt; Baseline([Baseline]) </pre> <p>Fig. 2: Short Overview of Verification and Validation Process</p>
Eu.VV. 115	Head	<b>2.4.2 Requirement Verification</b>
Eu.VV. 116	Info	Requirement verification confirms that the SEQUENCE DIAGRAM accurately reflects the requirements of all IMs. These user requirements are defined: <ul style="list-style-type: none"> <li>• as domain knowledge of the stakeholders,</li> <li>• in informal documents, e.g. CONCEPTUAL DOCUMENTS, FUNCTIONAL LISTs etc.</li> </ul>
Eu.VV. 117	Info	From these user requirements, the formal requirements are determined in the form of SEQUENCE DIAGRAMS. The verification of SEQUENCE DIAGRAMS is undertaken by the IM and captured as the IM endorsement of the specification.
Eu.VV. 118	Info	SEQUENCE DIAGRAMS are linked to the FUNCTIONAL LISTs at a high level, providing additional traceability for the verification of requirements.
Eu.VV. 21	Head	<b>2.4.3 Model Subset Verification</b>

ID	Type	Requirements
Eu.VV. 51	Info	Basic target of the model subset verification is to verify that the behaviour specified in the STATE MACHINE DIAGRAMS is consistent to the requirements specified as SEQUENCE DIAGRAMS. This is done by deriving a MODEL TEST SPECIFICATION from the SEQUENCE DIAGRAMS and testing the EXECUTABLE SIMULATOR.
Eu.VV. 52	Head	<b>2.4.3.1 Black-Box verification</b>
Eu.VV. 54	Info	During Black-Box verification it is verified, that the STATE MACHINE DIAGRAMS do not specify behaviour that violates the specification in the SEQUENCE DIAGRAMS.
Eu.VV. 55	Info	The MODEL VERIFIER creates a MODEL TEST SPECIFICATION (for verification) by using the SEQUENCE DIAGRAMS as input information representing the requirements. The SEQUENCE DIAGRAMS contain the stimulus-response-behaviour agreed upon with the participating IMs.
Eu.VV. 56	Info	The CLUSTER ENGINEER uses the same input information for developing the STATE MACHINE DIAGRAMS. Thus, the actual stimulus-response-behaviour in the EXECUTABLE SIMULATOR should not differ from the behaviour defines in the SEQUENCE DIAGRAMS.
Eu.VV. 57	Info	It is recommended that the Verifier finishes the Black Box verification before the White Box verification starts. The Black-Box verification by the Verifier should be performed without knowing or being influenced by the STATE MACHINE DIAGRAMS.
Eu.VV. 58	Info	During execution of the Black-Box verification, the EXECUTABLE SIMULATOR is stimulated according to the MODEL TEST SPECIFICATION (for verification). The actual responses from the EXECUTABLE SIMULATOR are checked against the expected responses defined in the MODEL TEST SPECIFICATION (for verification). If the actual response is different from the expected response, the test case fails.
Eu.VV. 59	Info	The Black-Box verification ends successfully, if: <ul style="list-style-type: none"> <li>• all test cases have been executed and</li> <li>• all tests cases pass.</li> </ul>
Eu.VV. 53	Head	<b>2.4.3.2 White-Box verification</b>
Eu.VV. 60	Info	During White-Box verification it is verified, that no implicit behaviour has been added to the STATE MACHINE DIAGRAMS by the CLUSTER ENGINEER while designing the STATE MACHINE DIAGRAMS. Implicit behaviour is defined as behaviour that has not been agreed upon by the IMs beforehand and is not specified as SEQUENCE DIAGRAMS.
Eu.VV. 61	Info	Thus the White-Box verification aims at <ul style="list-style-type: none"> <li>• a methodically correct STATE MACHINE DIAGRAM and</li> <li>• coverage analysis of the STATE MACHINE DIAGRAMS against the SEQUENCE DIAGRAMS</li> </ul>
Eu.VV. 62	Info	The first aspect is done by verifying the semantics and syntax of the STATE MACHINE DIAGRAM either manually or automatically by using script based checks.

ID	Type	Requirements
Eu.VV.63	Info	The second aspect is realised by evaluating the coverage of states and transitions of the STATE MACHINE DIAGRAM by the SEQUENCE DIAGRAMS. As coverage criterion, at least full state and transition coverage should be considered. The analysis itself is performed by marking the states and transitions that have been traversed while executing all scenarios defined in the SEQUENCE DIAGRAMS. If not all states and transitions are covered by SEQUENCE DIAGRAMS, these model elements may point at implicit knowledge in the STATE MACHINE DIAGRAM.
Eu.VV.64	Info	To analyse these model elements, additional scenarios are created that explicitly cover the yet uncovered elements.
Eu.VV.65	Info	Every additional scenario resulting from that analysis has to be analysed with the participating IMs.
Eu.VV.66	Info	If a scenario is accepted by the IMs as a valid, useful scenario, then: <ul style="list-style-type: none"> <li>• the scenario has to be added to the SEQUENCE DIAGRAMS and the associated function to the FUNCTIONAL LIST</li> <li>• the STATE MACHINE DIAGRAMS have to be re-verified.</li> </ul>
Eu.VV.67	Info	If a scenario is not valid, then: <ul style="list-style-type: none"> <li>• the model elements covered by this scenario have to be removed from the STATE MACHINE DIAGRAM.</li> </ul>
Eu.VV.68	Info	The White-Box verification ends successfully, if: <ul style="list-style-type: none"> <li>• the semantics and syntax of the STATE MACHINE DIAGRAM are correct and</li> <li>• the coverage criterion is met.</li> </ul>
Eu.VV.22	Head	<b>2.4.4 Model Validation</b>
Eu.VV.119	Head	<b>2.4.4.1 Model Subset Validation</b>
Eu.VV.69	Info	After verification, the verified MODEL SUBSET must be validated by the original stakeholders (e.g. the IMs) against their user requirements. These user requirements are as defined in Eu.VV.116.
Eu.VV.70	Info	During validation, the stakeholders become the MODEL VALIDATORS of the MODEL SUBSET.
Eu.VV.71	Info	To ensure a validation which is independent of the verification, <ul style="list-style-type: none"> <li>• the validation should be easily accessible for the domain experts and</li> <li>• as unrestricted by formal constraints as possible.</li> </ul>
Eu.VV.72	Info	The first aspect is fulfilled by the usage of the EXECUTABLE SIMULATOR that hides the complexity of the STATE MACHINE DIAGRAMS from the user of the EXECUTABLE SIMULATOR. The EXECUTABLE SIMULATOR can be run on all Windows PCs and therefore can be easily distributed to the MODEL VALIDATORS involved.

ID	Type	Requirements
Eu.VV.73	Info	The second aspect is realised by applying different TEST CASES from the TEST CASES used in the verification. Furthermore, the MODEL VALIDATOR is forced to create his/her own TEST CASES based on his/her domain knowledge and experiences. The FUNCTIONAL LIST can act as a guideline to assure coverage of all relevant functionality.
Eu.VV.74	Info	To document the validation process, two basic possibilities exist:
Eu.VV.75	Info	<ul style="list-style-type: none"> <li>• A-priori documentation of test cases: the MODEL VALIDATOR specifies his/her own TEST CASES by using an empty template document. The MODEL VALIDATOR then executes the TEST CASES by using the EXECUTABLE SIMULATOR and documents the results (pass/fail).</li> </ul>
Eu.VV.76	Info	<ul style="list-style-type: none"> <li>• A-posterior documentation of the test cases: the MODEL VALIDATOR uses their domain knowledge to design further test sequences that may attempt to demonstrate positive or negative behaviour that may not have been considered in advance. The MODEL VALIDATOR records the test sequences (stimuli and responses) and verdict (pass/fail) along with the a-priori documentation.</li> </ul>
Eu.VV.78	Info	The validation process is finished successfully when all participating IMs provide evidence that their user requirements are satisfied by the STATE MACHINE DIAGRAMS.
Eu.VV.79	Info	The successful validation process leads to a production of a new baseline.