EULYNX Initiative

Europe's Rail Joint Undertaking

Requirements specification for subsystem Level Crossing

Document number: Eu.Doc.108
Version: 2.3 (0.A)

Contents

1 Introduction 1
1.1 Release information 1
1.2 Impressum 1
1.3 Purpose 2
1.4 Applicable standards and regulations 2
1.5 Applicable documents 2
1.6 Terms and abbreviations 2
1.7 Variability management 2
1.8 Definition of object types 2
1.9 Modelling 2

2 Conditions of use 3
2.1 Functional packages 3

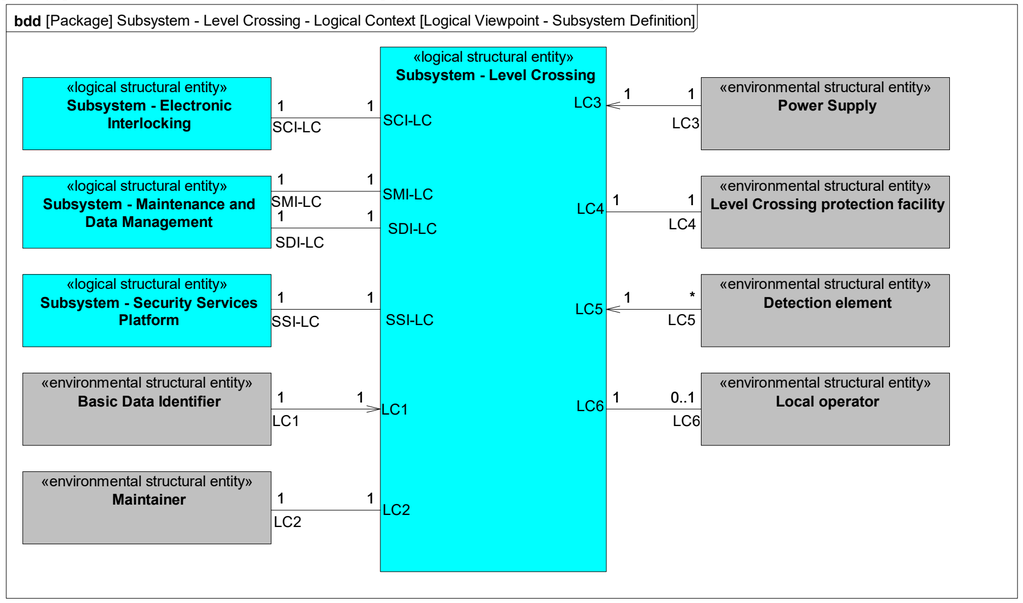
3 Functional requirements specification 3
3.1 Subsystem - Level Crossing - General Infos and Assumptions 3
3.2 Subsystem - Level Crossing - Logical Viewpoint 3
3.2.1 Subsystem - Level Crossing - Logical Context 3
3.3 Subsystem - Level Crossing - Functional Viewpoint 4
3.3.1 Definition of time values 4
3.3.2 Subsystem - Level Crossing - Functional Context 4
3.3.3 Subsystem - Level Crossing - Functional Partitioning 25
3.3.4 Subsystem - Level Crossing - Functional Architecture 26
3.3.5 Subsystem - Level Crossing - Functional Entities 28
3.4 Subsystem - Level Crossing - Interfaces 41
3.4.1 SCI-LC (Subsystem - Electronic Interlocking) 41
3.4.1.1 SCI-LC - Logical Viewpoint 41
3.4.1.1.1 SCI-LC - Logical Context 41
3.4.1.2 SCI-LC - Information Flows 41
3.4.1.3 SCI-LC - Functional Viewpoint 43
3.4.1.3.1 SCI-LC - Functional Partitioning 43
3.4.1.3.2 SCI-LC - Functional Architecture 43
3.4.1.3.3 SCI-LC - Functional Entities 44
3.4.2 SMI-LC (Subsystem - Maintenance and Data Management) 56
3.4.3 SDI-LC (Subsystem - Maintenance and Data Management) 56
3.4.4 SSI-LC (Subsystem - Security Services Platform) 56
3.4.5 LC1 (Basic Data identifier) 56
3.4.6 LC2 (Maintainer) 56
3.4.7 LC5 (Detection element) 56
3.4.8 LC6 (Local operator) 56
3.4.9 LC4 (Level Crossing protection facility) 57

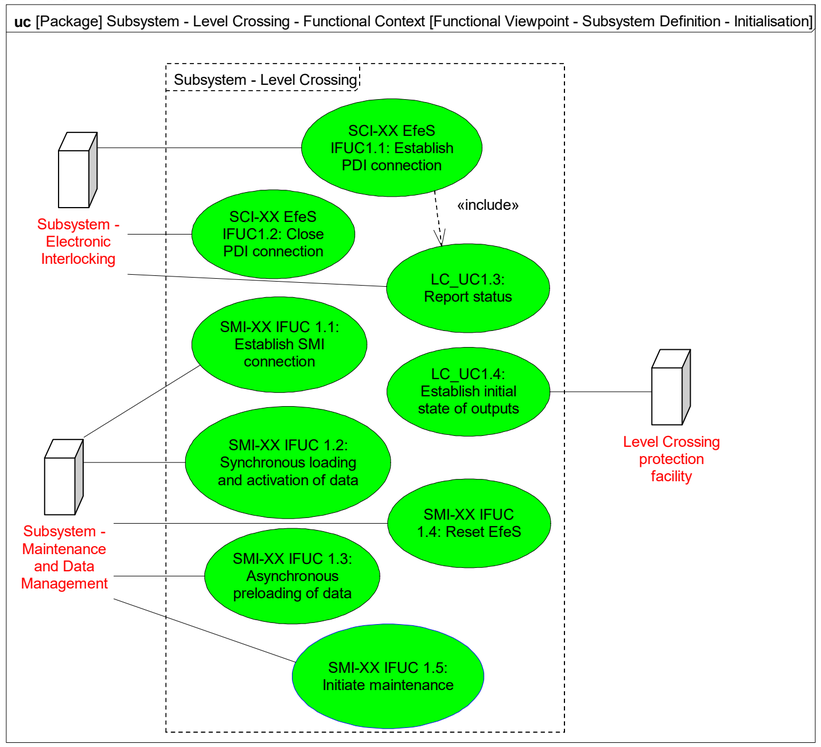
4 RAMSS requirements 57

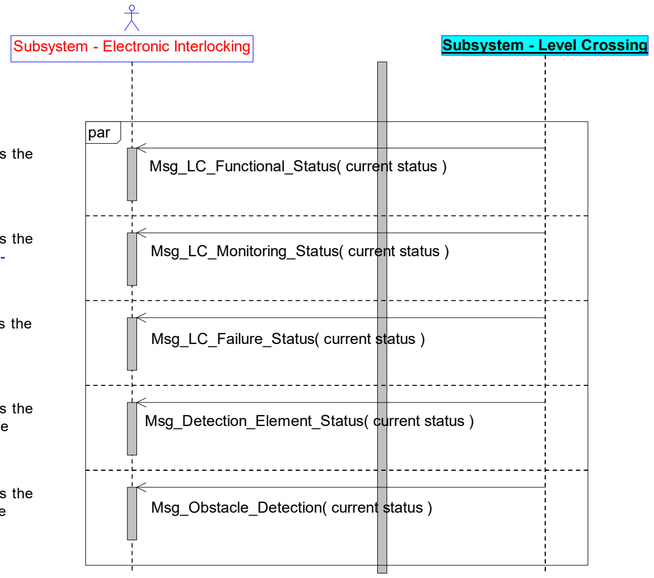
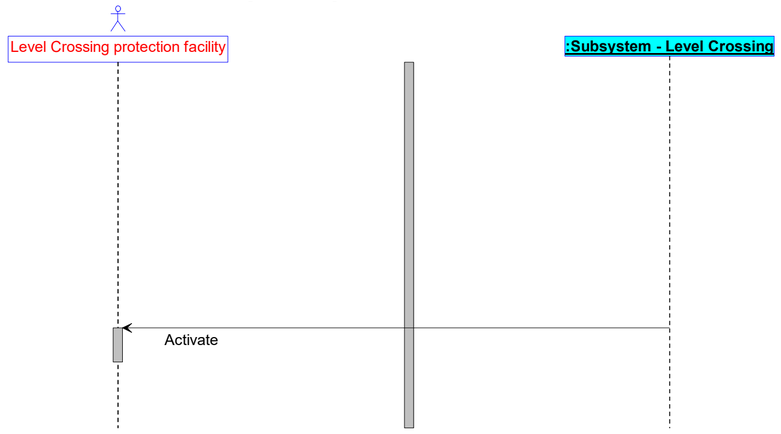
5 Technical requirements 57
5.1 Specific technical interface requirements 57
5.1.1 Interface to the Point of Service - Signalling (PoS - Signalling) 57
5.1.2 Interface to the Detection element 57
5.1.3 Interface to the Local operator 57
5.1.4 Interface to the Level Crossing protection facility 57
5.2 Time behaviour 58
5.2.1 Response times 58
5.3 Configuration and engineering data 58
5.3.1 Specific data 58
5.3.2 Value configuration 59

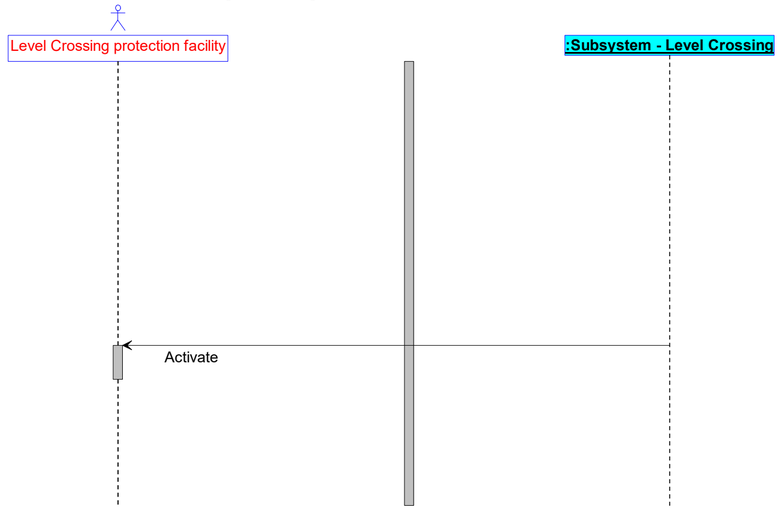
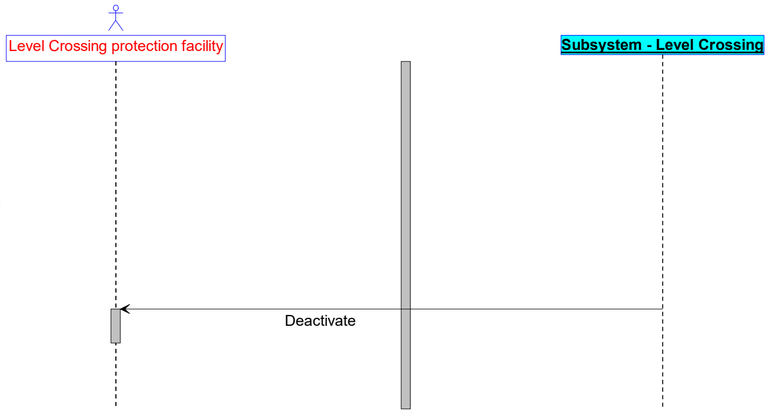
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.1	Head	**1 Introduction**		
Eu.LC.2	Head	**1.1 Release information**		
Eu.LC.3	Info	[Eu.Doc.108] Requirements specification for subsystem Level crossing CENELEC Phase: 4 Version: 2.3 (0.A) Approval date: 29.05.2024		
Eu.LC.4	Info	**Version history**		
Eu.LC.3305	Info	version number: 2.0 (0.A) date: 16.05.2022 author: Philipp Wolber model version: 18 Generic interface and subsystem requirements version: 4.0 (0.A) Generic interface and subsystem requirements for SCI version: 1.0 (0.A) review: CCB changes: EULX-485, EULX-490, EULX-491, EULX-496, EULX-500, EULX-503		
Eu.LC.3828	Info	version number: 2.1 (0.A) date: 29.03.2023 author: Philipp Wolber, Filip Giering, Marie Gehrmann, Dominik Smajgl model version: 21 Generic interface and subsystem requirements version: 4.0 (1.A) Generic interface and subsystem requirements for SCI version: 1.0 (1.A) review: Filip Giering changes: EULX-489, EULX-511, EULX-518, EULX-519, EULX-520, EULX-524, EULX-525, EULX-526, EULX-527, EULX-528, EULX-529, EULX-530, EULX-531		
Eu.LC.3882	Info	version number: 2.1 (1.A) date: 11.05.2023 author: Philipp Wolber model version: 22 Generic interface and subsystem requirements version: 4.0 (1.A) Generic interface and subsystem requirements for SCI version: 1.0 (1.A) review: cluster changes: EULX-534, EULX-535, EULX-536, EULX-537, EULX-538, EULX-542, EULX-543, EULX-546, EULX-548, EULX-550		
Eu.LC.3917	Info	version number: 2.2 (0.A) date: 28.06.2023 author: Philipp Wolber model version: 22 Generic interface and subsystem requirements version: 4.0 (3.A) Generic interface and subsystem requirements for SCI version: 1.0 (3.A) review: TACS Mirror Group changes: EULX-540, EULX-542, EULX-533, EULX-551, EULX-557, EULX-560, EULX-562, EULX-563		
Eu.LC.3942	Info	version number: 2.2 (1.A) date: 15.12.2023 author: Philipp Wolber model version: 25 Generic interface and subsystem requirements version: 4.0 (4.A) Generic interface and subsystem requirements for SCI version: 1.0 (4.A) review: M&T changes: EULX-517, EULX-565, EULX-566, EULX-567, EULX-569, EULX-570, EULX-574, EULX-575, EULX-576, EULX-577, EULX-579, EULX-580, EULX-582, EULX-585, EULX-589, EULX-591, EULX-592, EULX-594, EULX-595, EULX-597		
Eu.LC.4095	Info	version number: 2.2 (2.A) date: 17.04.2024 author: Philipp Wolber model version: 26 Generic interface and subsystem requirements version: 4.0 (4.A) Generic interface and subsystem requirements for SCI version: 1.0 (4.A) review: - changes: EULX-571, EULX-581, EULX-568, EULX-593, EULX-596, EULX-599, EULX-601, EULX-602, EULX-603, EULX-604, EULX-605, EULX-606, EULX-607, EULX-608, EULX-610, EULX-611, EULX-612		
Eu.LC.4264	Info	version number: 2.2 (3.A) date: 03.05.2024 author: Philipp Wolber model version: 26 Generic interface and subsystem requirements version: 4.0 (5.A) Generic interface and subsystem requirements for SCI version: 1.0 (5.A) review: cluster changes: EULX-613, EULX-614, EULX-617, EULX-618, EULX-619, EULX-627, EULX-628		
Eu.LC.4283	Info	version number: 2.3 (0.A) date: 18.06.2024 author: Philipp Wolber model version: 28 Generic interface and subsystem requirements version: 4.0 (6.A) Generic interface and subsystem requirements for SCI version: 1.1 (0.A) review: TACS Mirror Group changes: EULX-627, EULX-633, EULX-634, EULX-636, EULX-637, EULX-638, EULX-641		
Eu.LC.32	Head	**1.2 Impressum**		

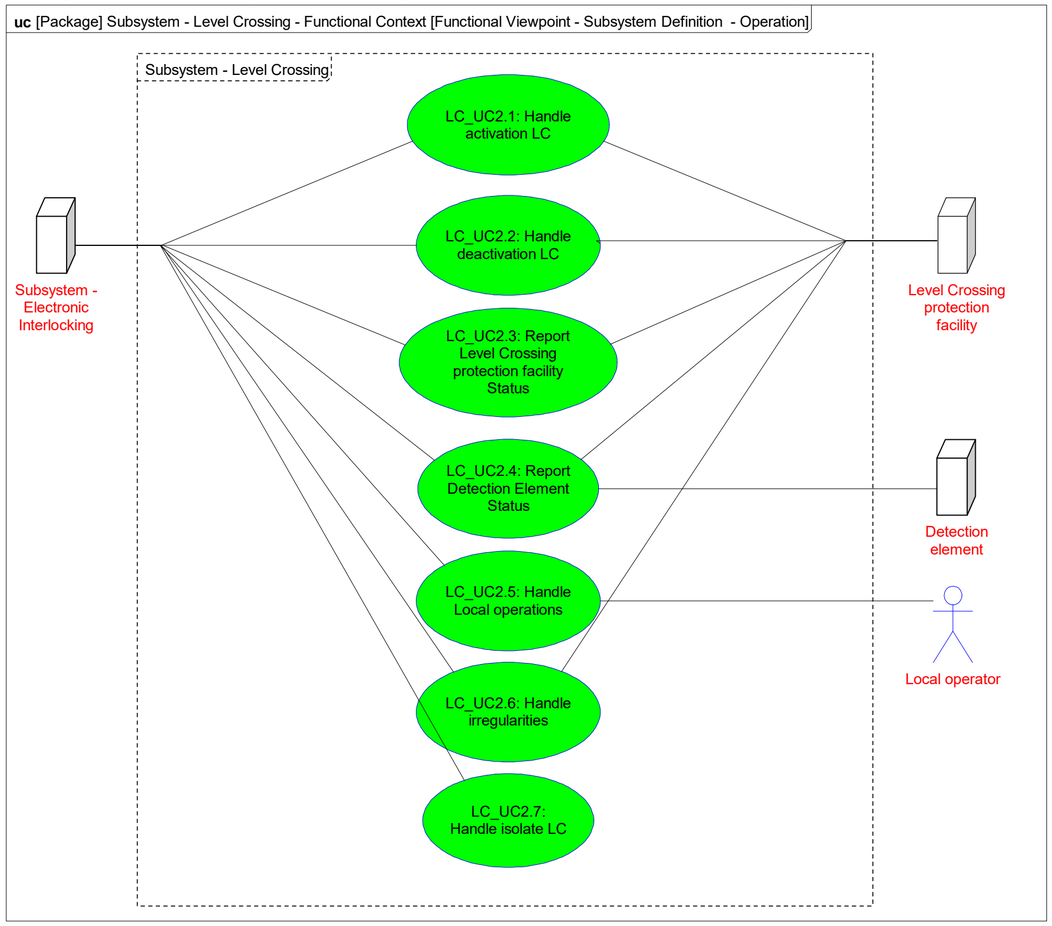
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.33	Info	Publisher: **Europe's Rail Joint Undertaking** https://rail-research.europa.eu/ **EULYNX Initiative** https://eulynx.eu/		
Eu.LC.34	Info	Responsible for this document: EU-Rail System Pillar Trackside Assets Control and Supervision domain		
Eu.LC.35	Info	This document is drafted by and belongs to EU Rail. EU Rail encourages the distribution and re-use of this document, the technical specifications and the information it contains. EU Rail holds several intellectual property rights, such as copyright and trade mark rights, which need to be considered when this document is used. EU Rail authorizes you to re-publish, re-use, copy and store this document without changing it, provided that you indicate its source and include the following mention [EU Rail trade mark, title of the document, year of publication, version of document]. EU Rail makes no representation or warranty as to the accuracy or completeness of the information contained within these documents. EU Rail shall have no liability to any party as a result of the use of the information contained herein. EU Rail will have no liability whatsoever for any indirect or consequential loss or damage, and any such liability is expressly excluded. You may study, research, implement, adapt, improve and otherwise use the information, the content and the models in this document for your own purposes. If you decide to publish or disclose any adapted, modified or improved version of this document, any amended implementation or derivative work, then you must indicate that you have modified this document, with a reference to the document name and the terms of use of this document. You may not use EU Rail's trade marks or name in any way that may state or suggest, directly or indirectly, that EU Rail is the author of your adaptations. EU Rail cannot be held responsible for your product, even if you have used this document and its content. It is your responsibility to verify the quality, completeness and the accuracy of the information you use, for your own purposes.		
Eu.LC.36	Head	**1.3 Purpose**		
Eu.LC.37	Info	The purpose of the document is the specification of functional requirements for the Subsystem - Level Crossing.		
Eu.LC.38	Info	This document describes the functional requirements for the Subsystem - Level Crossing.		
Eu.LC.39	Info	This document is intended for the following users: • safety authorities • infrastructure managers • safety assessors • signalling system suppliers • validators		
Eu.LC.40	Info	This document is the basis for the implementation by the supplier and for approval by the infrastructure manager.		
Eu.LC.3915	Info	This document is applicable for both the EU-Rail System Pillar target architecture and the EULYNX architecture. The document is delivered as a single specification fitting both the System Pillar documentation sets and the EULYNX documentation sets. EU-Rail System Pillar is the technical authority for this document.		
Eu.LC.41	Head	**1.4 Applicable standards and regulations**		
Eu.LC.42	Info	A list of applicable standards and regulations used in EULYNX is listed in the EULYNX Reference Document List [Eu.Doc.12].		
Eu.LC.43	Head	**1.5 Applicable documents**		
Eu.LC.44	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.LC.45	Head	**1.6 Terms and abbreviations**		
Eu.LC.46	Info	The terms and abbreviations are listed in the EULYNX Glossary [Eu.Doc.9].		
Eu.LC.47	Head	**1.7 Variability management**		
Eu.LC.48	Info	This document describes harmonised requirements. Variability management is not applicable.		
Eu.LC.49	Head	**1.8 Definition of object types**		
Eu.LC.50	Info	The following definition for object types is applied in this document:		
Eu.LC.51	Info	• "Req" - This denotes a mandatory requirement.		
Eu.LC.3946	Info	• "Def" - This denotes referenceable model elements that are used in the model-based creation of requirements		
Eu.LC.53	Info	• "Info" - This denotes additional information to help understand the specification. These objects do not specify any additional requirements.		
Eu.LC.54	Info	• "Head" - This denotes chapter headings.		
Eu.LC.55	Head	**1.9 Modelling**		
Eu.LC.56	Info	The section "Functional requirements specification" follows a model based systems engineering process using Systems Modelling Language (SysML) and defines the functional system requirements for the Subsystem - Level Crossing operational in stimulus-response form. Furthermore the information objects (stimuli and responses) exchanged over the interfaces of the Subsystem - Level Crossing are defined.		
Eu.LC.57	Info	The diagrams presented in this document are modelled in SysML [SysML].		
Eu.LC.58	Info	The rules for the interpretation of the model based parts of specification are defined in [Eu.Doc.29].		
Eu.LC.59	Info	In chapter 3 "Functional requirements specification" the functional system requirements, defined in the form of a SysML model in the PTC Integrity Modeler are depicted as a surrogate of this model in the form of DOORS-objects.		
Eu.LC.60	Info	A requirement thereby consists of the respective SysML model element, for instance a SysML diagram, and if necessary an additional extension of the requirement.		

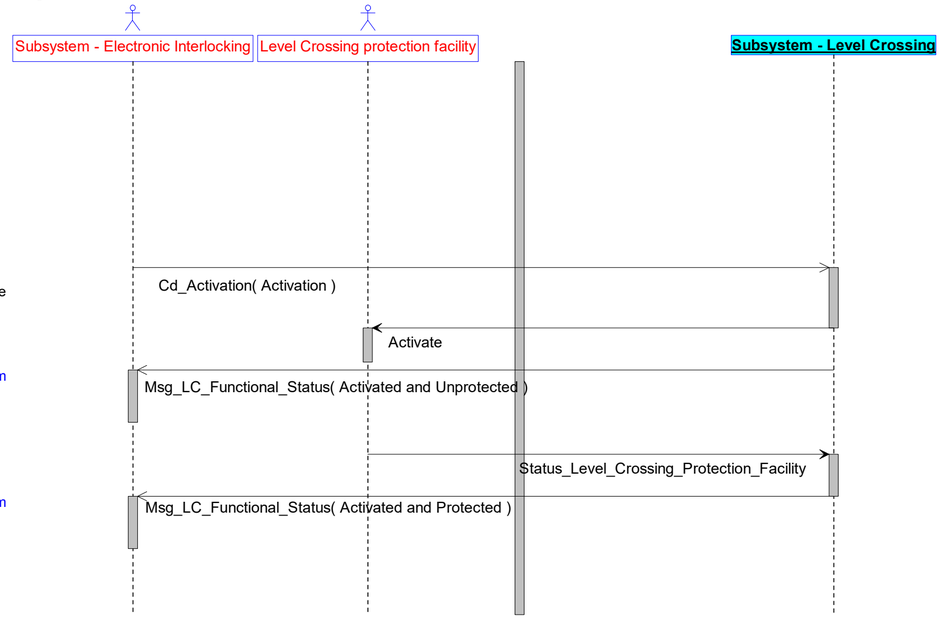
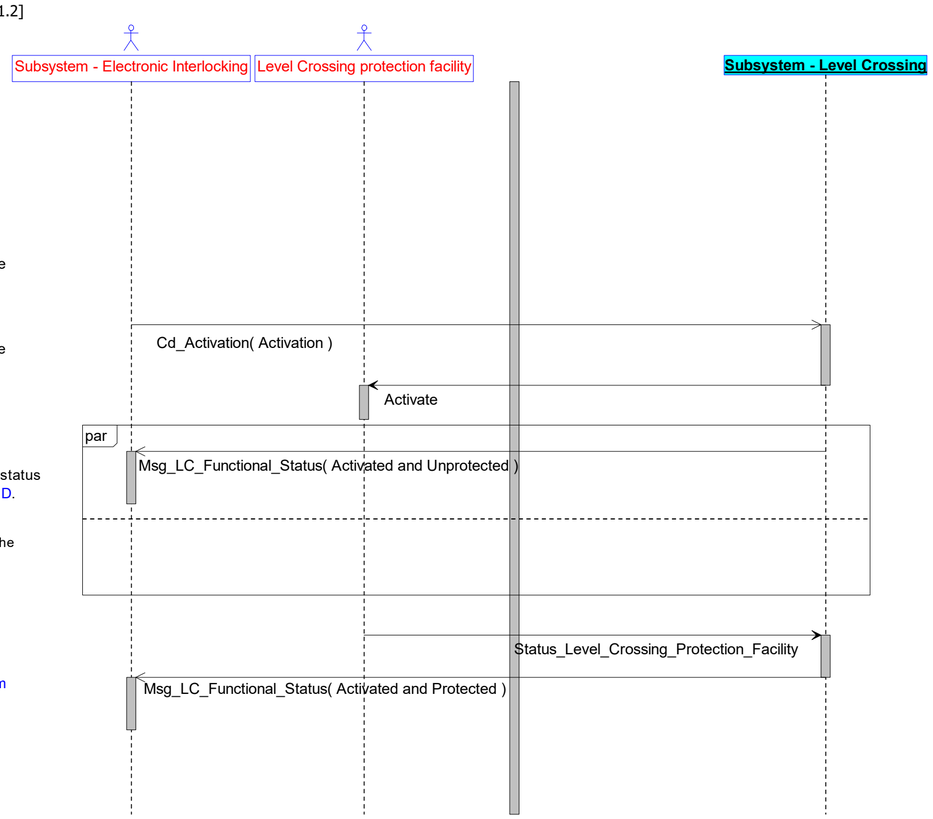
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.61	Info	In the column “Requirement Part 1” the particular SysML model element is depicted and in the column “Requirement Part 2” the corresponding extension of the definition is given. The stated object type normally applies both to “Requirement Part 1” and to “Requirement Part 2”.		
Eu.LC.62	Info	There are requirements with type "Req" given, where the column "Requirement Part 2" or a part of it is provided with the heading "Information". In this case, the defined type only applies to the column "Requirement Part 1" and the part of "Requirement Part 2", which is not labelled as "Information".		
Eu.LC.3945	Info	State machines or several state machines linked together in a Functional Architecture define the totality of all functional requirements of an SUS or an SIUS in a coherent and consistent manner. State diagrams of a corresponding state machine are marked with the object type “Req”. For the later design and implementation, it is not the description language SysML that is binding, but the domain-specific meaning expressed by it. The specified behaviour can be converted into a vendor specific language but must retain the domain specific meaning describing the functional requirements. The specific model elements are additionally specified and defined by object type “Def” to allow for traceability to supplier designs or test cases. The compliance of products to the specifications must be demonstrated by testing against EULYNX test cases, which are derived from the functionality specified by the models.		
Eu.LC.63	Head	**2 Conditions of use**		
Eu.LC.2371	Req	All references to [Eu.Doc.20] refer to version 4.0 (6.A) of that document.		
Eu.LC.3266	Req	All references to [Eu.Doc.119] refer to version 1.1 (0.A) of that document.		
Eu.LC.3267	Info	References to [Eu.Doc.120] do not refer to a concrete version of that document. The applicable version shall be defined by national specifications. Note: In future phases of the System Pillar, national specifications will be replaced by harmonised specifications.		
Eu.LC.64	Info	The specifications defined in this document shall follow the requirements of the EULYNX System Architecture Specification [Eu.Doc.16].		
Eu.LC.3173	Head	**2.1 Functional packages**		
Eu.LC.3174	Info	The specifications in this document are divided into functional packages. There are two types of packages related to the product capabilities.		
Eu.LC.3175	Info	‘Basic packages’: One or more packages, at least one of them must be implemented. It is allowed to combine and implement more than one ‘basic package’ in a product.		
Eu.LC.3176	Info	‘Optional package’: One or more packages that can be optionally implemented in addition to one or more basic packages.		
Eu.LC.3177	Info	The specifications of the Subsystem – Level Crossing are divided into the following functional packages:		
Eu.LC.3178	Info	Basic Level Crossing functionality (basic package) [Basic LC]		
Eu.LC.3179	Info	Interface for local operator handover (optional package) [Option LOH]		
Eu.LC.66	Head	**3 Functional requirements specification**		
Eu.LC.2588	Head	**3.1 Subsystem - Level Crossing - General Infos and Assumptions**		
Eu.LC.4148	Info	The defined model elements represent the Subsystem - Level Crossing in a general way. This refers to: -The defined number of monitoring state information as reported by the Level Crossing protection facility to the Subsystem - Level Crossing is limited to Parameter Set 1 and Parameter Set 2.		Basic LC
Eu.LC.2775	Head	**3.2 Subsystem - Level Crossing - Logical Viewpoint**		
Eu.LC.3263	Head	**3.2.1 Subsystem - Level Crossing - Logical Context**		
Eu.LC.2776	Def	[Package] Subsystem - Level Crossing - Logical Context [Logical Viewpoint - Subsystem Definition]  		Basic LC
Eu.LC.4265	Req	The Subsystem - Level Crossing shall provide a logical interface SCI-LC to exactly one Subsystem - Electronic Interlocking.		Basic LC
Eu.LC.4267	Req	The Subsystem - Level Crossing shall provide a logical interface SMI-LC to exactly one Subsystem - Maintenance and Data Management.		Basic LC
Eu.LC.4268	Req	The Subsystem - Level Crossing shall provide a logical interface SDI-LC to exactly one Subsystem - Maintenance and Data Management.		Basic LC
Eu.LC.4269	Req	The Subsystem - Level Crossing shall provide a logical interface SSI-LC to exactly one Subsystem - Security Services Platform.		Basic LC

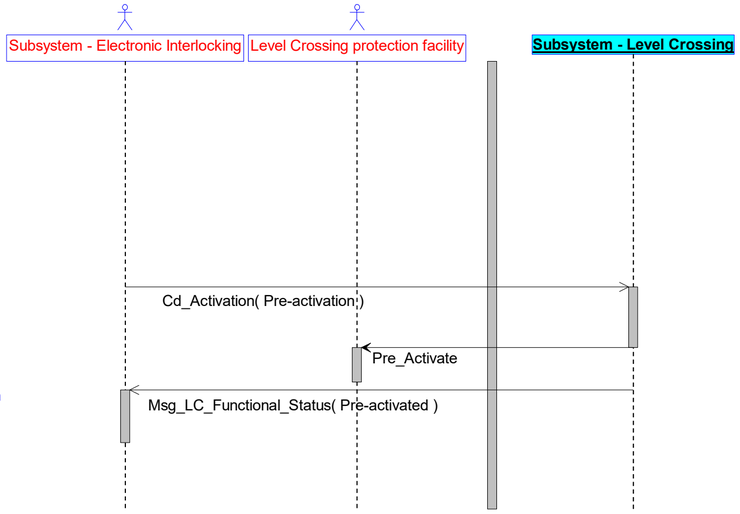
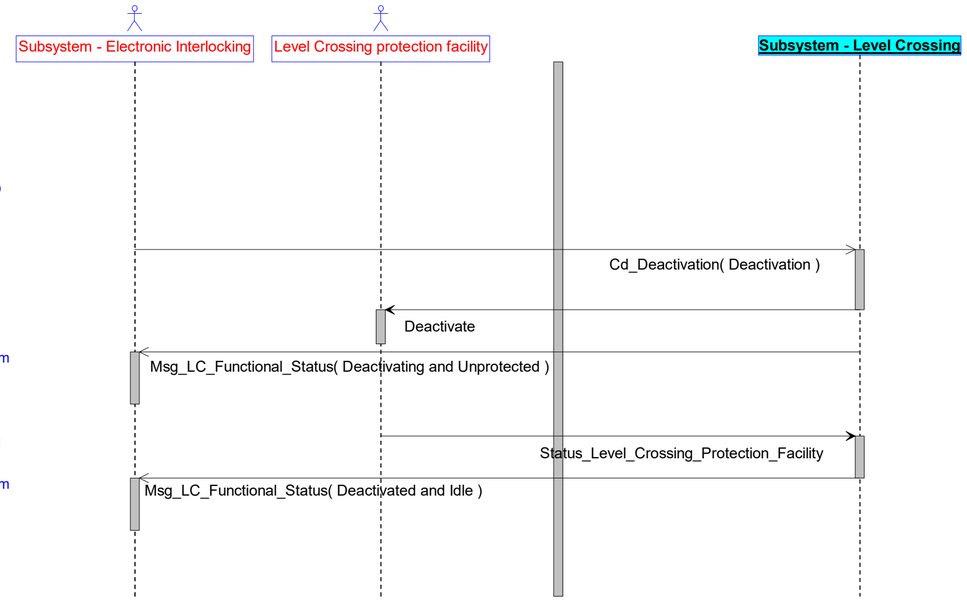
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.4270	Req	The Subsystem - Level Crossing shall provide a logical interface LC1 to exactly one Basic Data identifier.		Basic LC
Eu.LC.4271	Req	The Subsystem - Level Crossing shall provide a logical interface LC2 to exactly one Maintainer.		Basic LC
Eu.LC.4272	Req	The Subsystem - Level Crossing shall provide a logical interface LC3 to exactly one Power Supply.		Basic LC
Eu.LC.4273	Req	The Subsystem - Level Crossing shall provide a logical interface LC4 to exactly one Level Crossing protection facility.		Basic LC
Eu.LC.4274	Req	The Subsystem - Level Crossing shall provide a logical interface LC5 to each Detection element.		Basic LC
Eu.LC.4266	Req	The Subsystem - Level Crossing shall provide a logical interface LC6 to zero or one Local operator.		Option LOH
Eu.LC.2373	Head	**3.3 Subsystem - Level Crossing - Functional Viewpoint**		
Eu.LC.172	Head	**3.3.1 Definition of time values**		
Eu.LC.173	Info	The generic time values for SCI are specified in [Eu.Doc.119].		Basic LC
Eu.LC.3270	Info	The generic time values for SMI are specified in [Eu.Doc.120].		Basic LC
Eu.LC.175	Def	Con_tmax_Closure_Time	"Con_tmax_Closure_Time" defines the permissible duration for the activation of the level crossing. The time value is configurable and specific to a particular location.	Basic LC
Eu.LC.177	Def	Con_t_PDI_Loss_Deactivation_Delay	"Con_t_PDI_Loss_Deactivation_Delay" defines the delay to deactivate the Level Crossing protection facility after an activation caused by an interrupted Safe communication protocol connection. The time value is configurable and specific to a particular location.	Basic LC
Eu.LC.2376	Head	**3.3.2 Subsystem - Level Crossing - Functional Context**		
Eu.LC.189	Info	[Package] Subsystem - Level Crossing - Functional Context [Functional Viewpoint - Subsystem Definition - Initialisation]  		Basic LC
Eu.LC.188	Info	The generic UseCases SCI-XX EfeS IFUC1.1: Establish PDI connection and SCI-XX EfeS IFUC1.2: Close PDI connection are specified in [Eu.Doc.119]. The generic UseCases SMI-XX IFUC 1.1: Establish SMI connection, SMI-XX IFUC 1.2: Synchronous loading and activation of data, SMI-XX IFUC 1.3: Asynchronous preloading of data, SMI-XX IFUC 1.4: Reset EfeS and SMI-XX IFUC 1.5: Initiate maintenance are specified in [Eu.Doc.120].		Basic LC
Eu.LC.190	Info	LC_UC1.3: Report status	The Subsystem-UseCase LC_UC1.3: Report status defines a scenario about the transmission of status data of Subsystem - Level Crossing to Subsystem - Electronic Interlocking, while Process Data Interface protocol connection is establishing.	Basic LC

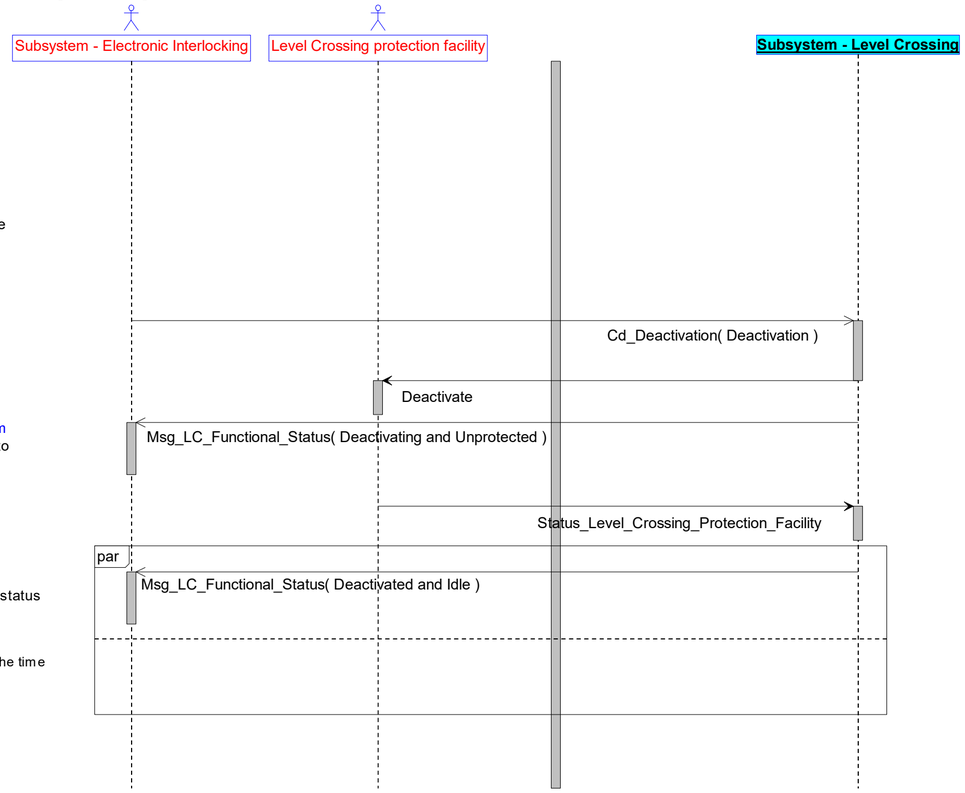
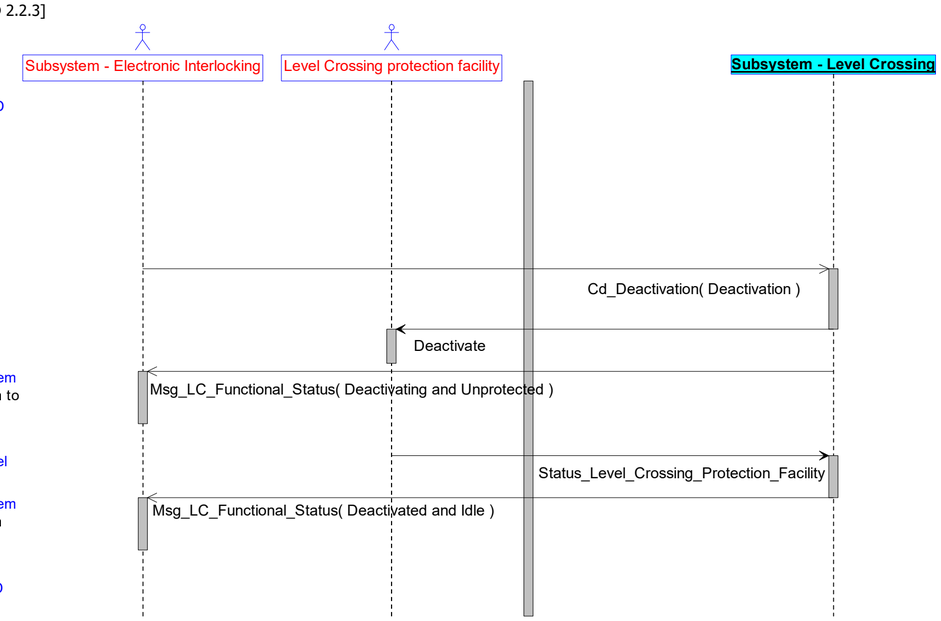
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.191	Info	Main Success Scenario: Report status [LC SD 1.3.1] **LC_UC1.3: Report status** Main Success Scenario: Report status [LC SD 1.3.1] **par** **1.a1** The Subsystem - Level Crossing reports the current functional status to the Subsystem - Electronic Interlocking. **also par** **1.b1** The Subsystem - Level Crossing reports the current monitoring status to the Subsystem - Electronic Interlocking. **also par** **1.c1** The Subsystem - Level Crossing reports the current failure status to the Subsystem - Electronic Interlocking. **also par** **1.d1** The Subsystem - Level Crossing reports the current status of the Detection element to the Subsystem - Electronic Interlocking. **also par** **1.e1** The Subsystem - Level Crossing reports the current status of the Obstacle detector to the Subsystem - Electronic Interlocking. **end par**  	This SD is part of [SCI-XX EfeS IF SD 1.1.1] in [Eu.Doc.119].	Basic LC
Eu.LC.212	Info	LC_UC1.4: Establish initial state of outputs	The Subsystem-UseCase LC_UC1.4: Establish initial state of outputs defines the main success scenario for establishing the initial state of outputs of the Subsystem - Level Crossing. While initialising, the Level Crossing protection facility will be activated by the Subsystem - Level Crossing.	Basic LC
Eu.LC.213	Info	Main Success Scenario: Establish initial state of outputs (case 1) with activation and without closure time [LC SD 1.4.1] **LC_UC1.4: Establish initial state of outputs** Main Success Scenario: Establish initial state of outputs (case 1) with activation and without closure time [LC SD 1.4.1] **Precondition:** The Subsystem - Level Crossing is in the state BOOTING or INITIALISING. The Initial State Of Outputs has not been established. The Subsystem - Level Crossing is configured to activate the Level Crossing protection facility as Initial State Of Outputs and after an interruption of the PDI connection. **Interaction 1.4.1.A:** **1.** The Subsystem - Level Crossing detects the readiness for establishing the Initial State Of Outputs. **2.** The Subsystem - Level Crossing activates the Level Crossing protection facility. **Postcondition:** The Subsystem - Level Crossing is activated. Initial State Of Outputs established.  		Basic LC

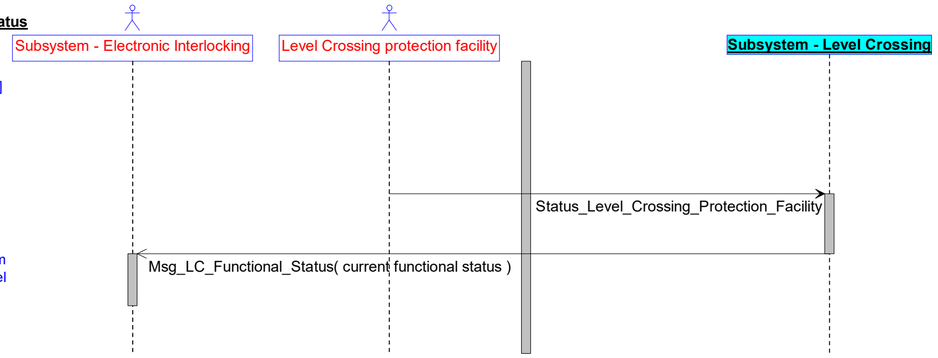
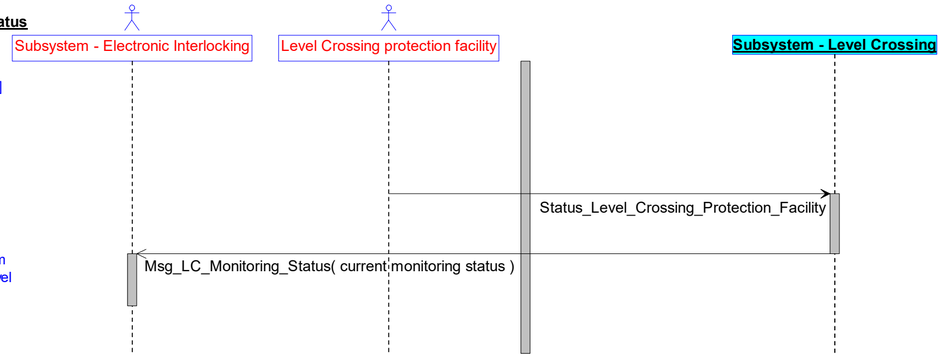
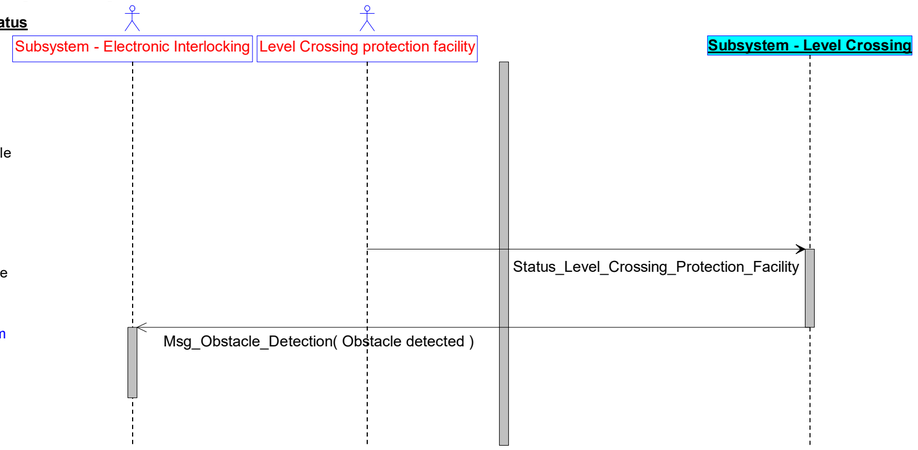
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.4133	Info	Main Success Scenario: Establish initial state of outputs (case 2) with activation and with closure time [LC SD 1.4.2] **LC_UC1.4: Establish initial state of outputs** Main Success Scenario: Establish initial state of outputs (case 2) with activation and with closure time [LC SD 1.4.2] **Precondition:** The Subsystem - Level Crossing is in the state BOOTING or INITIALISING. The Initial State Of Outputs has not been established. The Subsystem - Level Crossing is configured to activate the Level Crossing protection facility as Initial State Of Outputs and after an interruption of the PDI connection. The Subsystem - Level Crossing is configured to use Closure Time. **Interaction 1.4.2.A:** **1.** The Subsystem - Level Crossing detects the readiness for establishing the Initial State Of Outputs. **2.** The Subsystem - Level Crossing activates the Level Crossing protection facility. **3.** The Subsystem - Level Crossing starts to monitor the time period Con_tmax_Closure_Time. **Postcondition:** The Subsystem - Level Crossing is activated, with a currently monitored time value Con_tmax_Closure_Time. Initial State Of Outputs established.  		Basic LC
Eu.LC.4075	Info	Main Success Scenario: Establish initial state of outputs (case 3) with deactivation [LC SD 1.4.3] **LC_UC1.4: Establish initial state of outputs** Main Success Scenario: Establish initial state of outputs (case 3) with deactivation [LC SD 1.4.3] **Precondition:** The Subsystem - Level Crossing is in the state BOOTING or INITIALISING. The Initial State Of Outputs has not been established. The Subsystem - Level Crossing is configured to deactivate the Level Crossing protection facility as Initial State Of Outputs. **Interaction 1.4.3.A:** **1.** The Subsystem - Level Crossing detects the readiness for establishing the Initial State Of Outputs. **2.** The Subsystem - Level Crossing deactivates the Level Crossing protection facility. **Postcondition:** The Subsystem - Level Crossing is deactivated. Initial State Of Outputs established.  		Basic LC

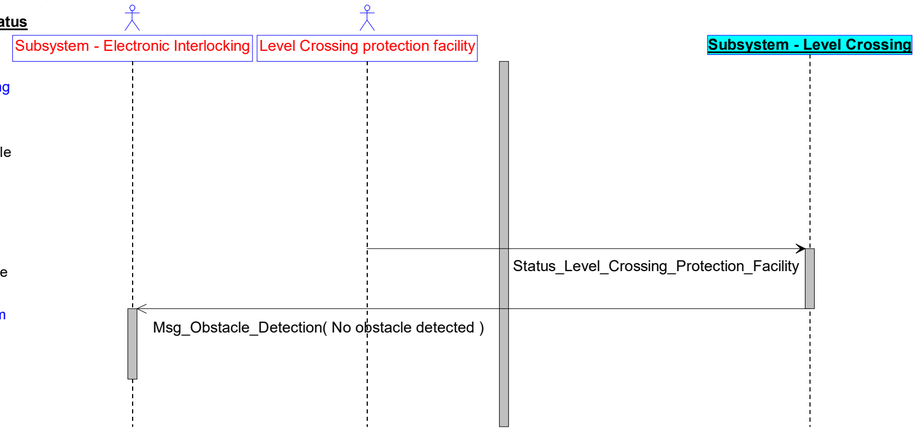
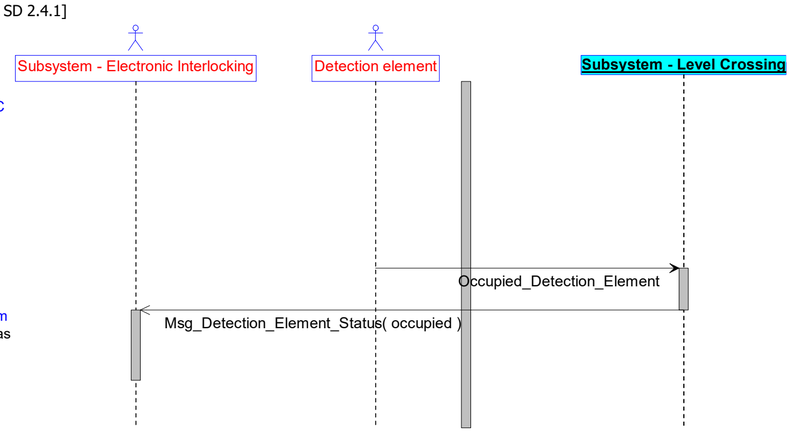
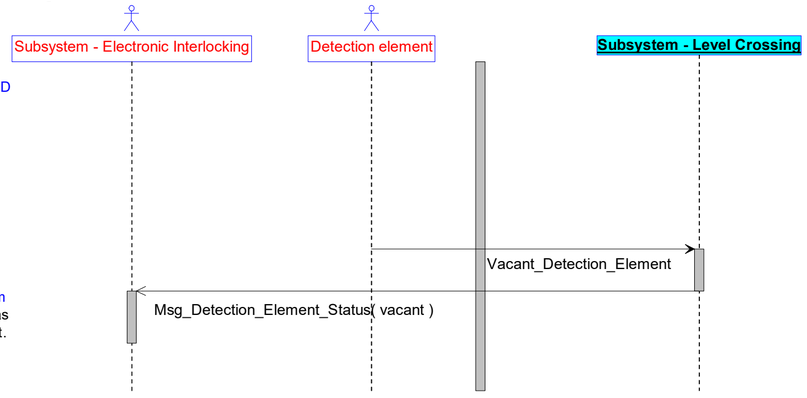
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.225	Info	[Package] Subsystem - Level Crossing - Functional Context [Functional Viewpoint - Subsystem Definition - Operation]  		Basic LC Option LOH
Eu.LC.227	Info	LC_UC2.1: Handle activation LC	The Subsystem-UseCase LC_UC2.1: Handle activation LC defines the activation of the Subsystem - Level Crossing for the interface functions. More detailed descriptions about the activation of the Level Crossing protection facility are subject to national requirements.	Basic LC

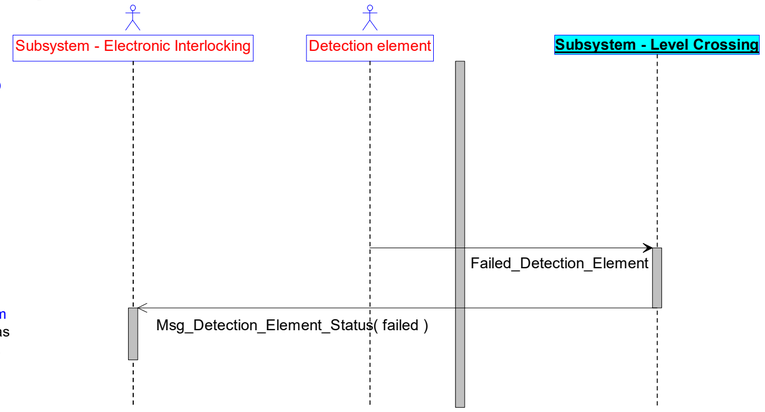
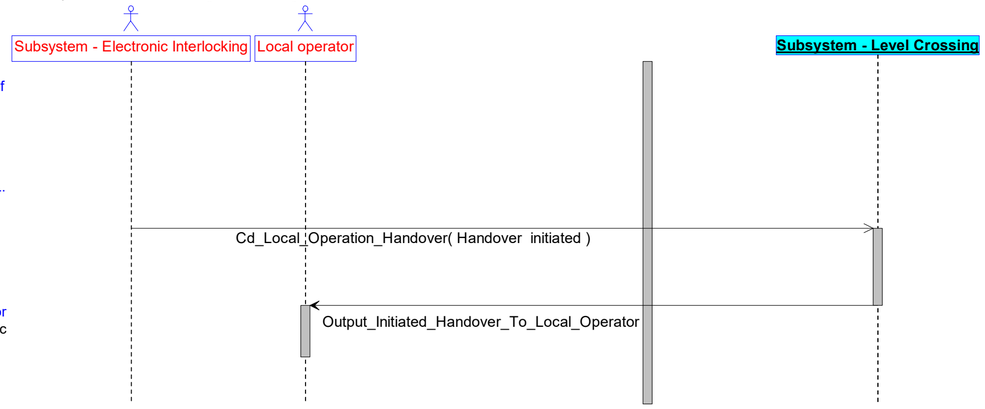
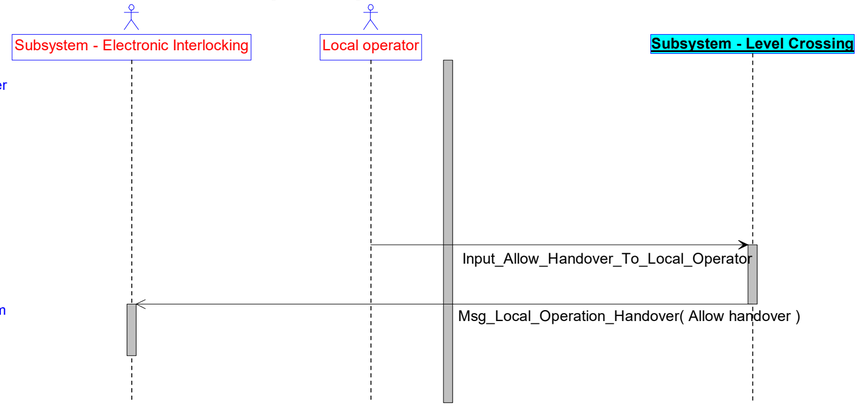
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.3963	Info	Main Success Scenario: Activate without closure time [LC SD 2.1.1] **LC_UC2.1: Handle activation LC** Main Success Scenario: Activate without closure time [LC SD 2.1.1] **Precondition:** The Subsystem - Level Crossing is in the state OPERATIONAL. The Subsystem - Level Crossing is either in the state DEACTIVATED AND IDLE or DEACTIVATING AND UNPROTECTED or PRE-ACTIVATED. **Interaction 2.1.1.A:** **1.** - The Subsystem - Level Crossing receives from the Subsystem - Electronic Interlocking the command to activate the Level Crossing protection facility. **2.** The Subsystem - Level Crossing activates the Level Crossing protection facility. **3.** The Subsystem - Level Crossing reports to the Subsystem - Electronic Interlocking that the functional status has been changed to ACTIVATED AND UNPROTECTED. **Interaction 2.1.1.B:** **4.** - The Subsystem - Level Crossing receives from the Level Crossing protection facility the new status protected. **5.** The Subsystem - Level Crossing reports to the Subsystem - Electronic Interlocking that the functional status has been changed to ACTIVATED AND PROTECTED. **Postcondition:** The Subsystem - Level Crossing is in the state ACTIVATED AND PROTECTED.  		Basic LC
Eu.LC.228	Info	Main Success Scenario: Activate with closure time [LC SD 2.1.2] **LC_UC2.1: Handle activation LC** Main Success Scenario: Activate with closure time [LC SD 2.1.2] **Precondition:** The Subsystem - Level Crossing is in the state OPERATIONAL. The Subsystem - Level Crossing is either in the state DEACTIVATED AND IDLE or DEACTIVATING AND UNPROTECTED or PRE-ACTIVATED. The Subsystem - Level Crossing is configured to use Closure Time. **Interaction 2.1.2.A:** **1.** - The Subsystem - Level Crossing receives from the Subsystem - Electronic Interlocking the command to activate the Level Crossing protection facility. **2.** The Subsystem - Level Crossing activates the Level Crossing protection facility. **par** **3.a1** The Subsystem - Level Crossing reports to the Subsystem - Electronic Interlocking that the functional status has been changed to ACTIVATED AND UNPROTECTED. **also par** **3.b1** The Subsystem - Level Crossing starts to monitor the time period Con_tmax_Closure_Time. **end par** **Interaction 2.1.2.B:** **4.** - The Subsystem - Level Crossing receives from the Level Crossing protection facility the new status protected. **5.** The Subsystem - Level Crossing reports to the Subsystem - Electronic Interlocking that the functional status has been changed to ACTIVATED AND PROTECTED. **Postcondition:** The Subsystem - Level Crossing is in the state ACTIVATED AND PROTECTED, with a currently monitored time value Con_tmax_Closure_Time.  		Basic LC

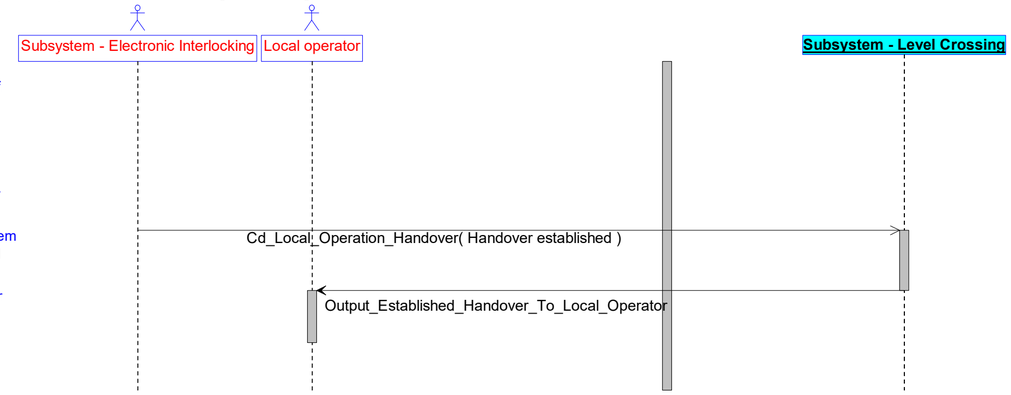
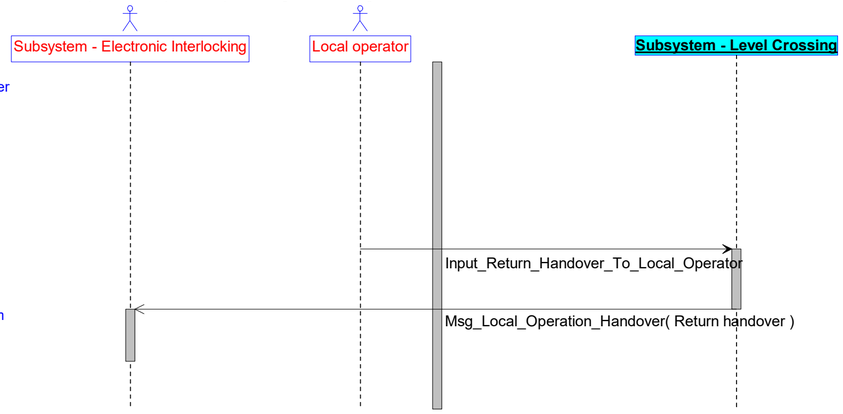
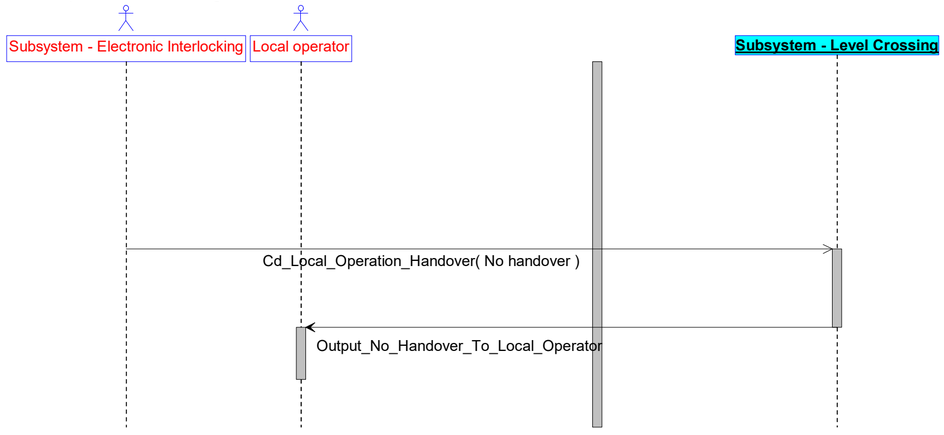
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.321	Info	Alternative Scenario: Pre-activate [LC SD 2.1.3] **LC_UC2.1: Handle activation LC** Alternative Scenario: Pre-activate [LC SD 2.1.3] **Precondition:** The Subsystem - Level Crossing is configured to use pre-activation. The Subsystem - Level Crossing is in the state OPERATIONAL. The Subsystem - Level Crossing is in the state DEACTIVATED AND IDLE or DEACTIVATING AND UNPROTECTED. **Interaction 2.1.3.A:** **1.** - The Subsystem - Level Crossing receives from the Subsystem - Electronic Interlocking the command to pre-activate the Level Crossing protection facility. **2.** The Subsystem - Level Crossing pre-activates the Level Crossing protection facility. **3.** The Subsystem - Level Crossing reports to the Subsystem - Electronic Interlocking that the functional status has been changed to PRE-ACTIVATED. **Postcondition:** The Subsystem - Level Crossing is in state PRE-ACTIVATED.  		Basic LC
Eu.LC.446	Info	LC_UC2.2: Handle deactivation LC	The Subsystem-UseCase LC_UC2.2: Handle deactivation LC defines the deactivation of the Subsystem - Level Crossing for the interface functions. More detailed descriptions about the deactivation of the Level Crossing protection facility are subject to national requirements.	Basic LC
Eu.LC.447	Info	Main Success Scenario: Deactivate after Activation without closure time [LC SD 2.2.1] **LC_UC2.2: Handle deactivation LC** Main Success Scenario: Deactivate after Activation without closure time [LC SD 2.2.1] **Precondition:** The Subsystem - Level Crossing is in the state OPERATIONAL. The Subsystem - Level Crossing is in the state ACTIVATED AND UNPROTECTED or ACTIVATED AND PROTECTED. **Interaction 2.2.1.A:** **1.** - The Subsystem - Level Crossing receives from the Subsystem - Electronic Interlocking the command to deactivate the Level Crossing protection facility. **2.** The Subsystem - Level Crossing deactivates the Level Crossing protection facility. **3.** The Subsystem - Level Crossing reports to the Subsystem - Electronic Interlocking that the functional status has been changed to DEACTIVATING AND UNPROTECTED. **Interaction 2.2.1.B:** **4.** - The Subsystem - Level Crossing receives from the Level Crossing protection facility the new status idle. **5.** The Subsystem - Level Crossing reports to the Subsystem - Electronic Interlocking that the functional status has been changed to DEACTIVATED AND IDLE. **Postcondition:** The Subsystem - Level Crossing is in state DEACTIVATED AND IDLE.  		Basic LC

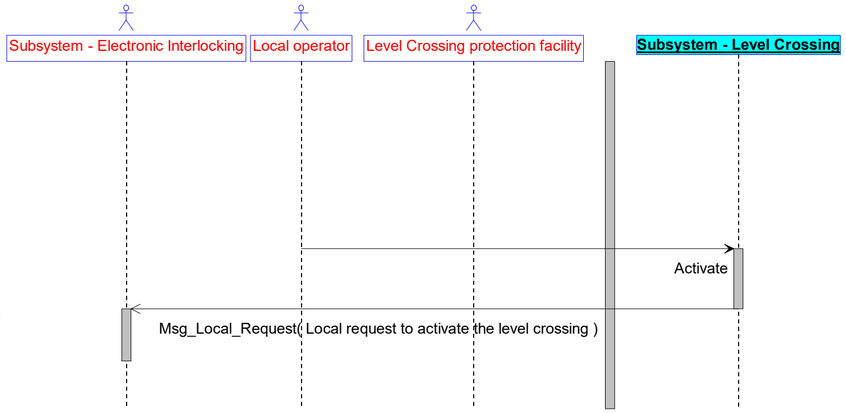
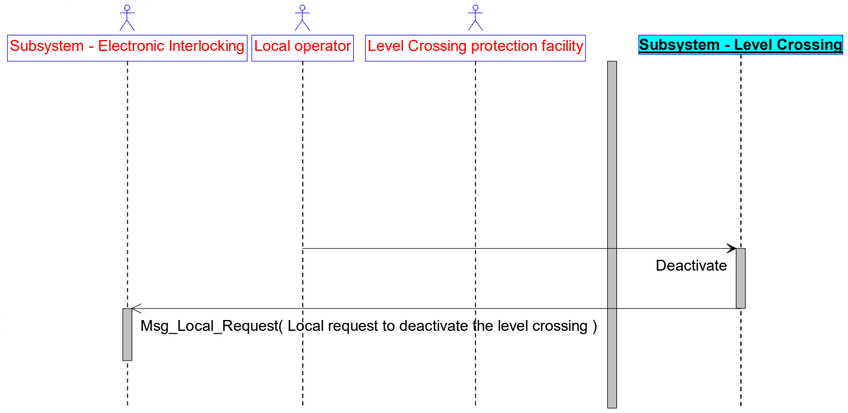
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.3964	Info	Main Success Scenario: Deactivate after Activation with closure time [LC SD 2.2.2] **LC_UC2.2: Handle deactivation LC** Main Success Scenario: Deactivate after Activation with closure time [LC SD 2.2.2] **Precondition:** The Subsystem - Level Crossing is in the state OPERATIONAL. The Subsystem - Level Crossing is in the state ACTIVATED AND UNPROTECTED or ACTIVATED AND PROTECTED. The Subsystem - Level Crossing is configured to use Closure Time. The Subsystem - Level Crossing is currently monitoring the time value Con_tmax_Closure_Time. **Interaction 2.2.2.A:** **1.** - The Subsystem - Level Crossing receives from the Subsystem - Electronic Interlocking the command to deactivate the Level Crossing protection facility. **2.** The Subsystem - Level Crossing deactivates the Level Crossing protection facility. **3.** The Subsystem - Level Crossing reports to the Subsystem - Electronic Interlocking that the functional status has been to DEACTIVATING AND UNPROTECTED. **Interaction 2.2.2.B:** **4.** - The Subsystem - Level Crossing receives from the Level Crossing protection facility the new status idle. **par** **5.a1** The Subsystem - Level Crossing reports to the Subsystem - Electronic Interlocking that the functional status has been changed to DEACTIVATED AND IDLE. **also par** **5.b1** The Subsystem - Level Crossing stops to monitor the time period Con_tmax_Closure_Time. **end par** **Postcondition:** The Subsystem - Level Crossing is in state DEACTIVATED AND IDLE.  		Basic LC
Eu.LC.3865	Info	Alternative Scenario: Deactivate after Pre-Activation [LC SD 2.2.3] **LC_UC2.2: Handle deactivation LC** Alternative Scenario: Deactivate after Pre-Activation [LC SD 2.2.3] **Precondition:** The Subsystem - Level Crossing is in the state OPERATIONAL. The Subsystem - Level Crossing is in the state PRE-ACTIVATED. **Interaction 2.2.3.A:** **1.** - The Subsystem - Level Crossing receives from the Subsystem - Electronic Interlocking the command to deactivate the Level Crossing protection facility. **2.** The Subsystem - Level Crossing deactivates the Level Crossing protection facility. **3.** The Subsystem - Level Crossing reports to the Subsystem - Electronic Interlocking that the functional status has been to DEACTIVATING AND UNPROTECTED. **Interaction 2.2.3.B:** **4.** - The Subsystem - Level Crossing receives from the Level Crossing protection facility the new status idle. **5.** The Subsystem - Level Crossing reports to the Subsystem - Electronic Interlocking that the functional status has been changed to DEACTIVATED AND IDLE. **Postcondition:** The Subsystem - Level Crossing is in state DEACTIVATED AND IDLE.  		
Eu.LC.570	Info	LC_UC2.3: Report Level Crossing protection facility Status	The Subsystem-UseCase LC_UC2.3: Report Level Crossing protection facility Status defines the report of a changed status the Subsystem - Level Crossing detected. For example if the protection status of the Level Crossing protection facility is reached.	Basic LC

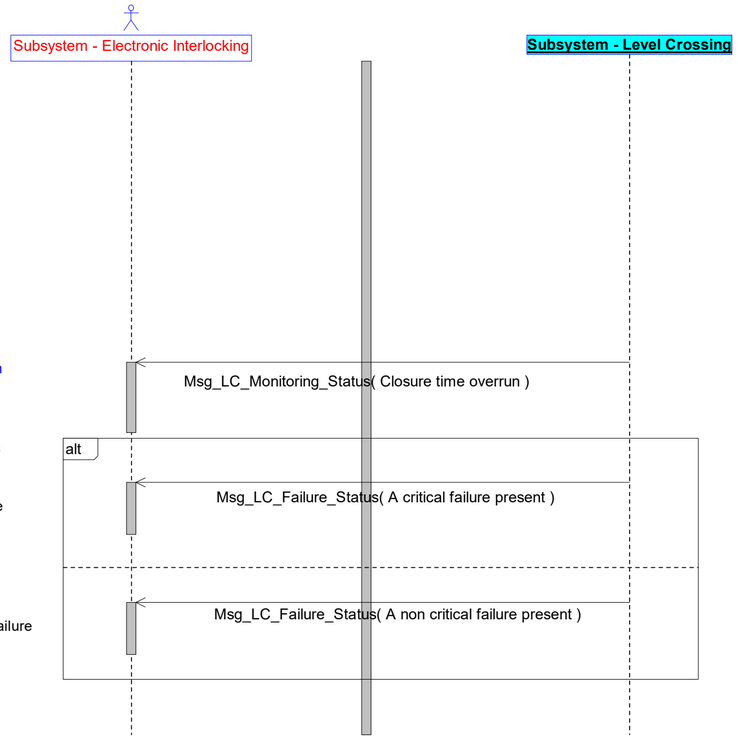
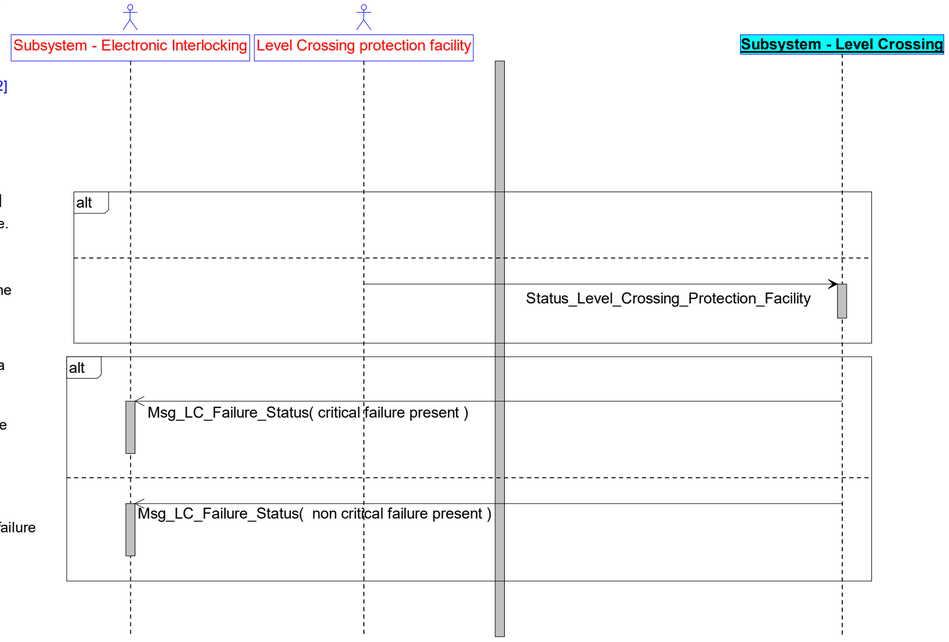
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.1327	Info	Alternative Scenario: Report Functional Status [LC SD 2.3.1] **LC_UC2.3: Report Level Crossing protection facility Status** Alternative Scenario: Report Functional Status [LC SD 2.3.1] **Precondition:** The Subsystem - Level Crossing is in the state OPERATIONAL. **Interaction 2.3.1.A:** **1.** - The Subsystem - Level Crossing receives from the Level Crossing protection facility a change in the functional parameters of the Level Crossing protection facility. **2.** The Subsystem - Level Crossing reports to the Subsystem - Electronic Interlocking the new functional status of the Level Crossing protection facility. **Postcondition:** ---  		Basic LC
Eu.LC.571	Info	Alternative Scenario: Report Monitoring Status [LC SD 2.3.2] **LC_UC2.3: Report Level Crossing protection facility Status** Alternative Scenario: Report Monitoring Status [LC SD 2.3.2] **Precondition:** The Subsystem - Level Crossing is in the state OPERATIONAL. **Interaction 2.3.2.A:** **1.** - The Subsystem - Level Crossing receives from the Level Crossing protection facility a change in the monitoring parameters of the Level Crossing protection facility. **2.** The Subsystem - Level Crossing reports to the Subsystem - Electronic Interlocking the new monitoring status of the Level Crossing protection facility. **Postcondition:** ---  		Basic LC
Eu.LC.604	Info	Alternative Scenario: Detect obstacle in the Level Crossing area [LC SD 2.3.3] **LC_UC2.3: Report Level Crossing protection facility Status** Alternative Scenario: Detect obstacle in the Level Crossing area [LC SD 2.3.3] **Precondition:** The Subsystem - Level Crossing is configured to use obstacle detection. The Subsystem - Level Crossing is in the state OPERATIONAL. **Interaction 2.3.3.A:** **1.** - The Subsystem - Level Crossing receives from the Level Crossing protection facility that an obstacle is detected in the Level Crossing area. The conditions for activation are not fulfilled. **2.** The Subsystem - Level Crossing reports to the Subsystem - Electronic Interlocking that the obstacle detection status has been changed. The status includes the information that an obstacle is detected in the Level Crossing area. **Postcondition:** ---  	The Level Crossing protection facility monitors whether the Obstacle detector shall be activated or not.	Basic LC

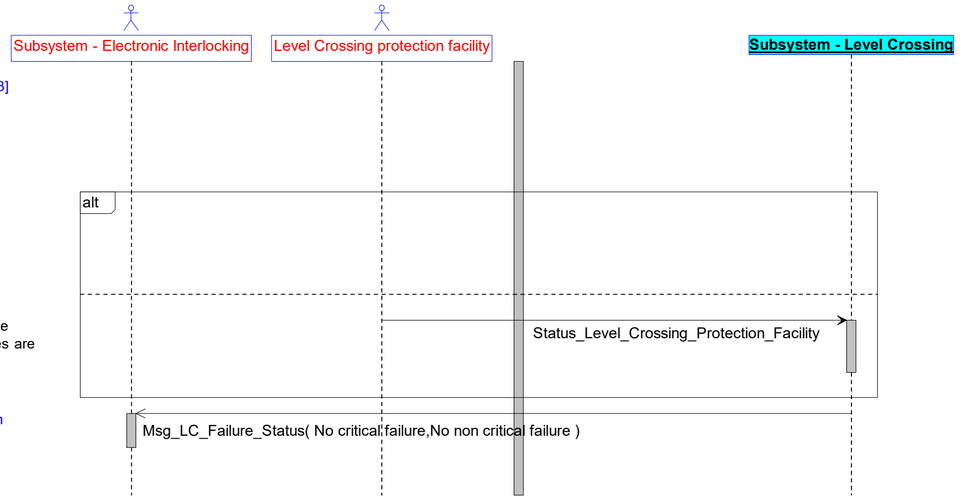
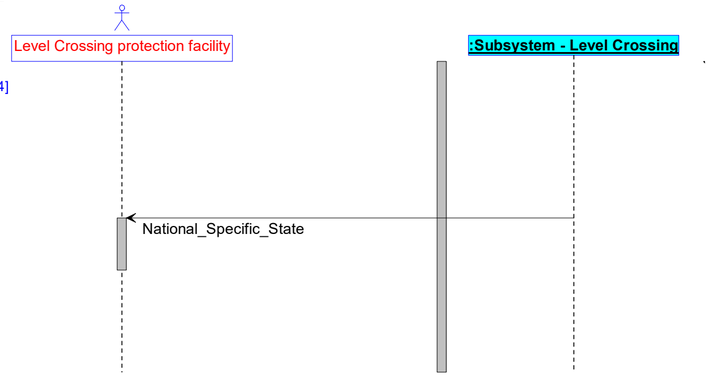
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.3304	Info	Alternative Scenario: Detect no obstacle in the Level Crossing area [LC SD 2.3.4] **LC_UC2.3: Report Level Crossing protection facility Status** Alternative Scenario: Detect no obstacle in the Level Crossing area [LC SD 2.3.4] **Precondition:** The Subsystem - Level Crossing is configured to use obstacle detection. The Subsystem - Level Crossing is in the state OPERATIONAL. **Interaction 2.3.4.A:** **1.** - The Subsystem - Level Crossing receives from the Level Crossing protection facility that no obstacle is detected in the Level Crossing area. **2.** The Subsystem - Level Crossing reports to the Subsystem - Electronic Interlocking that the obstacle detection status has been changed. The status includes the information that no obstacle is detected in the Level Crossing area. **Postcondition:** ---  	The Level Crossing protection facility monitors whether the Obstacle detector shall be activated or not.	Basic LC
Eu.LC.615	Info	LC_UC2.4: Report Detection Element Status	The Subsystem-UseCase LC_UC2.4: Report Detection Element Status defines reporting of the statuses of the Detection element to the Subsystem - Level Crossing.	Basic LC
Eu.LC.616	Info	Alternative Scenario: Report occupied Detection element [LC SD 2.4.1] **LC_UC2.4: Report Detection Element Status** Alternative Scenario: Report occupied Detection element [LC SD 2.4.1] **Precondition:** The Subsystem - Level Crossing is in state the OPERATIONAL. The Subsystem - Level Crossing is configured to use the Detection element. **Interaction 2.4.1.A:** **1.** - The Subsystem - Level Crossing detects that the Detection element is occupied. **2.** The Subsystem - Level Crossing reports to the Subsystem - Electronic Interlocking that the detection element status has been changed. The status of the Detection element is occupied. **Postcondition:** ---  		Basic LC
Eu.LC.625	Info	Alternative Scenario: Report vacant Detection element [LC SD 2.4.2] **LC_UC2.4: Report Detection Element Status** Alternative Scenario: Report vacant Detection element [LC SD 2.4.2] **Precondition:** The Subsystem - Level Crossing is in state the OPERATIONAL. The Subsystem - Level Crossing is configured to use the Detection element. **Interaction 2.4.2.A:** **1.** - The Subsystem - Level Crossing detects that the Detection element is vacant. **2.** The Subsystem - Level Crossing reports to the Subsystem - Electronic Interlocking that the detection element status has been changed. The status of the Detection element is vacant. **Postcondition:** ---  		Basic LC

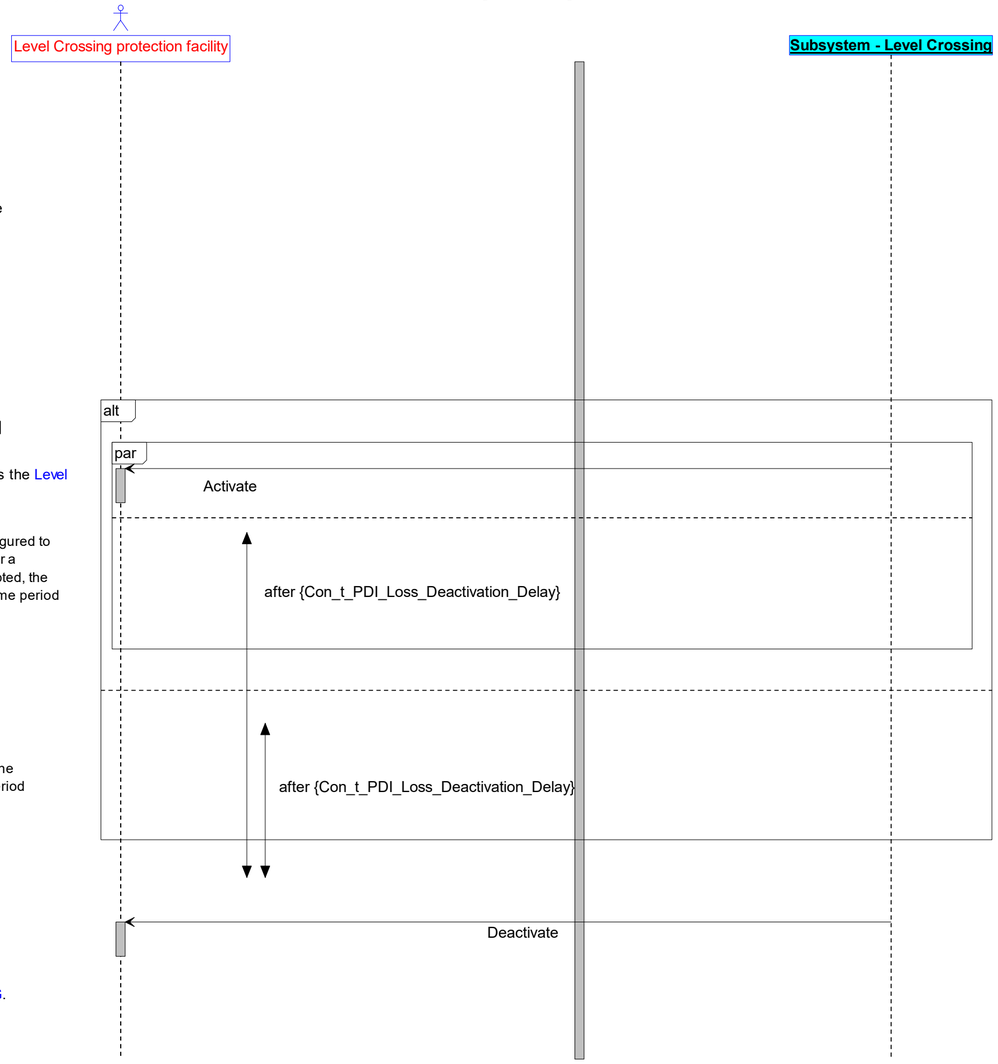
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.634	Info	Alternative Scenario: Report failed detection element [LC SD 2.4.3] **LC_UC2.4: Report Detection Element Status** Alternative Scenario: Report failed detection element [LC SD 2.4.3] **Precondition:** The Subsystem - Level Crossing is in the state OPERATIONAL. The Subsystem - Level Crossing is configured to use the Detection element. **Interaction 2.4.3.A:** **1.** - The Subsystem - Level Crossing detects a technical failure for the detection element related deactivation of the Level Crossing protection facility. **2.** The Subsystem - Level Crossing reports to the Subsystem - Electronic Interlocking that the detection element status has been changed. The status of the Detection element is failed. **Postcondition:** ---  		Basic LC
Eu.LC.643	Info	LC_UC2.5: Handle Local operations	The Subsystem-UseCase LC_UC2.5: Handle Local operations defines the handle of a request to activate or deactivate the Level Crossing protection facility from the Local operator to the Subsystem - Level Crossing.	Option LOH
Eu.LC.644	Info	Alternative Scenario: Receiving a command that the handover of local operations is initiated [LC SD 2.5.1] **LC_UC2.5: Handle Local operations** Alternative Scenario: Receiving a command that the handover of local operations is initiated [LC SD 2.5.1] **Precondition:** The Subsystem - Level Crossing is configured to use Local Operation. The Subsystem - Level Crossing is in the state OPERATIONAL. **Interaction 2.5.1.A:** **1.** - The Subsystem - Level Crossing receives from the Subsystem - Electronic Interlocking the command that the handover of local operations for a specific index (e.g. track) is initiated. **2.** The Subsystem - Level Crossing reports to the Local operator that the handover of the local operations is initiated for a specific index (e.g. track). **Postcondition:** ---  		Option LOH
Eu.LC.653	Info	Alternative Scenario: Receiving an input to allow the handover of local operations by Local operator [LC SD 2.5.2] **LC_UC2.5: Handle Local operations** Alternative Scenario: Receiving an input to allow the handover of local operations by Local operator [LC SD 2.5.2] **Precondition:** The Subsystem - Level Crossing is configured to use Local Operation. The Subsystem - Level Crossing is in the state OPERATIONAL. **Interaction 2.5.2.A:** **1.** - The Subsystem - Level Crossing receives the input from the Local operator to allow handover of local operation for a specific index (e.g. track). **2.** The Subsystem - Level Crossing reports to the Subsystem - Electronic Interlocking that the Local operator has allowed the handover of local operations. **Postcondition:** ---  		Option LOH

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.664	Info	Alternative Scenario: Receiving a command that the handover of local operations is established [LC SD 2.5.3] **LC_UC2.5: Handle Local operations** Alternative Scenario: Receiving a command that the handover of local operations is established [LC SD 2.5.3] **Precondition:** The Subsystem - Level Crossing is configured to use Local Operation. The Subsystem - Level Crossing is in the state OPERATIONAL. **Interaction 2.5.3.A:** **1.** - The Subsystem - Level Crossing receives from the Subsystem - Electronic Interlocking the command that the handover of local operations for a specific index (e.g. track) is established. **2.** The Subsystem - Level Crossing reports to the Local operator that the handover of the local operations is established for a specific index (e.g. track). **Postcondition:** ---  		Option LOH
Eu.LC.673	Info	Alternative Scenario: Receiving an input to return the handover of local operations by Local operator [LC SD 2.5.4] **LC_UC2.5: Handle Local operations** Alternative Scenario: Receiving an input to return the handover of local operations by Local operator [LC SD 2.5.4] **Precondition:** The Subsystem - Level Crossing is configured to use Local Operation. The Subsystem - Level Crossing is in the state OPERATIONAL. **Interaction 2.5.4.A:** **1.** - The Subsystem - Level Crossing receives the input from the Local operator to return handover of the local operations for a specific index (e.g. track). **2.** The Subsystem - Level Crossing reports to the Subsystem - Electronic Interlocking that the Local operator has returned the handover of local operations. **Postcondition:** ---  		Option LOH
Eu.LC.704	Info	Alternative Scenario: Receiving a command that the handover of local operations is returned [LC SD 2.5.5] **LC_UC2.5: Handle Local operations** Alternative Scenario: Receiving a command that the handover of local operations is returned [LC SD 2.5.5] **Precondition:** The Subsystem - Level Crossing is configured to use Local Operation. The Subsystem - Level Crossing is in the state OPERATIONAL. **Interaction 2.5.5.A:** **1.** - The Subsystem - Level Crossing receives from the Subsystem - Electronic Interlocking the command that the handover of local operations for a specific index (e.g. track) is returned. **2.** The Subsystem - Level Crossing reports to the Local operator that the handover of the local operations is returned for a specific index (e.g. track). **Postcondition:** ---  		Option LOH

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.741	Info	Alternative Scenario: Activate the level crossing via local operations [LC SD 2.5.6] **LC_UC2.5: Handle Local operations** Alternative Scenario: Activate the level crossing via local operations [LC SD 2.5.6] **Precondition:** The Subsystem - Level Crossing is configured to use Local Operation. The Subsystem - Level Crossing is in the state OPERATIONAL. **Interaction 2.5.6.A:** **1.** - The Subsystem - Level Crossing detects a request for an activation from the Local operator for activating the Level Crossing protection facility. **2.** The Subsystem - Level Crossing reports to the Subsystem - Electronic Interlocking that a local operator has requested a local activation. **Postcondition:** ---  	Note: If the conditions for a local operation are fulfilled the Main Success Scenario: Activate LC SD 2.1.1 will be started. This SD is part of [LC SD 2.1.1].	Option LOH
Eu.LC.753	Info	Alternative Scenario: Deactivate the level crossing via local operations [LC SD 2.5.7] **LC_UC2.5: Handle Local operations** Alternative Scenario: Deactivate the level crossing via local operations [LC SD 2.5.7] **Precondition:** The Subsystem - Level Crossing is configured to use Local Operation. The Subsystem - Level Crossing is in the state OPERATIONAL. **Interaction 2.5.7.A:** **1.** - The Subsystem - Level Crossing detects a request for a deactivation from the Local operator for activating the Level Crossing protection facility. **2.** The Subsystem - Level Crossing reports to the Subsystem - Electronic Interlocking that a local operator has requested a local deactivation. **Postcondition:** ---  	Note: If the conditions for a local operation are fulfilled the Main Success Scenario: Deactivation LC SD 2.2.1 will be started. This SD is part of [LC SD 2.2.1].	Option LOH
Eu.LC.963	Info	LC_UC2.6: Handle irregularities	The Subsystem-UseCase LC_UC2.6: Handle irregularities defines the behaviour of the Subsystem - Level Crossing when an irregularity occurs.	Basic LC

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.964	Info	Alternative Scenario: Closure time value exceeded [LC SD 2.6.1] **LC_UC2.6: Handle irregularities** Alternative Scenario: Closure time value exceeded [LC SD 2.6.1] **Precondition:** The Subsystem - Level Crossing is configured to use the Closure Time. The Subsystem - Level Crossing is in the state OPERATIONAL. The Subsystem - Level Crossing is in the state ACTIVATED AND UNPROTECTED or ACTIVATED AND PROTECTED. The Subsystem - Level Crossing is currently monitoring the time value Con_tmax_Closure_Time. **Interaction 2.6.1.A:** **1.** - The Subsystem - Level Crossing detects that the time period Con_tmax_Closure_Time has exceeded. **2.** The Subsystem - Level Crossing reports to the Subsystem - Electronic Interlocking that the monitoring status has been changed. The status includes the information that Closure time has exceeded. **alt** [The Subsystem - Level Crossing is configured to report a critical failure after a Closure time value has exceeded] **3.a1** The Subsystem - Level Crossing reports to the Subsystem - Electronic Interlocking that a critical failure occurred. **else alt** [The Subsystem - Level Crossing is configured to report a non-critical failure after a Closure time value has exceeded] **3.a2** The Subsystem - Level Crossing reports to the Subsystem - Electronic Interlocking that a non-critical failure occurred. **end alt** **Postcondition:** ---  		Basic LC
Eu.LC.978	Info	Alternative Scenario: Report an occurred failure [LC SD 2.6.2] **LC_UC2.6: Handle irregularities** Alternative Scenario: Report an occurred failure [LC SD 2.6.2] **Precondition:** The Subsystem - Level Crossing is in the state OPERATIONAL. **Interaction 2.6.2.A:** **alt** [There is a failure inside the Subsystem - Level Crossing] **1.a1** - The Subsystem - Level Crossing detects a failure. **else alt** [There is a failure inside the Level Crossing protection facility] **1.b1** - The Subsystem - Level Crossing receives from the Level Crossing protection facility a detected failure. **end alt** **alt** [The Subsystem - Level Crossing is configured to report a critical failure after a failure was detected] **2.a1** The Subsystem - Level Crossing reports to the Subsystem - Electronic Interlocking that a critical failure occurred. **else alt** [The Subsystem - Level Crossing is configured to report a non-critical failure after a failure was detected] **2.b1** The Subsystem - Level Crossing reports to the Subsystem - Electronic Interlocking that a non-critical failure occurred. **end alt** **Postcondition:** ---  		Basic LC

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.1000	Info	Alternative Scenario: Report removed all failures [LC SD 2.6.3] **LC_UC2.6: Handle irregularities** Alternative Scenario: Report removed all failures [LC SD 2.6.3] **Precondition:** The Subsystem - Level Crossing is in the state OPERATIONAL. **Interaction 2.6.3.A:** **alt** [There was a failure inside the Subsystem - Level Crossing] **1.a1** - The Subsystem - Level Crossing detects that all occurred failures are removed. **else alt** [There was a failure inside the Level Crossing protection facility] **1.b1** - The Subsystem - Level Crossing receives from the Level Crossing protection facility that all occurred failures are removed. **end alt** **2.** The Subsystem - Level Crossing reports to the Subsystem - Electronic Interlocking that no failures are present. **Postcondition:** ---  		Basic LC
Eu.LC.1013	Info	Alternative Scenario: Perform fallback operation [LC SD 2.6.4] **LC_UC2.6: Handle irregularities** Alternative Scenario: Perform fallback operation [LC SD 2.6.4] **Precondition:** --- **Interaction 2.6.4.A:** **1.** - The Subsystem - Level Crossing enters the state FALLBACK_MODE. **2.** The Subsystem - Level Crossing requests the Level Crossing protection facility to change to the most safe national specific state. **Postcondition:** The Subsystem - Level Crossing is in the state FALLBACK_MODE. The Subsystem - Level Crossing is in the state according to the national requirements.  		Basic LC

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.1023	Info	Alternative Scenario: Handling of interrupted PDI connection (case 1) with activation followed by deactivation without closure time [LC SD 2.6.5] **LC_UC2.6: Handle irregularities** Alternative Scenario: Handling of interrupted PDI connection (case 1) with activation followed by deactivation without closure time [LC SD 2.6.5] **Precondition:** The Subsystem - Level Crossing is in the state OPERATIONAL. The Subsystem - Level Crossing is configured to activate the Level Crossing protection facility as Initial State Of Outputs and after an interruption of the PDI connection. The Subsystem - Level Crossing is configured to deactivate the Level Crossing protection facility after it was activated caused by an interrupted PDI connection. The Subsystem - Level Crossing is not in the state ISOLATED LC. **Interaction 2.6.5.A:** **1.** - The PDI connection has been terminated. **alt** [The Subsystem - Level Crossing is in state DEACTIVATING AND UNPROTECTED or PRE-ACTIVATED] **par** **2.a1.a1** The Subsystem - Level Crossing activates the Level Crossing protection facility. **also par** **2.a1.a2** If the Subsystem - Level Crossing is configured to deactivate the Level Crossing protection facility after a configured delay when the PDI connection is interrupted, the Subsystem - Level Crossing starts to monitor the time period Con_t_PDI_Loss_Deactivation_Delay. **end par** **else alt** [The Subsystem - Level Crossing is in state ACTIVATED AND PROTECTED or ACTIVATED AND UNPROTECTED] **2.b1** If the Subsystem - Level Crossing is configured to deactivate the Level Crossing protection facility after a configured delay when the PDI connection is interrupted, the Subsystem - Level Crossing starts to monitor the time period Con_t_PDI_Loss_Deactivation_Delay. **end alt** **Interaction 2.6.5.B:** **3.** - The Subsystem - Level Crossing detects that Con_t_PDI_Loss_Deactivation_Delay has expired. **4.** - The Subsystem - Level Crossing deactivates the Level Crossing protection facility. **Postcondition:** The Subsystem - Level Crossing is in the state INITIALISING. The PDI connection is terminated. The Subsystem - Level Crossing is in state DEACTIVATING AND UNPROTECTED.  	The following functionality remains available within the state INITIALISING after the termination of the PDI connection: • All functionality related to the maintainer interface LC2 (for example as in LC_UC3.4: Update specific software)	Basic LC

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.3965	Info	Alternative Scenario: Handling of interrupted PDI connection (case 2) with activation followed by deactivation with closure time [LC SD 2.6.6]	If the PDI connection is re-established before the expiration of Con_t_PDI_Loss_Deactivation_Delay, the monitoring of the delay stops. The following functionality remains available within the state INITIALISING after the termination of the PDI connection: • Reaction to expiration of Con_t_PDI_Loss_Deactivation_Delay (Alternative Scenario: Handling of interrupted PDI connection (case 1) with activation followed by deactivation without closure time [LC SD 2.6.5], Alternative Scenario: Handling of interrupted PDI connection (case 2) with activation followed by deactivation with closure time [LC SD 2.6.6]) • All functionality related to the local maintainer interface (LC_UC3.4: Update specific software)	Basic LC

LC_UC2.6: Handle irregularities

Alternative Scenario: Handling of interrupted PDI connection (case 2) with activation followed by deactivation with closure time [LC SD 2.6.6]

Precondition:

The Subsystem - Level Crossing is in the state OPERATIONAL.
The Subsystem - Level Crossing is configured to activate the Level Crossing protection facility as Initial State Of Outputs and after an interruption of the PDI connection.
The Subsystem - Level Crossing is configured to deactivate the Level Crossing protection facility after it was activated caused by an interrupted PDI connection.
The Subsystem - Level Crossing is not in the state ISOLATED LC.
The Subsystem - Level Crossing is configured to use the Closure Time.

Interaction 2.6.6.A:

1. - The PDI connection has been terminated.

alt [The Subsystem - Level Crossing is in state DEACTIVATING AND UNPROTECTED or PRE-ACTIVATED]

par

2.a1.a1 The Subsystem - Level Crossing activates the Level Crossing protection facility.

also par

2.a1.a2 If the Subsystem - Level Crossing is configured to deactivate the Level Crossing protection facility after a configured delay when the PDI connection is interrupted, the Subsystem - Level Crossing starts to monitor the time period Con_t_PDI_Loss_Deactivation_Delay.

also par

2.a1.a3 The Subsystem - Level Crossing starts to monitor the time period Con_tmax_Closure_Time.

end par

else alt [The Subsystem - Level Crossing is in state ACTIVATED AND PROTECTED or ACTIVATED AND UNPROTECTED]

2.b1 If the Subsystem - Level Crossing is configured to deactivate the Level Crossing protection facility after a configured delay when the PDI connection is interrupted, the Subsystem - Level Crossing starts to monitor the time period Con_t_PDI_Loss_Deactivation_Delay.

end alt

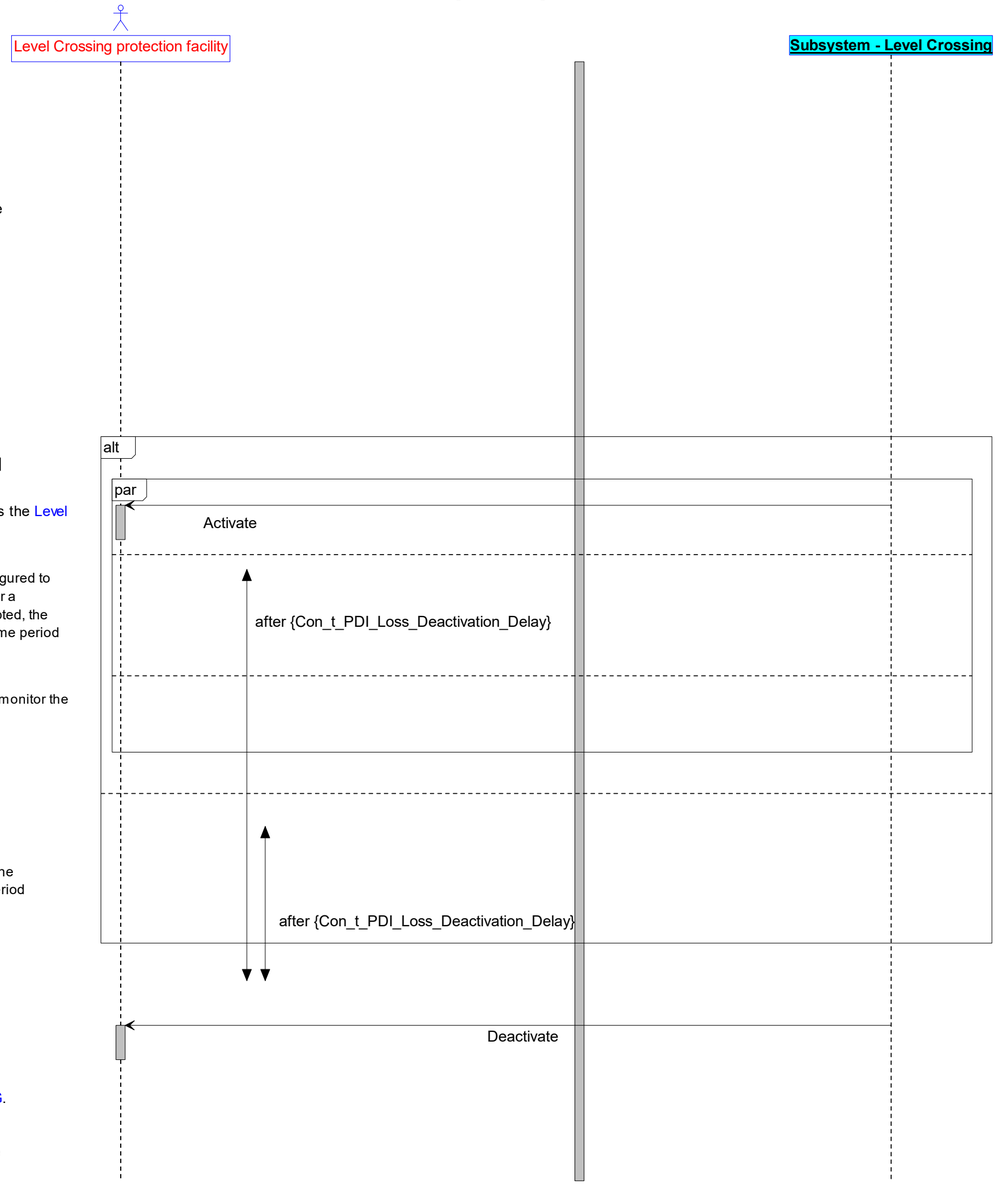
Interaction 2.6.6.B:

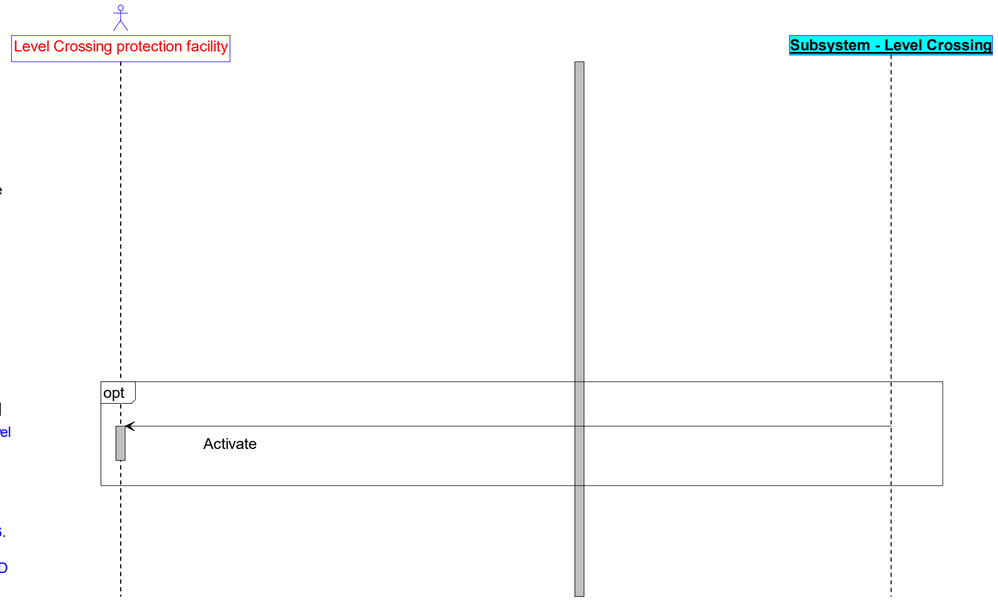
3. - The Subsystem - Level Crossing detects that Con_t_PDI_Loss_Deactivation_Delay has expired.

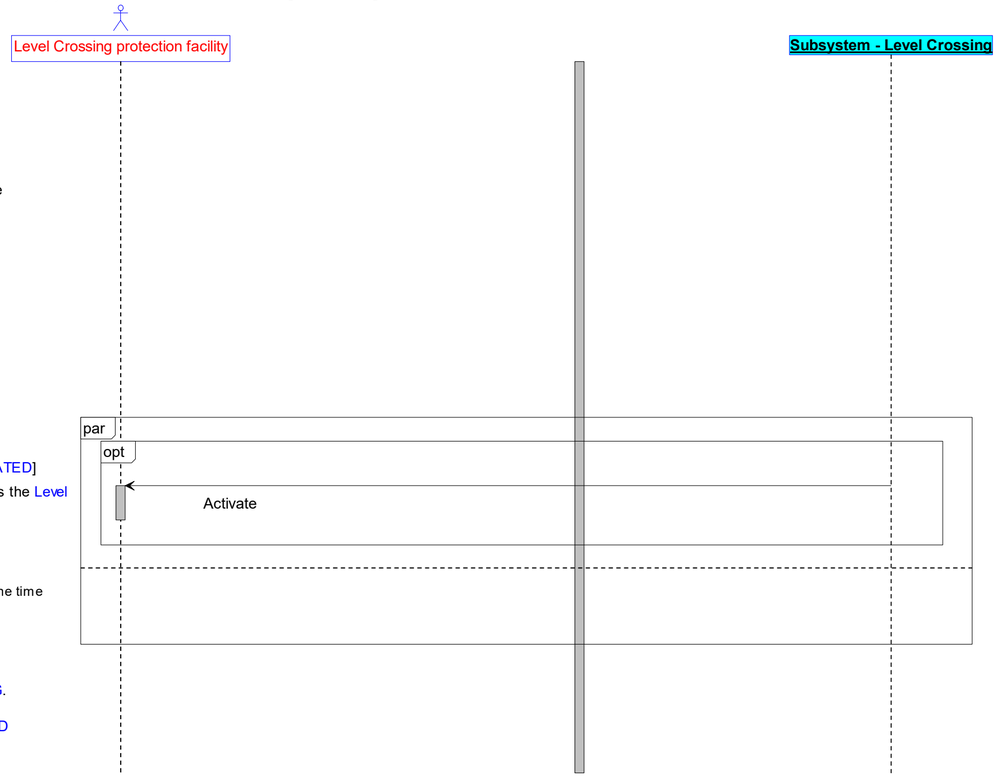
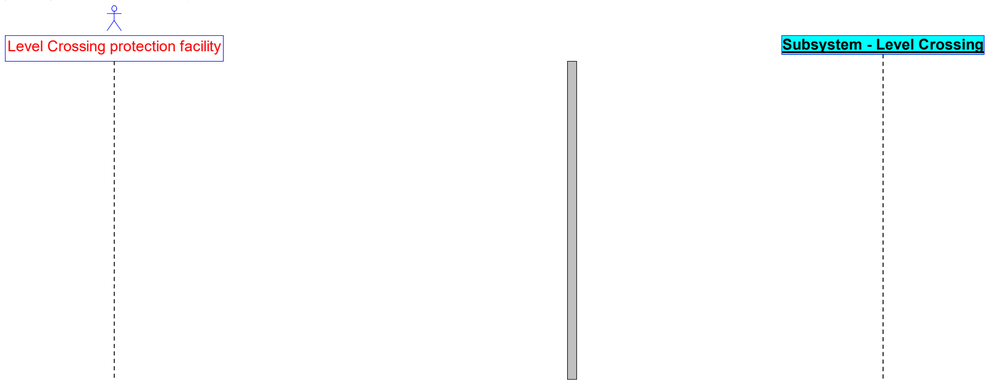
4. - The Subsystem - Level Crossing deactivates the Level Crossing protection facility.

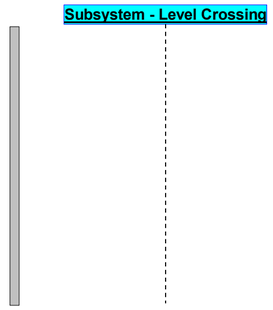
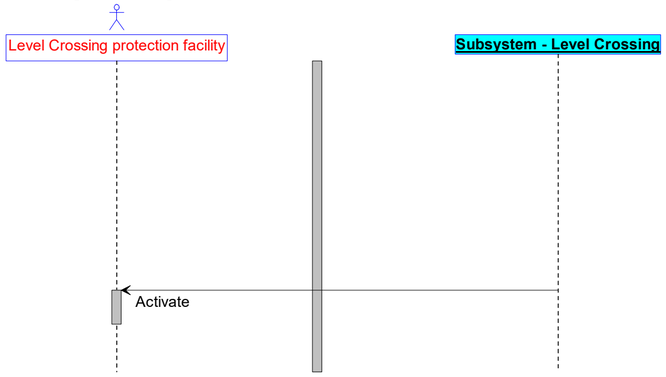
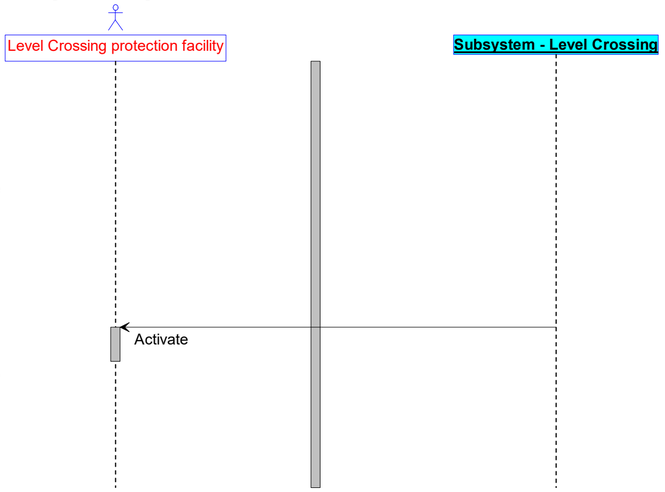
Postcondition:

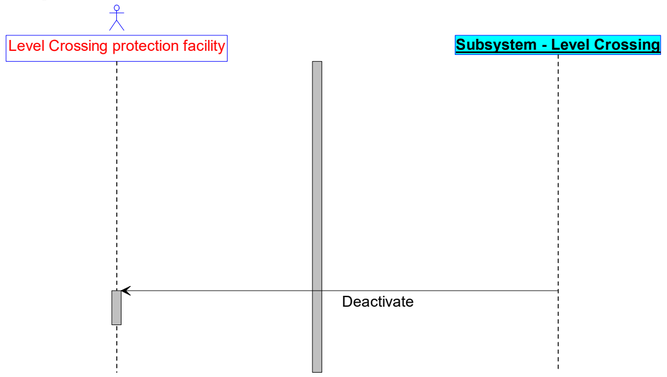
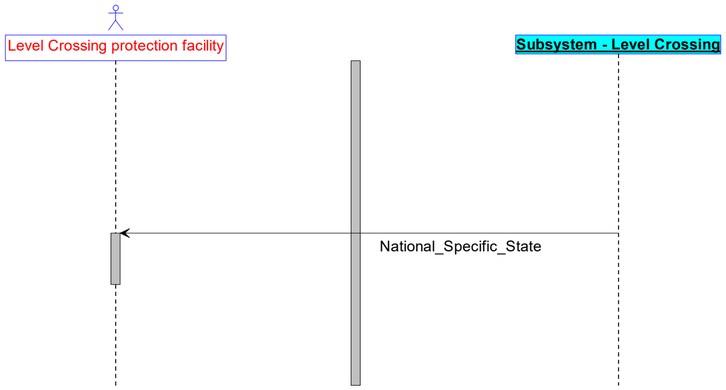
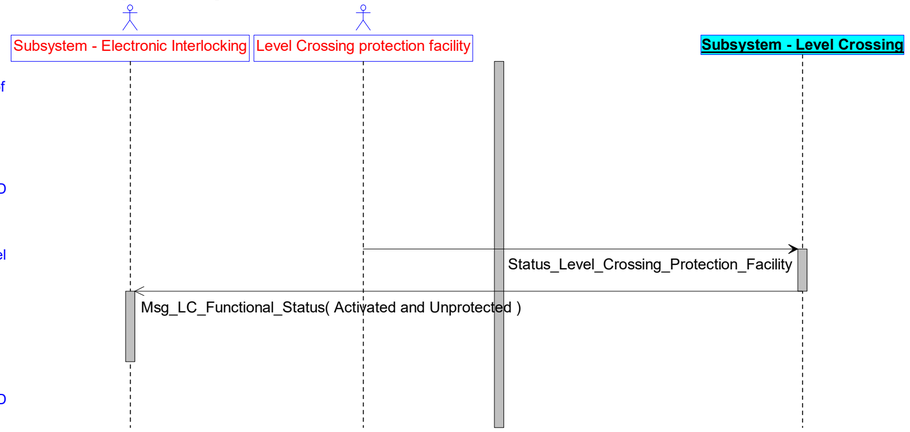
The Subsystem - Level Crossing is in the state INITIALISING.
The PDI connection is terminated.
The Subsystem - Level Crossing is in state DEACTIVATING AND UNPROTECTED, with a currently monitored time value Con_tmax_Closure_Time.

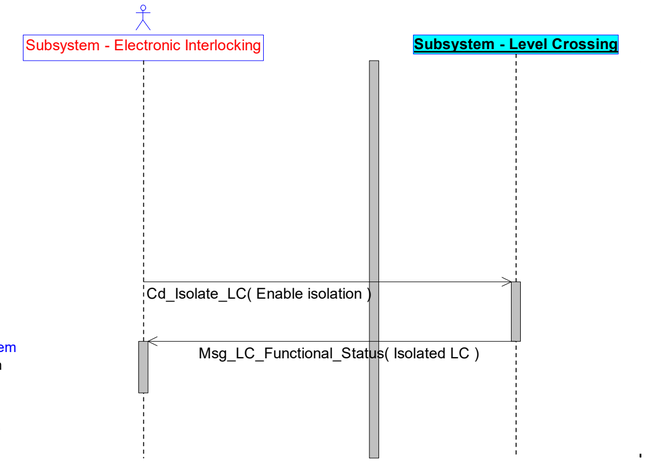
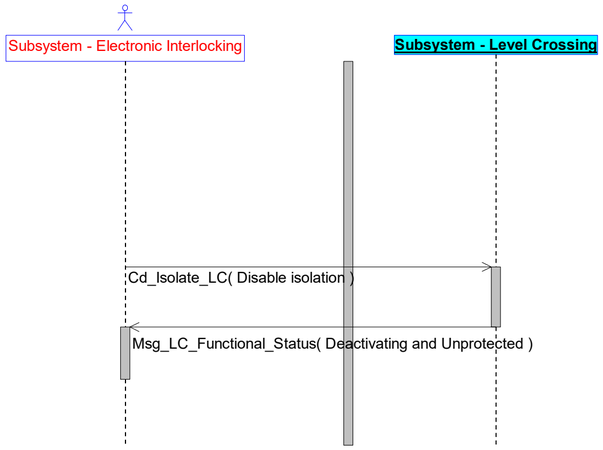


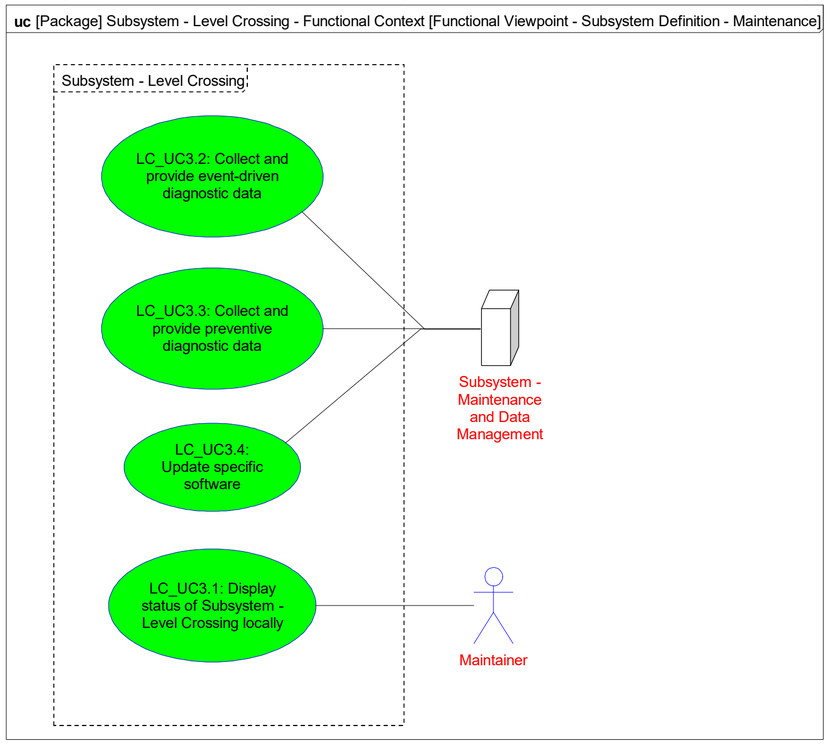
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.4079	Info	Alternative Scenario: Handling of interrupted PDI connection (case 3) with activation without closure time [LC SD 2.6.7] **LC_UC2.6: Handle irregularities** Alternative Scenario: Handling of interrupted PDI connection (case 3) with activation without closure time [LC SD 2.6.7] **Precondition:** The Subsystem - Level Crossing is in the state OPERATIONAL. The Subsystem - Level Crossing is configured to activate the Level Crossing protection facility as Initial State Of Outputs and after an interruption of the PDI connection. The Subsystem - Level Crossing is configured to not deactivate the Level Crossing protection facility after it was activated caused by an interrupted PDI connection. The Subsystem - Level Crossing is not in the state ISOLATED LC. **Interaction 2.6.7.A:** **1.** - The PDI connection has been terminated. **opt** [The Subsystem - Level Crossing is in state DEACTIVATING AND UNPROTECTED or PRE-ACTIVATED] **2.a1** The Subsystem - Level Crossing activates the Level Crossing protection facility. **end alt** **Postcondition:** The Subsystem - Level Crossing is in the state INITIALISING. The PDI connection is terminated. The Subsystem - Level Crossing is in state ACTIVATED AND UNPROTECTED.  	If the PDI connection is re-established before the expiration of Con_t_PDI_Loss_Deactivation_Delay, the monitoring of the delay stops. The following functionality remains available within the state INITIALISING after the termination of the PDI connection: • Reaction to expiration of Con_t_PDI_Loss_Deactivation_Delay (Alternative Scenario: Handling of interrupted PDI connection (case 1) with activation followed by deactivation without closure time [LC SD 2.6.5], Alternative Scenario: Handling of interrupted PDI connection (case 2) with activation followed by deactivation with closure time [LC SD 2.6.6]) • All functionality related to the local maintainer interface (LC_UC3.4: Update specific software)	Basic LC

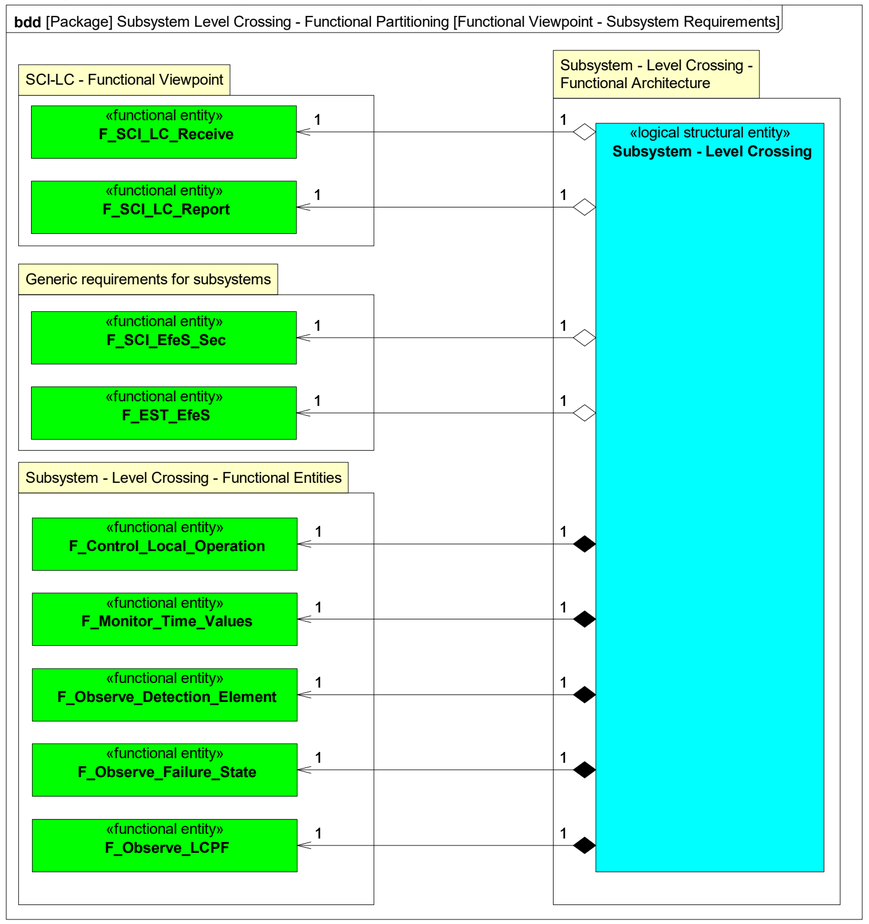
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.4080	Info	Alternative Scenario: Handling of interrupted PDI connection (case 4) with activation with closure time [LC SD 2.6.8] **LC_UC2.6: Handle irregularities** Alternative Scenario: Handling of interrupted PDI connection (case 4) with activation with closure time [LC SD 2.6.8] **Precondition:** The Subsystem - Level Crossing is in the state OPERATIONAL. The Subsystem - Level Crossing is configured to activate the Level Crossing protection facility as Initial State Of Outputs and after an interruption of the PDI connection. The Subsystem - Level Crossing is configured to not deactivate the Level Crossing protection facility after it was activated caused by an interrupted PDI connection. The Subsystem - Level Crossing is not in the state ISOLATED LC. The Subsystem - Level Crossing is configured to use the Closure Time. **Interaction 2.6.8.A:** **1.** - The PDI connection has been terminated. **par** **opt** [The Subsystem - Level Crossing is in state DEACTIVATING AND UNPROTECTED or PRE-ACTIVATED] **2.a1.a1** The Subsystem - Level Crossing activates the Level Crossing protection facility. **end alt** **also par** **2.b1** The Subsystem - Level Crossing starts to monitor the time period Con_tmax_Closure_Time. **end par** **Postcondition:** The Subsystem - Level Crossing is in the state INITIALISING. The PDI connection is terminated. The Subsystem - Level Crossing is in state ACTIVATED AND UNPROTECTED, with a currently monitored time value Con_tmax_Closure_Time.  	If the PDI connection is re-established before the expiration of Con_t_PDI_Loss_Deactivation_Delay, the monitoring of the delay stops. The following functionality remains available within the state INITIALISING after the termination of the PDI connection: • Reaction to expiration of Con_t_PDI_Loss_Deactivation_Delay (Alternative Scenario: Handling of interrupted PDI connection (case 1) with activation followed by deactivation without closure time [LC SD 2.6.5], Alternative Scenario: Handling of interrupted PDI connection (case 2) with activation followed by deactivation with closure time [LC SD 2.6.6]) • All functionality related to the local maintainer interface (LC_UC3.4: Update specific software)	Basic LC
Eu.LC.4076	Info	Alternative Scenario: Handling of interrupted PDI connection (case 5) without activation [LC SD 2.6.9] **LC_UC2.6: Handle irregularities** Alternative Scenario: Handling of interrupted PDI connection (case 5) without activation [LC SD 2.6.9] **Precondition:** The Subsystem - Level Crossing is in the state OPERATIONAL. The Subsystem - Level Crossing is configured to deactivate the Level Crossing protection facility as Initial State Of Outputs. The Subsystem - Level Crossing is not in the state ISOLATED LC. **Interaction 2.6.9.A:** **1.** - The PDI connection has been terminated. **Postcondition:** The Subsystem - Level Crossing is in the state INITIALISING. The PDI connection is terminated.  	If the PDI connection is re-established before the expiration of Con_t_PDI_Loss_Deactivation_Delay, the monitoring of the delay stops. The following functionality remains available within the state INITIALISING after the termination of the PDI connection: • Reaction to expiration of Con_t_PDI_Loss_Deactivation_Delay (Alternative Scenario: Handling of interrupted PDI connection (case 1) with activation followed by deactivation without closure time [LC SD 2.6.5], Alternative Scenario: Handling of interrupted PDI connection (case 2) with activation followed by deactivation with closure time [LC SD 2.6.6]) • All functionality related to the local maintainer interface (LC_UC3.4: Update specific software)	Basic LC

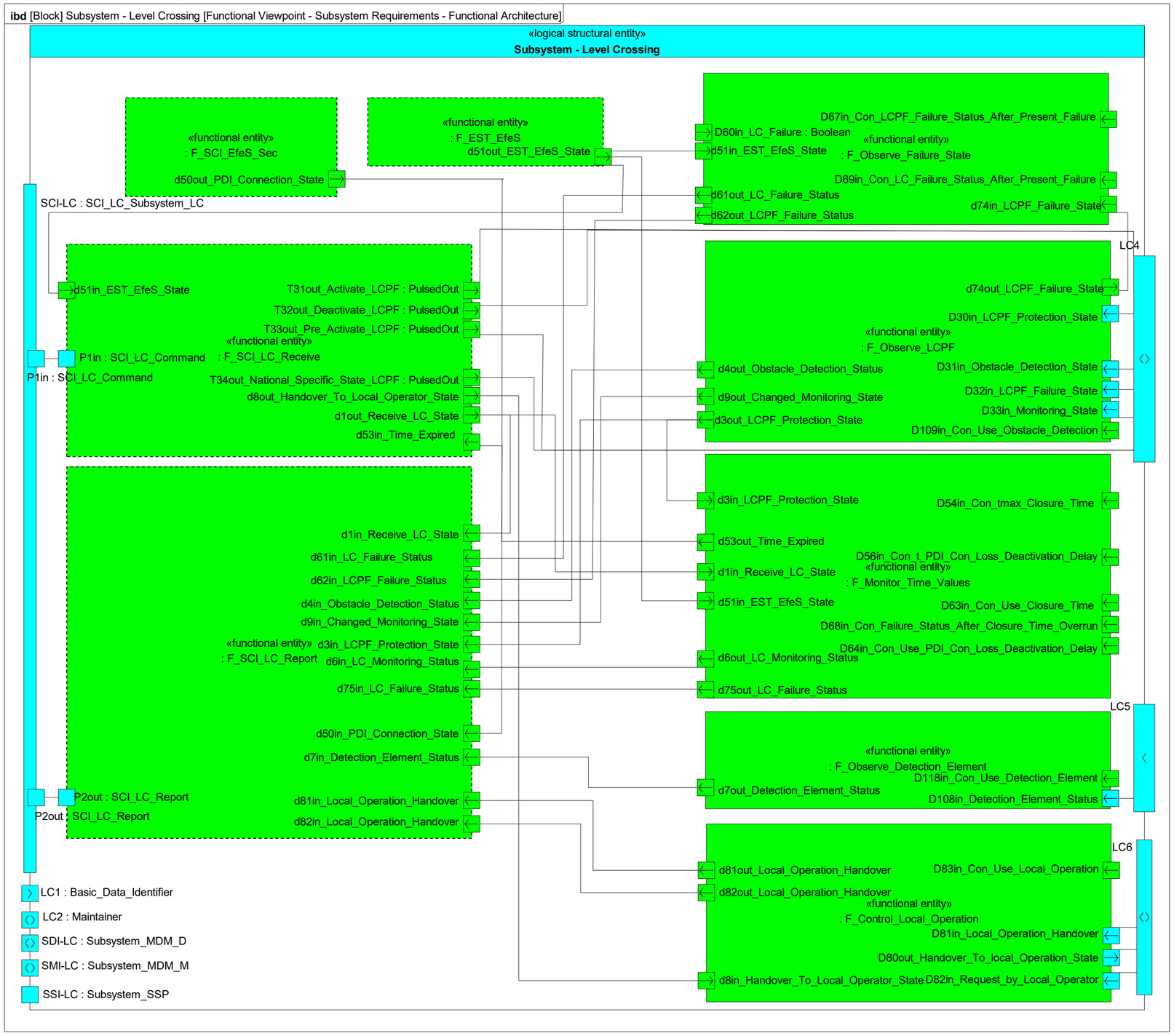
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.1140	Info	Alternative Scenario: Handling of interrupted PDI connection (case 6) Isolated [LC SD 2.6.10] **LC_UC2.6: Handle irregularities**  Alternative Scenario: Handling of interrupted PDI connection (case 6) Isolated [LC SD 2.6.10] **Precondition:** The Subsystem - Level Crossing is configured to use Isolation. The Subsystem - Level Crossing is in the state OPERATIONAL. The Subsystem - Level Crossing is in the state ISOLATED LC. **Interaction 2.6.10.A:** **1.** - The PDI connection has been terminated. **2.** The Subsystem - Level Crossing stays in the state ISOLATED LC. **Postcondition:** The Subsystem - Level Crossing is in the state INITIALISING. The PDI connection is terminated. The Subsystem - Level Crossing is in state ISOLATED LC.		Basic LC
Eu.LC.1045	Info	Alternative Scenario: Reset occurs (case 1) activate without closure time [LC SD 2.6.11] **LC_UC2.6: Handle irregularities**  Alternative Scenario: Reset occurs (case 1) activate without closure time [LC SD 2.6.11] **Precondition:** The Subsystem - Level Crossing is in the state INITIALISING or OPERATIONAL. The Subsystem - Level Crossing is configured to activate the Level Crossing protection facility as Initial State Of Outputs and after an interruption of the PDI connection. **Interaction 2.6.11.A:** **1.** - A reset has occurred. **2.** The Subsystem - Level Crossing activates the Level Crossing protection facility. **Postcondition:** The Subsystem - Level Crossing is in the state BOOTING.		Basic LC
Eu.LC.3966	Info	Alternative Scenario: Reset occurs (case 2) activate with closure time [LC SD 2.6.12] **LC_UC2.6: Handle irregularities**  Alternative Scenario: Reset occurs (case 2) activate with closure time [LC SD 2.6.12] **Precondition:** The Subsystem - Level Crossing is in the state INITIALISING or OPERATIONAL. The Subsystem - Level Crossing is configured to use Closure Time. The Subsystem - Level Crossing is configured to activate the Level Crossing protection facility as Initial State Of Outputs and after an interruption of the PDI connection. **Interaction 2.6.12.A:** **1.** - A reset has occurred. **2.** The Subsystem - Level Crossing activates the Level Crossing protection facility. **3.** The Subsystem - Level Crossing starts to monitor the time period Con_tmax_Closure_Time. **Postcondition:** The Subsystem - Level Crossing is in the state BOOTING, with a currently monitored time value Con_tmax_Closure_Time.		Basic LC

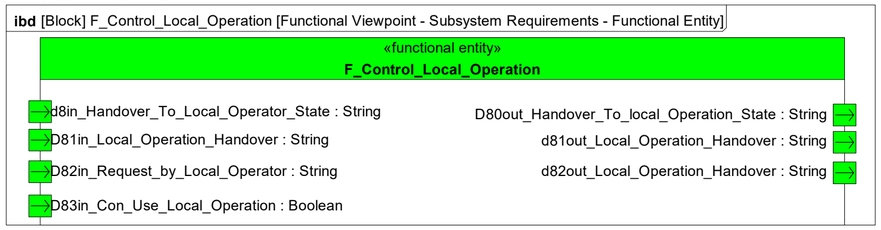
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.4077	Info	Alternative Scenario: Reset occurs (case 3) deactivate [LC SD 2.6.13] **LC_UC2.6: Handle irregularities** Alternative Scenario: Reset occurs (case 3) deactivate [LC SD 2.6.13] **Precondition:** The Subsystem - Level Crossing is in the state INITIALISING or OPERATIONAL. The Subsystem - Level Crossing is configured to deactivate the Level Crossing protection facility as Initial State Of Outputs. **Interaction 2.6.13.A:** **1.** - A reset has occurred. **2.** The Subsystem - Level Crossing deactivates the Level Crossing protection facility. **Postcondition:** The Subsystem - Level Crossing is in the state BOOTING.  		Basic LC
Eu.LC.1073	Info	Alternative Scenario: Supply voltage of the Subsystem has gone outside the required range [LC SD 2.6.14] **LC_UC2.6: Handle irregularities** Alternative Scenario: Supply voltage of the Subsystem has gone outside the required range [LC SD 2.6.14] **Precondition:** --- **Interaction 2.6.14.A:** **1.** - The Subsystem - Level Crossing enters the state NO_OPERATING_VOLTAGE. **2.** The Subsystem - Level Crossing requests the Level Crossing protection facility to change to the most safe national specific state. **Postcondition:** The Subsystem - Level Crossing is in the state NO_OPERATING_VOLTAGE. The Subsystem - Level Crossing is in the state according to the national requirements.  		Basic LC
Eu.LC.1126	Info	Alternative Scenario: Report changed status of protection of an activated LC [LC SD 2.6.15] **LC_UC2.6: Handle irregularities** Alternative Scenario: Report changed status of protection of an activated LC [LC SD 2.6.15] **Precondition:** The Subsystem - Level Crossing is in the state OPERATIONAL. The Subsystem - Level Crossing is in the state ACTIVATED AND PROTECTED. **Interaction 2.6.15.A:** **1.** - The Subsystem - Level Crossing receives from the Level Crossing protection facility the new status unprotected. **2.** The Subsystem - Level Crossing reports to the Subsystem - Electronic Interlocking that the functional status has been changed to ACTIVATED AND UNPROTECTED. **Postcondition:** The Subsystem - Level Crossing is in the state ACTIVATED AND UNPROTECTED.  		Basic LC
Eu.LC.1164	Info	LC_UC2.7: Handle isolate LC	The Subsystem-UseCase LC_UC2.7: Handle isolate LC defines the behaviour of the Subsystem - Level Crossing in case of a command Isolate LC.	Basic LC

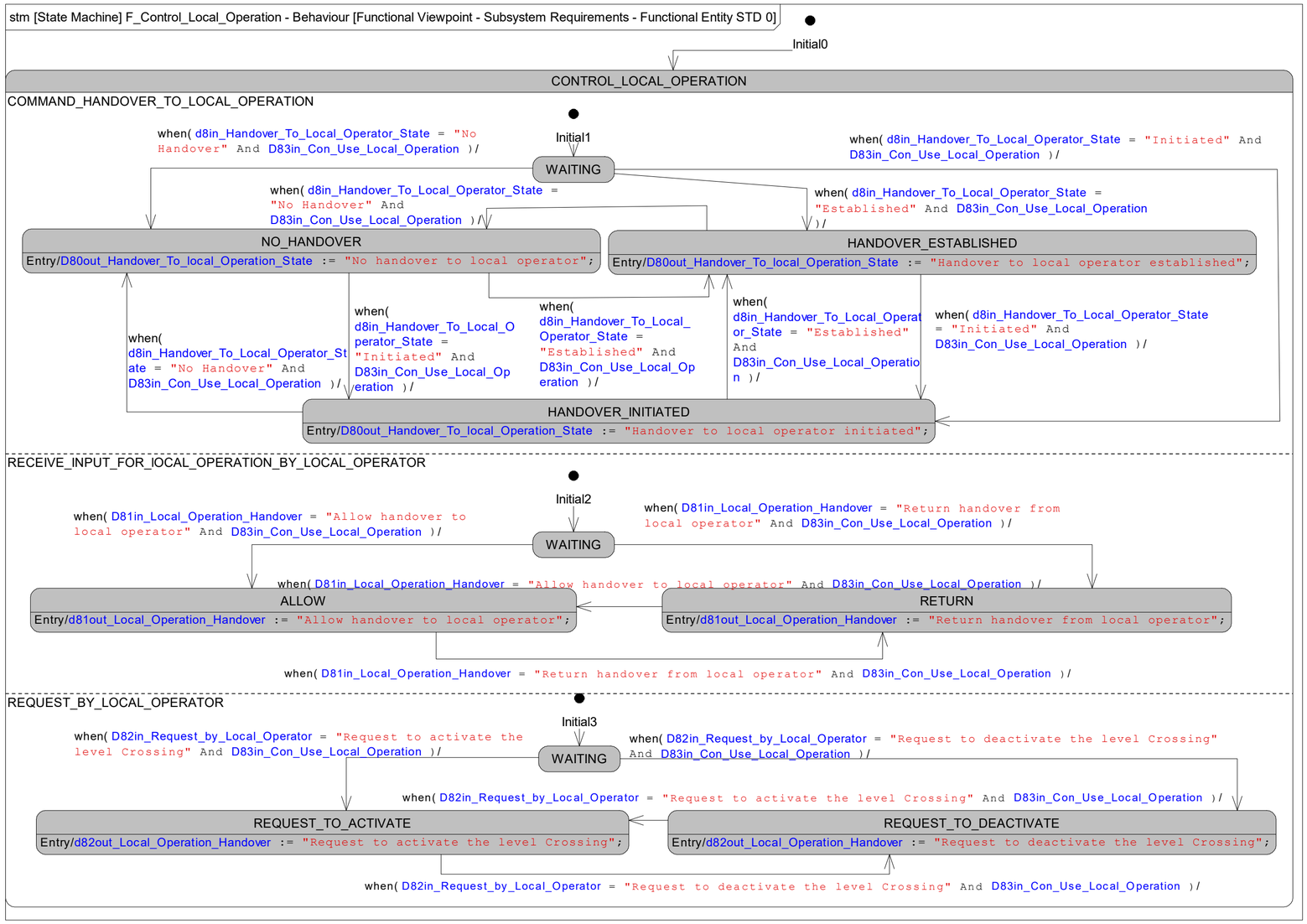
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.1165	Info	Alternative Scenario: Isolate LC [LC 2.7.1] **LC_UC2.7: Handle isolate LC** Alternative Scenario: Isolate LC [LC 2.7.1] **Precondition:** The Subsystem - Level Crossing is configured to use isolation. The Subsystem - Level Crossing is in the state OPERATIONAL. The Subsystem - Level Crossing is in the state DEACTIVATED AND IDLE or DEACTIVATING AND UNPROTECTED. **Interaction 2.7.1.A:** **1.** - The Subsystem - Level Crossing receives from the Subsystem - Electronic Interlocking the command to be isolated. **2.** The Subsystem - Level Crossing reports to the Subsystem - Electronic Interlocking that the functional status has been changed. **Postcondition:** The Subsystem - Level Crossing is in the state ISOLATED LC.  		Basic LC
Eu.LC.1174	Info	Alternative Scenario: Not isolate LC [LC 2.7.2] **LC_UC2.7: Handle isolate LC** Alternative Scenario: Not isolate LC [LC 2.7.2] **Precondition:** The Subsystem - Level Crossing is configured to use isolation. The Subsystem - Level Crossing is in the state OPERATIONAL. The Subsystem - Level Crossing is in the state ISOLATED LC. **Interaction 2.7.2.A:** **1.** - The Subsystem - Level Crossing receives from the Subsystem - Electronic Interlocking the command to be not isolated. **2.** The Subsystem - Level Crossing reports to the Subsystem - Electronic Interlocking that the functional status has been changed. **Postcondition:** The Subsystem - Level Crossing is in the state DEACTIVATING AND UNPROTECTED.  		Basic LC

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.1249	Info	[Package] Subsystem - Level Crossing - Functional Context [Functional Viewpoint - Subsystem Definition - Maintenance]  		Basic LC
Eu.LC.1250	Info	LC_UC3.2: Collect and provide event-driven diagnostic data	Information: The Subsystem-UseCase "LC_UC3.2: Collect and provide event-driven diagnostic data" defines the event driven collection and provision of diagnostic data in case of irregularities.	Basic LC
Eu.LC.1251	Info	LC_UC3.3: Collect and provide preventive diagnostic data	Information: The Subsystem-UseCase "LC_UC3.3: Collect and provide preventive diagnostic data" defines the continuous collection and provision of diagnostic data for preventive maintenance.	Basic LC
Eu.LC.1252	Info	LC_UC3.4: Update specific software	Information: The Subsystem-UseCase "LC_UC3.4: Update specific software" defines the process of updating the specific software between Subsystem - Maintenance and Data Management and the Subsystem.	Basic LC
Eu.LC.1253	Info	LC_UC3.1: Display status of Subsystem - Level Crossing locally	Information: The Subsystem-UseCase "LC_UC3.1: Display status of Subsystem - Level Crossing locally" defines the local display of the EULYNX field element Subsystem. See ID Eu.LC.138.	Basic LC
Eu.LC.3222	Head	**3.3.3 Subsystem - Level Crossing - Functional Partitioning**		

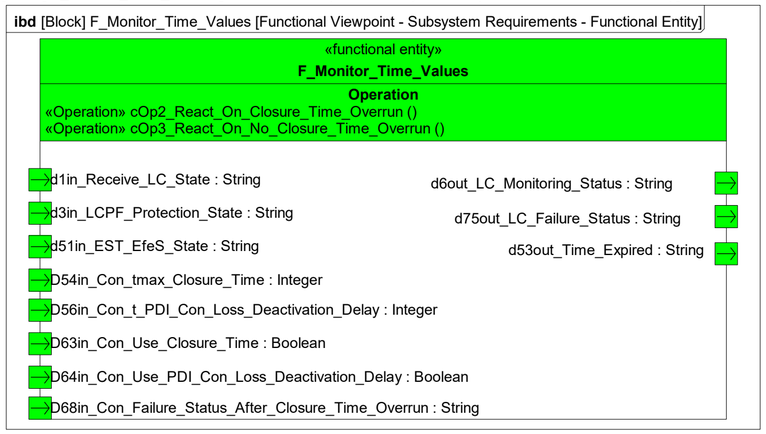
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.1985	Def	[Package] Subsystem Level Crossing - Functional Partitioning [Functional Viewpoint - Subsystem Requirements]  		Basic LC
Eu.LC.2374	Head	**3.3.4 Subsystem - Level Crossing - Functional Architecture**		
Eu.LC.72	Info	Subsystem - Level Crossing		Basic LC

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.73	Def	[Block] Subsystem - Level Crossing [Functional Viewpoint - Subsystem Requirements - Functional Architecture]  		Basic LC
Eu.LC.3086	Def	SCI-LC	The functional Process Data interface to the Subsystem - Electronic Interlocking (SCI: Standard Communication Interface). The InformationFlow through the interface is further defined in SCI-LS (Subsystem - Electronic Interlocking).	Basic LC
Eu.LC.77	Def	LC1	The functional System Data interface to the Basic Data identifier. The InformationFlow through the interface is defined by "Basic_Data_Identifier".	Basic LC

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.3211	Def	LC2	The functional Maintenance/Operation/Display interface to the Maintainer. The InformationFlow through the interface is defined by "Maintainer".	Basic LC
Eu.LC.76	Def	SDI-LC	The functional Diagnostic interface to the Subsystem - Maintenance and Data Management for the InformationFlow through the interface, which is defined by "Subsystem_MDM_D".	Basic LC
Eu.LC.75	Def	SMI-LC	The functional Maintenance Interface to the Subsystem - Maintenance and Data Management for the InformationFlow through the interface, which is defined by "Subsystem_MDM_M".	Basic LC
Eu.LC.79	Def	LC4	The functional Control interface to the Level Crossing protection facility. The InformationFlow through the interface is defined by "Level_Crossing_protection_facility".	Basic LC
Eu.LC.81	Def	LC5	The functional Control interface to the Detection element. The InformationFlow through the interface is defined by "Detection_element".	Basic LC
Eu.LC.82	Def	LC6	The functional Local Control and Display interface to the Local operator. The InformationFlow through the interface is defined by Local_operator".	Option LOH
Eu.LC.3271	Def	SSI-LC	The Security Service Interface to the Subsystem - Security Services Platform. The InformationFlow through the interface is further defined in SSI-LC (Subsystem - Security Services Platform).	Basic LC
Eu.LC.2377	Head	**3.3.5 Subsystem - Level Crossing - Functional Entities**		
Eu.LC.2378	Info	F_Control_Local_Operation		Option LOH
Eu.LC.2379	Req	[Block] F_Control_Local_Operation [Functional Viewpoint - Subsystem Requirements - Functional Entity]  		Option LOH
Eu.LC.2380	Def	D80out_Handover_To_local_Operation_State	The port D80out_Handover_To_local_Operation_State refines the Informationflow Output_Established_Handover_To_Local_Operator, Output_No_Handover_To_Local_Operator and Output_Initiated_Handover_To_Local_Operator.	Option LOH
Eu.LC.2381	Def	D81in_Local_Operation_Handover	The port D81in_Local_Operation_Handover refines the Informationflow Input_Allow_Handover_To_Local_Operator and Input_Return_Handover_To_Local_Operator.	Option LOH
Eu.LC.2382	Def	d81out_Local_Operation_Handover		Option LOH
Eu.LC.2383	Def	D82in_Request_by_Local_Operator	The port D82in_Request_by_Local_Operator refines the Informationflow Activate and Deactivate.	Option LOH
Eu.LC.2384	Def	d82out_Local_Operation_Handover		Option LOH
Eu.LC.2385	Def	d8in_Handover_To_Local_Operator_State		Option LOH
Eu.LC.3799	Def	D83in_Con_Use_Local_Operation	The port D83in_Con_Use_Local_Operation provides configuration values for the Local_operator. true: Local_operator is used false: Local_operator is not used	Option LOH
Eu.LC.2386	Info	F_Control_Local_Operation - Behaviour		Option LOH

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.2387	Req	Functional Viewpoint - Subsystem Requirements - Functional Entity STD 0  	This state machine diagram describes the requirements for the following functionalities: - receives the status of the handover to the local operator - receives the information to allow or return handover to local operator - receives the requests by the local operator to activate or deactivate the level crossing	Option LOH
Eu.LC.2388	Def	CONTROL_LOCAL_OPERATION		Option LOH
Eu.LC.2389	Def	COMMAND_HANDOVER_TO_LOCAL_OPERATION		Option LOH
Eu.LC.2390	Def	HANDOVER_ESTABLISHED		Option LOH
Eu.LC.2391	Def	when(d8in_Handover_To_Local_Operator_State = "No Handover" And D83in_Con_Use_Local_Operation)/{HANDOVER_ESTABLISHED - NO_HANDOVER}		Option LOH
Eu.LC.2392	Def	when(d8in_Handover_To_Local_Operator_State = "Initiated" And D83in_Con_Use_Local_Operation)/{HANDOVER_ESTABLISHED - HANDOVER_INITIATED}		Option LOH
Eu.LC.3000	Def	entry/D80out_Handover_To_local_Operation_State := "Handover to local operator established";{State-internal in HANDOVER_ESTABLISHED}		Option LOH

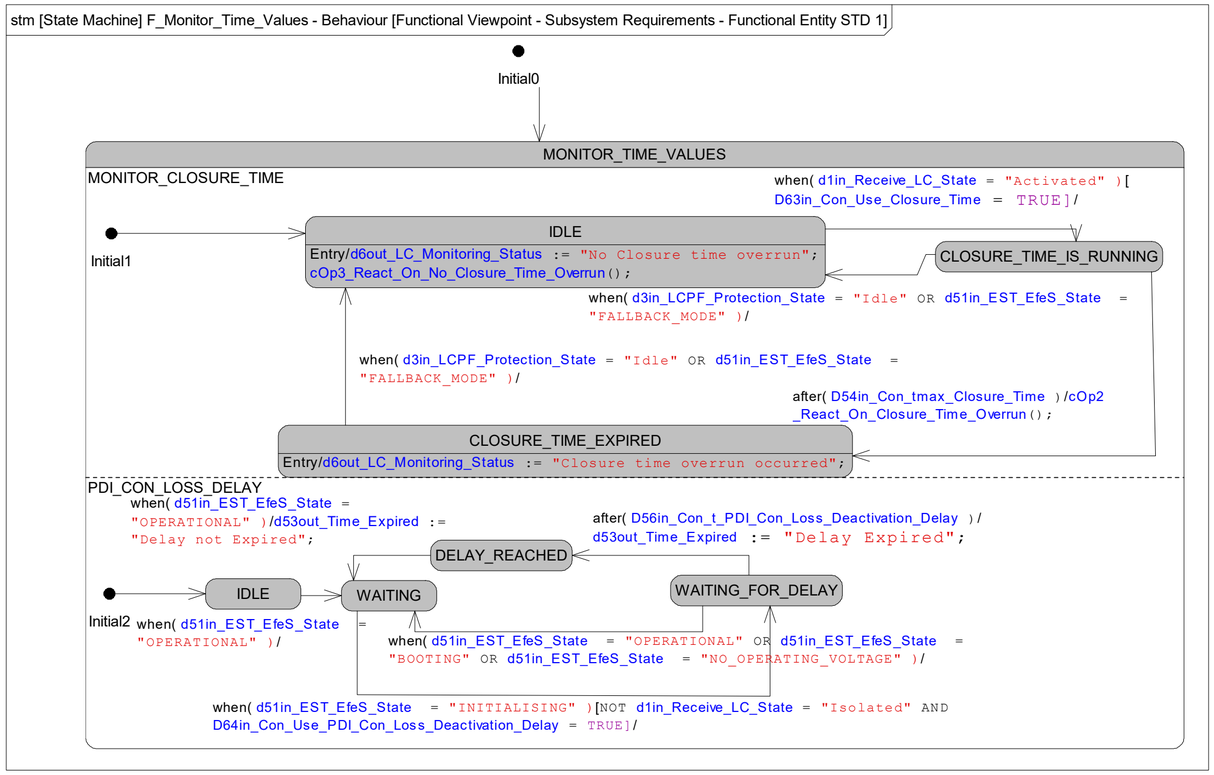
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.2393	Def	HANDOVER_INITIATED		Option LOH
Eu.LC.2394	Def	when(d8in_Handover_To_Local_Operator_State = "Established" And D83in_Con_Use_Local_Operation)/{HANDOVER_INITIATED - HANDOVER_ESTABLISHED}		Option LOH
Eu.LC.2395	Def	when(d8in_Handover_To_Local_Operator_State = "No Handover" And D83in_Con_Use_Local_Operation)/{HANDOVER_INITIATED - NO_HANDOVER}		Option LOH
Eu.LC.3001	Def	entry/D80out_Handover_To_local_Operation_State := "Handover to local operator initiated";{State-internal in HANDOVER_INITIATED}		Option LOH
Eu.LC.2396	Def	Initial1		Option LOH
Eu.LC.2397	Def	/{Initial1 - WAITING}		Option LOH
Eu.LC.2398	Def	NO_HANDOVER		Option LOH
Eu.LC.2399	Def	when(d8in_Handover_To_Local_Operator_State = "Initiated" And D83in_Con_Use_Local_Operation)/{NO_HANDOVER - HANDOVER_INITIATED}		Option LOH
Eu.LC.2400	Def	when(d8in_Handover_To_Local_Operator_State = "Established" And D83in_Con_Use_Local_Operation)/{NO_HANDOVER - HANDOVER_ESTABLISHED}		Option LOH
Eu.LC.3002	Def	entry/D80out_Handover_To_local_Operation_State := "No handover to local operator";{State-internal in NO_HANDOVER}		Option LOH
Eu.LC.2401	Def	WAITING		Option LOH
Eu.LC.2402	Def	when(d8in_Handover_To_Local_Operator_State = "Established" And D83in_Con_Use_Local_Operation)/{WAITING - HANDOVER_ESTABLISHED}		Option LOH
Eu.LC.2403	Def	when(d8in_Handover_To_Local_Operator_State = "Initiated" And D83in_Con_Use_Local_Operation)/{WAITING - HANDOVER_INITIATED}		Option LOH
Eu.LC.2404	Def	when(d8in_Handover_To_Local_Operator_State = "No Handover" And D83in_Con_Use_Local_Operation)/{WAITING - NO_HANDOVER}		Option LOH
Eu.LC.2405	Def	RECEIVE_INPUT_FOR_lOCAL_OPERATION_BY_LOCAL_OPERATOR		Option LOH
Eu.LC.2406	Def	ALLOW		Option LOH
Eu.LC.2407	Def	when(D81in_Local_Operation_Handover = "Return handover from local operator" And D83in_Con_Use_Local_Operation)/{ALLOW - RETURN}		Option LOH
Eu.LC.3003	Def	entry/d81out_Local_Operation_Handover := "Allow handover to local operator";{State-internal in ALLOW}		Option LOH
Eu.LC.2408	Def	Initial2		Option LOH
Eu.LC.2409	Def	/{Initial2 - WAITING}		Option LOH
Eu.LC.2410	Def	RETURN		Option LOH
Eu.LC.2411	Def	when(D81in_Local_Operation_Handover = "Allow handover to local operator" And D83in_Con_Use_Local_Operation)/{RETURN - ALLOW}		Option LOH
Eu.LC.3004	Def	entry/d81out_Local_Operation_Handover := "Return handover from local operator";{State-internal in RETURN}		Option LOH
Eu.LC.2412	Def	WAITING		Option LOH
Eu.LC.2413	Def	when(D81in_Local_Operation_Handover = "Allow handover to local operator" And D83in_Con_Use_Local_Operation)/{WAITING - ALLOW}		Option LOH
Eu.LC.2414	Def	when(D81in_Local_Operation_Handover = "Return handover from local operator" And D83in_Con_Use_Local_Operation)/{WAITING - RETURN}		Option LOH
Eu.LC.2415	Def	REQUEST_BY_LOCAL_OPERATOR		Option LOH
Eu.LC.2416	Def	Initial3		Option LOH
Eu.LC.2417	Def	/{Initial3 - WAITING}		Option LOH
Eu.LC.2418	Def	REQUEST_TO_ACTIVATE		Option LOH
Eu.LC.2419	Def	when(D82in_Request_by_Local_Operator = "Request to deactivate the level Crossing" And D83in_Con_Use_Local_Operation)/{REQUEST_TO_ACTIVATE - REQUEST_TO_DEACTIVATE}		Option LOH

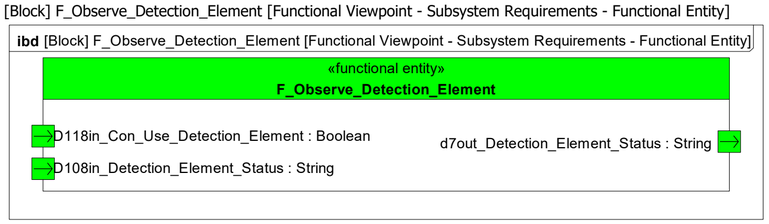
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.3005	Def	entry/d82out_Local_Operation_Handover := "Request to activate the level Crossing";{State-internal in REQUEST_TO_ACTIVATE}		Option LOH
Eu.LC.2420	Def	REQUEST_TO_DEACTIVATE		Option LOH
Eu.LC.2421	Def	when(D82in_Request_by_Local_Operator = "Request to activate the level Crossing" And D83in_Con_Use_Local_Operation)/{REQUEST_TO_DEACTIVATE - REQUEST_TO_ACTIVATE}		Option LOH
Eu.LC.3006	Def	entry/d82out_Local_Operation_Handover := "Request to deactivate the level Crossing";{State-internal in REQUEST_TO_DEACTIVATE}		Option LOH
Eu.LC.2422	Def	WAITING		Option LOH
Eu.LC.2423	Def	when(D82in_Request_by_Local_Operator = "Request to activate the level Crossing" And D83in_Con_Use_Local_Operation)/{WAITING - REQUEST_TO_ACTIVATE}		Option LOH
Eu.LC.2424	Def	when(D82in_Request_by_Local_Operator = "Request to deactivate the level Crossing" And D83in_Con_Use_Local_Operation)/{WAITING - REQUEST_TO_DEACTIVATE}		Option LOH
Eu.LC.2425	Def	Initial0		Option LOH
Eu.LC.2426	Def	/{Initial0 - CONTROL_LOCAL_OPERATION}		Option LOH
Eu.LC.2427	Info	F_Monitor_Time_Values		Basic LC
Eu.LC.2428	Req	[Block] F_Monitor_Time_Values [Functional Viewpoint - Subsystem Requirements - Functional Entity]  		Basic LC

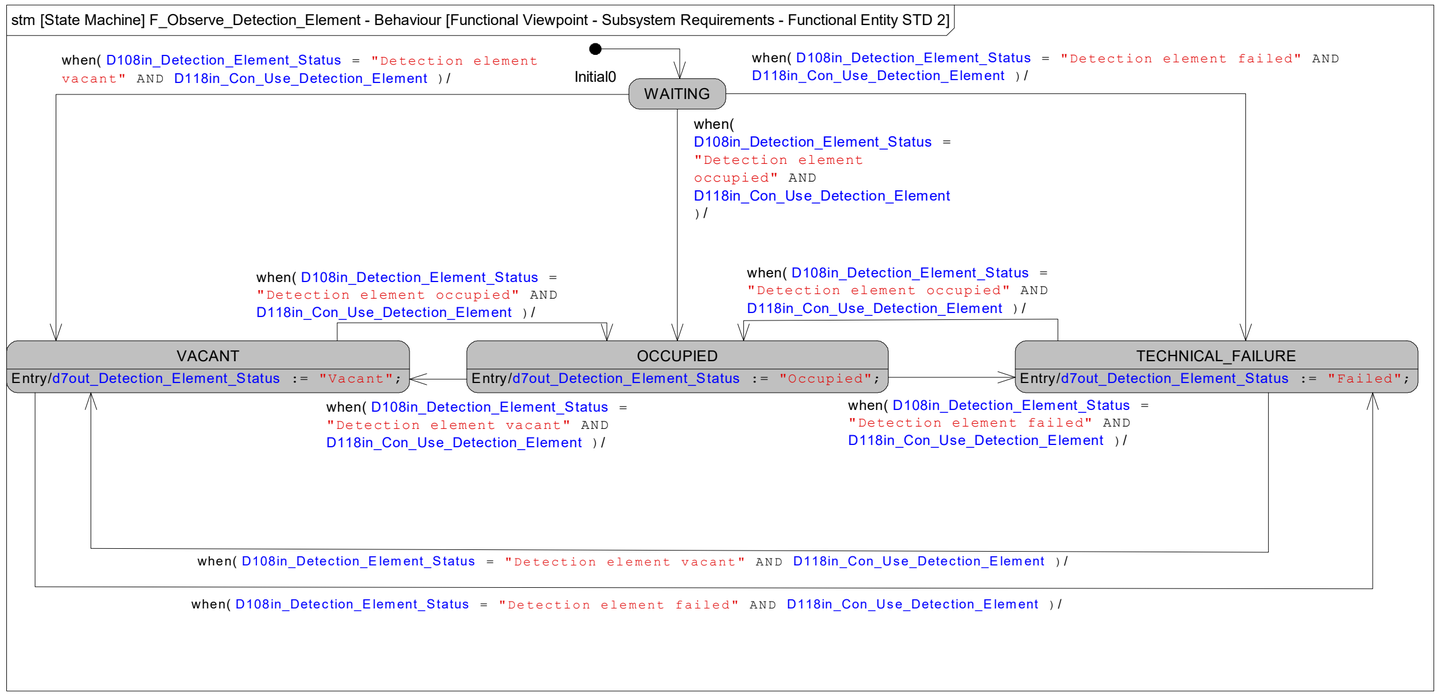
Eu.LC.2429	Def	/* cOp2_React_On_Closure_Time_Overrun */
		
		if D68in_Con_Failure_Status_After_Closure_Time_Overrun = "non critical failure report" then
		d75out_LC_Failure_Status := "A non critical failure is present";
		elseif D68in_Con_Failure_Status_After_Closure_Time_Overrun = "critical failure report" then
		d75out_LC_Failure_Status := "A critical failure is present";
		end if
			cOp2_React_On_Closure_Time_Overrun	Basic LC

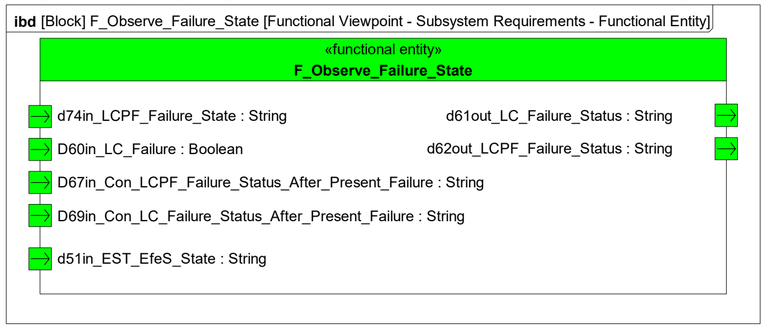
Eu.LC.2430	Def	/* cOp3_React_On_No_Closure_Time_Overrun */
		
		if D68in_Con_Failure_Status_After_Closure_Time_Overrun = "non critical failure report" then
		d75out_LC_Failure_Status := "No non critical failure present";
		elseif D68in_Con_Failure_Status_After_Closure_Time_Overrun = "critical failure report" then
		d75out_LC_Failure_Status := "No critical failure present";
		end if
			cOp3_React_On_No_Closure_Time_Overrun	Basic LC

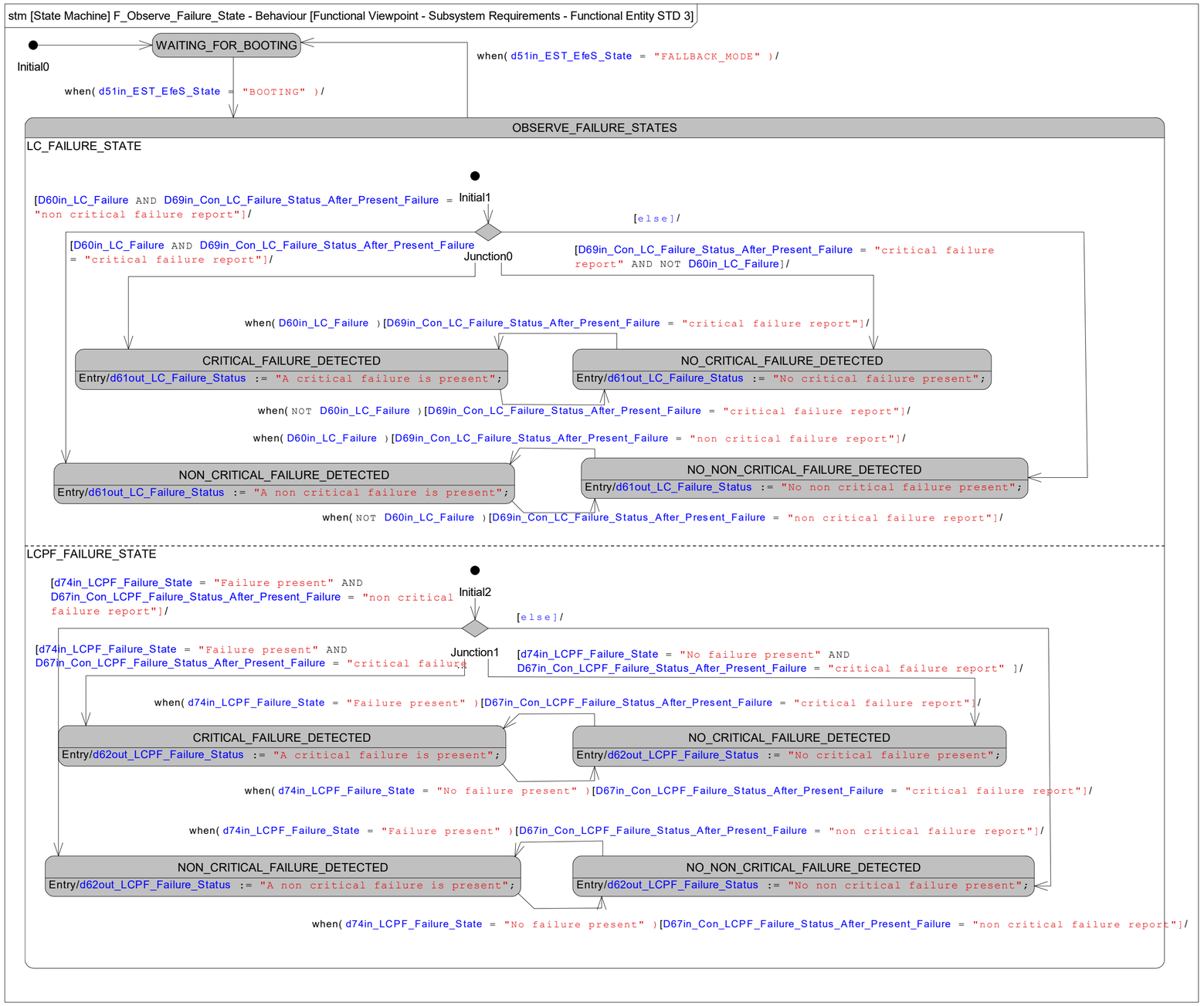
Eu.LC.2431	Def	d1in_Receive_LC_State		Basic LC
Eu.LC.2432	Def	d3in_LCPF_Protection_State		Basic LC
Eu.LC.2433	Def	d51in_EST_EfeS_State		Basic LC
Eu.LC.2434	Def	D54in_Con_tmax_Closure_Time	The port D54in_Con_tmax_Closure_Time refines the time value for Con_tmax_Closure_Time. The following values are permitted: - 1 up to any number	Basic LC
Eu.LC.2435	Def	d53out_Time_Expired		Basic LC

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.2436	Def	D56in_Con_t_PDI_Con_Loss_Deactivation_Delay	The port D56in_Con_t_PDI_Con_Loss_Deactivation_Delay refines the time value for Con_t_PDI_Loss_Deactivation_Delay. The following values are permitted: - 1 up to any number	Basic LC
Eu.LC.2437	Def	D63in_Con_Use_Closure_Time	The port D63in_Con_Use_Closure_Time provides configuration values for the Con_tmax_Closure_Time. true: Con_tmax_Closure_Time is used false: Con_tmax_Closure_Time is not used	Basic LC
Eu.LC.2438	Def	D64in_Con_Use_PDI_Con_Loss_Deactivation_Delay	The port D64in_Con_Use_PDI_Con_Loss_Deactivation_Delay provides configuration values for the Con_t_PDI_Loss_Deactivation_Delay. true: Con_t_PDI_Loss_Deactivation_Delay is used false: Con_t_PDI_Loss_Deactivation_Delay is not used	Basic LC
Eu.LC.2439	Def	D68in_Con_Failure_Status_After_Closure_Time_Overrun	The port D68in_Con_Failure_Status_After_Closure_Time_Overrun provides the configuration value what failure status the Subsystem - Level Crossing is configured to report after a closure time overrun occurred.	Basic LC
Eu.LC.2440	Def	d6out_LC_Monitoring_Status		Basic LC
Eu.LC.2441	Def	d75out_LC_Failure_Status		Basic LC
Eu.LC.2442	Info	F_Monitor_Time_Values - Behaviour		Basic LC
Eu.LC.2443	Req	Functional Viewpoint - Subsystem Requirements - Functional Entity STD 1  	This state machine diagram describes the requirements for the following functionalities: - monitors the Closure time overrun and forwards an overrun - monitors the PDI Con Loss Delay and forwards an overrun	Basic LC
Eu.LC.2444	Def	FALLBACKMODE		Basic LC
Eu.LC.2445	Def	Initial0		Basic LC

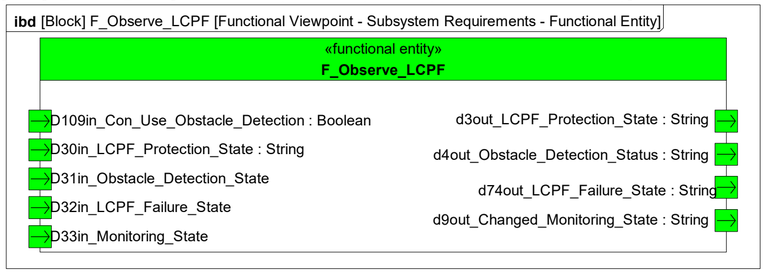
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.2446	Def	/{Initial0 - MONITOR_TIME_VALUES}		Basic LC
Eu.LC.2447	Def	MONITOR_TIME_VALUES		Basic LC
Eu.LC.2448	Def	MONITOR_CLOSURE_TIME		Basic LC
Eu.LC.2449	Def	CLOSURE_TIME_EXPIRED		Basic LC
Eu.LC.2450	Def	when(d3in_LCPF_Protection_State = "Idle" OR d51in_EST_EfeS_State = "FALLBACK_MODE")/{CLOSURE_TIME_EXPIRED - IDLE}		Basic LC
Eu.LC.3214	Def	entry/d6out_LC_Monitoring_Status := "Closure time overrun occurred";{State-internal in CLOSURE_TIME_EXPIRED}		Basic LC
Eu.LC.2451	Def	CLOSURE_TIME_IS_RUNNING		Basic LC
Eu.LC.2452	Def	after(D54in_Con_tmax_Closure_Time)/cOp2_React_On_Closure_Time_Overrun();{CLOSURE_TIME_IS_RUNNING - CLOSURE_TIME_EXPIRED}		Basic LC
Eu.LC.2453	Def	when(d3in_LCPF_Protection_State = "Idle" OR d51in_EST_EfeS_State = "FALLBACK_MODE")/{CLOSURE_TIME_IS_RUNNING - IDLE}		Basic LC
Eu.LC.2454	Def	IDLE		Basic LC
Eu.LC.2455	Def	when(d1in_Receive_LC_State = "Activated")[D63in_Con_Use_Closure_Time = TRUE]/{IDLE - CLOSURE_TIME_IS_RUNNING}		Basic LC
Eu.LC.3215	Def	entry/d6out_LC_Monitoring_Status := "No Closure time overrun"; cOp3_React_On_No_Closure_Time_Overrun();{State-internal in IDLE}		Basic LC
Eu.LC.2456	Def	Initial1		Basic LC
Eu.LC.2457	Def	/{Initial1 - IDLE}		Basic LC
Eu.LC.2458	Def	PDI_CON_LOSS_DELAY		Basic LC
Eu.LC.2459	Def	IDLE		Basic LC
Eu.LC.3216	Def	when(d51in_EST_EfeS_State = "OPERATIONAL")/{IDLE - WAITING}		
Eu.LC.2461	Def	Initial2		Basic LC
Eu.LC.2462	Def	/{Initial2 - IDLE}		Basic LC
Eu.LC.2463	Def	DELAY_REACHED		Basic LC
Eu.LC.2464	Def	when(d51in_EST_EfeS_State = "OPERATIONAL")/d53out_Time_Expired := "Delay not Expired";{DELAY_REACHED - WAITING}		Basic LC
Eu.LC.2465	Def	WAITING_FOR_DELAY		Basic LC
Eu.LC.2466	Def	after(D56in_Con_t_PDI_Con_Loss_Deactivation_Delay)/ d53out_Time_Expired := "Delay Expired";{WAITING_FOR_DELAY - DELAY_REACHED}		Basic LC
Eu.LC.3217	Def	when(d51in_EST_EfeS_State = "OPERATIONAL" OR d51in_EST_EfeS_State = "BOOTING" OR d51in_EST_EfeS_State = "NO_OPERATING_VOLTAGE")/{WAITING_FOR_DELAY - WAITING}		Basic LC
Eu.LC.3218	Def	WAITING		Basic LC
Eu.LC.2460	Def	when(d51in_EST_EfeS_State = "INITIALISING")[NOT d1in_Receive_LC_State = "Isolated" AND D64in_Con_Use_PDI_Con_Loss_Deactivation_Delay = TRUE]/{WAITING - WAITING_FOR_DELAY}		Basic LC
Eu.LC.2467	Def	TECHNICAL_DISTURBANCE		Basic LC
Eu.LC.2468	Def	WAITING_FOR_FINISH_BOOTING		Basic LC
Eu.LC.2469	Info	F_Observe_Detection_Element		Basic LC
Eu.LC.2470	Req	[Block] F_Observe_Detection_Element [Functional Viewpoint - Subsystem Requirements - Functional Entity]  		Basic LC
Eu.LC.2471	Def	D118in_Con_Use_Detection_Element	The port D118in_Con_Use_Detection_Element provides configuration values for the detection element. true: Detection element is used false: Detection element is not used	Basic LC
Eu.LC.2472	Def	D108in_Detection_Element_Status	The port D108in_Detection_Element_Status provides the status of the detection element and refines the Informationflow Vacated_Detection_Element, Occupied_Detection_Element and Failed_Detection_Element.	Basic LC
Eu.LC.2473	Def	d7out_Detection_Element_Status		Basic LC
Eu.LC.2474	Info	F_Observe_Detection_Element - Behaviour		Basic LC

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.2475	Req	Functional Viewpoint - Subsystem Requirements - Functional Entity STD 2  	This state machine diagram describes the requirements for the following functionalities: - observe the status of the detection elements - forward the observed detection element status	Basic LC
Eu.LC.2476	Def	Initial0		Basic LC
Eu.LC.2477	Def	/{Initial0 - WAITING}		Basic LC
Eu.LC.2478	Def	OCCUPIED		Basic LC
Eu.LC.2479	Def	when(D108in_Detection_Element_Status = "Detection element vacant" AND D118in_Con_Use_Detection_Element)/{OCCUPIED - VACANT}		Basic LC
Eu.LC.2480	Def	when(D108in_Detection_Element_Status = "Detection element failed" AND D118in_Con_Use_Detection_Element)/{OCCUPIED - TECHNICAL_FAILURE}		Basic LC
Eu.LC.3009	Def	entry/d7out_Detection_Element_Status := "Occupied";{State-internal in OCCUPIED}		Basic LC
Eu.LC.2481	Def	TECHNICAL_FAILURE		Basic LC
Eu.LC.2482	Def	when(D108in_Detection_Element_Status = "Detection element occupied" AND D118in_Con_Use_Detection_Element)/{TECHNICAL_FAILURE - OCCUPIED}		Basic LC
Eu.LC.2483	Def	when(D108in_Detection_Element_Status = "Detection element vacant" AND D118in_Con_Use_Detection_Element)/{TECHNICAL_FAILURE - VACANT}		Basic LC
Eu.LC.3010	Def	entry/d7out_Detection_Element_Status := "Failed";{State-internal in TECHNICAL_FAILURE}		Basic LC
Eu.LC.2484	Def	VACANT		Basic LC
Eu.LC.2485	Def	when(D108in_Detection_Element_Status = "Detection element occupied" AND D118in_Con_Use_Detection_Element)/{VACANT - OCCUPIED}		Basic LC
Eu.LC.2486	Def	when(D108in_Detection_Element_Status = "Detection element failed" AND D118in_Con_Use_Detection_Element)/{VACANT - TECHNICAL_FAILURE}		Basic LC
Eu.LC.3011	Def	entry/d7out_Detection_Element_Status := "Vacant";{State-internal in VACANT}		Basic LC
Eu.LC.2487	Def	WAITING		Basic LC
Eu.LC.2488	Def	when(D108in_Detection_Element_Status = "Detection element occupied" AND D118in_Con_Use_Detection_Element)/{WAITING - OCCUPIED}		Basic LC
Eu.LC.2489	Def	when(D108in_Detection_Element_Status = "Detection element vacant" AND D118in_Con_Use_Detection_Element)/{WAITING - VACANT}		Basic LC
Eu.LC.2490	Def	when(D108in_Detection_Element_Status = "Detection element failed" AND D118in_Con_Use_Detection_Element)/{WAITING - TECHNICAL_FAILURE}		Basic LC
Eu.LC.2491	Info	F_Observe_Failure_State		Basic LC

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.2492	Req	[Block] F_Observe_Failure_State [Functional Viewpoint - Subsystem Requirements - Functional Entity]  		Basic LC
Eu.LC.2494	Def	d61out_LC_Failure_Status		Basic LC
Eu.LC.2495	Def	d62out_LCPF_Failure_Status		Basic LC
Eu.LC.2496	Def	D67in_Con_LCPF_Failure_Status_After_Present_Failure	The port D67in_Con_LCPF_Failure_Status_After_Present_Failure provides the configuration value what failure status the Subsystem - Level Crossing is configured to report after a failure is present.	Basic LC
Eu.LC.2497	Def	D69in_Con_LC_Failure_Status_After_Present_Failure	The port D69in_Con_LC_Failure_Status_After_Present_Failure provides the configuration value what failure status the Subsystem - Level Crossing is configured to report after a failure is present.	Basic LC
Eu.LC.2498	Def	d74in_LCPF_Failure_State		Basic LC
Eu.LC.3272	Def	D60in_LC_Failure	The port D60in_LC_Failure represents a failure in the LCPF. true: Failure is present false: Failure is not present	Basic LC
Eu.LC.3830	Def	d51in_EST_EfeS_State		Basic LC
Eu.LC.2499	Info	F_Observe_Failure_State - Behaviour		Basic LC

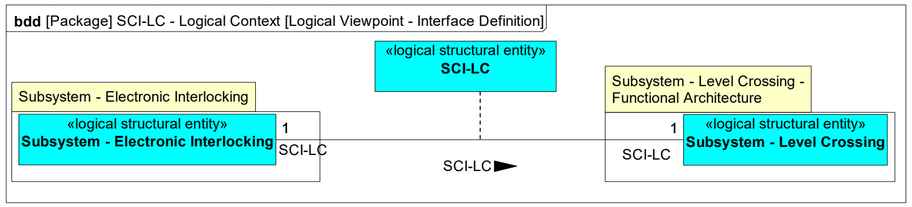
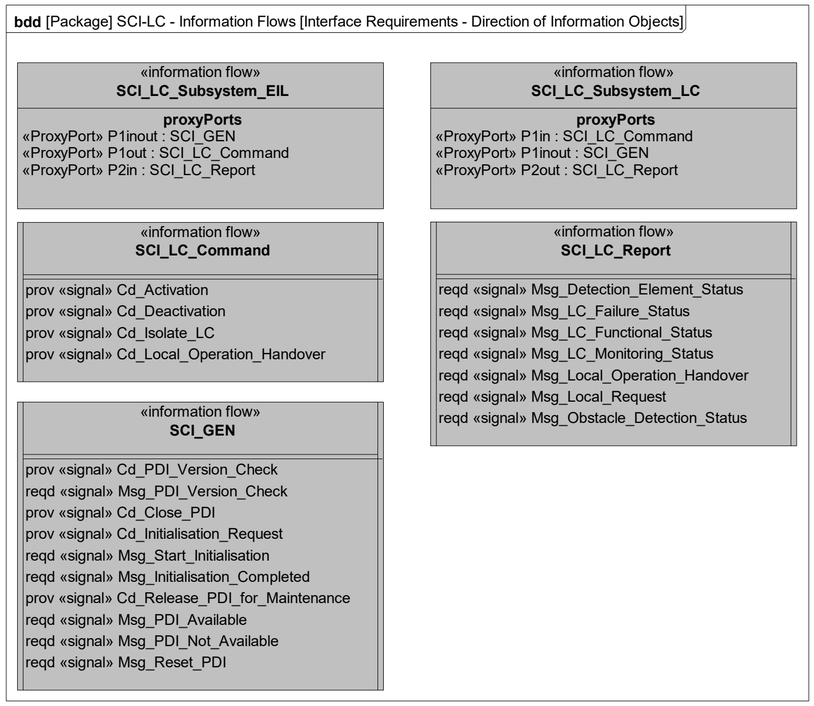
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.2500	Req	Functional Viewpoint - Subsystem Requirements - Functional Entity STD 3  	This state machine diagram describes the requirements for the following functionalities: - observes the critical LC Failure state - observes the non-critical LC Failure state - observes the critical LCPF Failure state - observes the non-critical LCPF Failure state - forwards failure states	Basic LC
Eu.LC.2501	Def	Initial0		Basic LC
Eu.LC.2502	Def	/{Initial0 - WAITING_FOR_BOOTING}		Basic LC
Eu.LC.2503	Def	OBSERVE_FAILURE_STATES		Basic LC
Eu.LC.2504	Def	LC_FAILURE_STATE		Basic LC
Eu.LC.2505	Def	CRITICAL_FAILURE_DETECTED		Basic LC

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.2506	Def	when(NOT D60in_LC_Failure)[D69in_Con_LC_Failure_Status_After_Present_Failure = "critical failure report"]/{CRITICAL_FAILURE_DETECTED - NO_CRITICAL_FAILURE_DETECTED}		Basic LC
Eu.LC.3012	Def	entry/d61out_LC_Failure_Status := "A critical failure is present";{State-internal in CRITICAL_FAILURE_DETECTED}		Basic LC
Eu.LC.2507	Def	Initial1		Basic LC
Eu.LC.2508	Def	/{Initial1 - Junction0}		Basic LC
Eu.LC.2509	Def	NO_CRITICAL_FAILURE_DETECTED		Basic LC
Eu.LC.2510	Def	when(D60in_LC_Failure)[D69in_Con_LC_Failure_Status_After_Present_Failure = "critical failure report"]/{NO_CRITICAL_FAILURE_DETECTED - CRITICAL_FAILURE_DETECTED}		Basic LC
Eu.LC.3013	Def	entry/d61out_LC_Failure_Status := "No critical failure present";{State-internal in NO_CRITICAL_FAILURE_DETECTED}		Basic LC
Eu.LC.3800	Def	NO_NON_CRITICAL_FAILURE_DETECTED		Basic LC
Eu.LC.3801	Def	when(D60in_LC_Failure)[D69in_Con_LC_Failure_Status_After_Present_Failure = "non critical failure report"]/{NO_NON_CRITICAL_FAILURE_DETECTED - NON_CRITICAL_FAILURE_DETECTED}		Basic LC
Eu.LC.3833	Def	entry/d61out_LC_Failure_Status := "No non critical failure present";{State-internal in NO_NON_CRITICAL_FAILURE_DETECTED}		Basic LC
Eu.LC.3802	Def	NON_CRITICAL_FAILURE_DETECTED		Basic LC
Eu.LC.3803	Def	when(NOT D60in_LC_Failure)[D69in_Con_LC_Failure_Status_After_Present_Failure = "non critical failure report"]/{NON_CRITICAL_FAILURE_DETECTED - NO_NON_CRITICAL_FAILURE_DETECTED}		Basic LC
Eu.LC.3834	Def	entry/d61out_LC_Failure_Status := "A non critical failure is present";{State-internal in NON_CRITICAL_FAILURE_DETECTED}		Basic LC
Eu.LC.3831	Def	Junction0		Basic LC
Eu.LC.3832	Def	[D60in_LC_Failure AND D69in_Con_LC_Failure_Status_After_Present_Failure = "critical failure report"]/{Junction0 - CRITICAL_FAILURE_DETECTED}		Basic LC
Eu.LC.2519	Def	[D69in_Con_LC_Failure_Status_After_Present_Failure = "critical failure report" AND NOT D60in_LC_Failure]/{Junction0 - NO_CRITICAL_FAILURE_DETECTED}		Basic LC
Eu.LC.3807	Def	[else]/{Junction0 - NO_NON_CRITICAL_FAILURE_DETECTED}		Basic LC
Eu.LC.3805	Def	[D60in_LC_Failure AND D69in_Con_LC_Failure_Status_After_Present_Failure = "non critical failure report"]/{Junction0 - NON_CRITICAL_FAILURE_DETECTED}		Basic LC
Eu.LC.2520	Def	LCPF_FAILURE_STATE		Basic LC
Eu.LC.2521	Def	CRITICAL_FAILURE_DETECTED		Basic LC
Eu.LC.2522	Def	when(d74in_LCPF_Failure_State = "No failure present")[D67in_Con_LCPF_Failure_Status_After_Present_Failure = "critical failure report"]/{CRITICAL_FAILURE_DETECTED - NO_CRITICAL_FAILURE_DETECTED}		Basic LC
Eu.LC.3016	Def	entry/d62out_LCPF_Failure_Status := "A critical failure is present";{State-internal in CRITICAL_FAILURE_DETECTED}		Basic LC
Eu.LC.2523	Def	Initial2		Basic LC
Eu.LC.3274	Def	/{Initial2 - Junction1}		Basic LC
Eu.LC.2525	Def	NO_CRITICAL_FAILURE_DETECTED		Basic LC
Eu.LC.2526	Def	when(d74in_LCPF_Failure_State = "Failure present")[D67in_Con_LCPF_Failure_Status_After_Present_Failure = "critical failure report"]/{NO_CRITICAL_FAILURE_DETECTED - CRITICAL_FAILURE_DETECTED}		Basic LC
Eu.LC.3017	Def	entry/d62out_LCPF_Failure_Status := "No critical failure present";{State-internal in NO_CRITICAL_FAILURE_DETECTED}		Basic LC
Eu.LC.3808	Def	NO_NON_CRITICAL_FAILURE_DETECTED		Basic LC
Eu.LC.3809	Def	when(d74in_LCPF_Failure_State = "Failure present")[D67in_Con_LCPF_Failure_Status_After_Present_Failure = "non critical failure report"]/{NO_NON_CRITICAL_FAILURE_DETECTED - NON_CRITICAL_FAILURE_DETECTED}		Basic LC
Eu.LC.3836	Def	entry/d62out_LCPF_Failure_Status := "No non critical failure present";{State-internal in NO_NON_CRITICAL_FAILURE_DETECTED}		Basic LC
Eu.LC.3810	Def	NON_CRITICAL_FAILURE_DETECTED		Basic LC
Eu.LC.3811	Def	when(d74in_LCPF_Failure_State = "No failure present")[D67in_Con_LCPF_Failure_Status_After_Present_Failure = "non critical failure report"]/{NON_CRITICAL_FAILURE_DETECTED - NO_NON_CRITICAL_FAILURE_DETECTED}		Basic LC
Eu.LC.3837	Def	entry/d62out_LCPF_Failure_Status := "A non critical failure is present";{State-internal in NON_CRITICAL_FAILURE_DETECTED}		Basic LC
Eu.LC.3835	Def	Junction1		Basic LC
Eu.LC.3814	Def	[d74in_LCPF_Failure_State = "Failure present" AND D67in_Con_LCPF_Failure_Status_After_Present_Failure = "critical failure report"]/{Junction1 - CRITICAL_FAILURE_DETECTED}		Basic LC
Eu.LC.2535	Def	[d74in_LCPF_Failure_State = "No failure present" AND D67in_Con_LCPF_Failure_Status_After_Present_Failure = "critical failure report"]/{Junction1 - NO_CRITICAL_FAILURE_DETECTED}		Basic LC
Eu.LC.3815	Def	[else]/{Junction1 - NO_NON_CRITICAL_FAILURE_DETECTED}		Basic LC
Eu.LC.3813	Def	[d74in_LCPF_Failure_State = "Failure present" AND D67in_Con_LCPF_Failure_Status_After_Present_Failure = "non critical failure report"]/{Junction1 - NON_CRITICAL_FAILURE_DETECTED}		Basic LC
Eu.LC.3838	Def	when(d51in_EST_EfeS_State = "FALLBACK_MODE")/{OBSERVE_FAILURE_STATES - WAITING_FOR_BOOTING}		Basic LC
Eu.LC.3839	Def	WAITING_FOR_BOOTING		Basic LC
Eu.LC.3840	Def	when(d51in_EST_EfeS_State = "BOOTING")/{WAITING_FOR_BOOTING - OBSERVE_FAILURE_STATES}		Basic LC
Eu.LC.2536	Info	F_Observe_LCPF		Basic LC

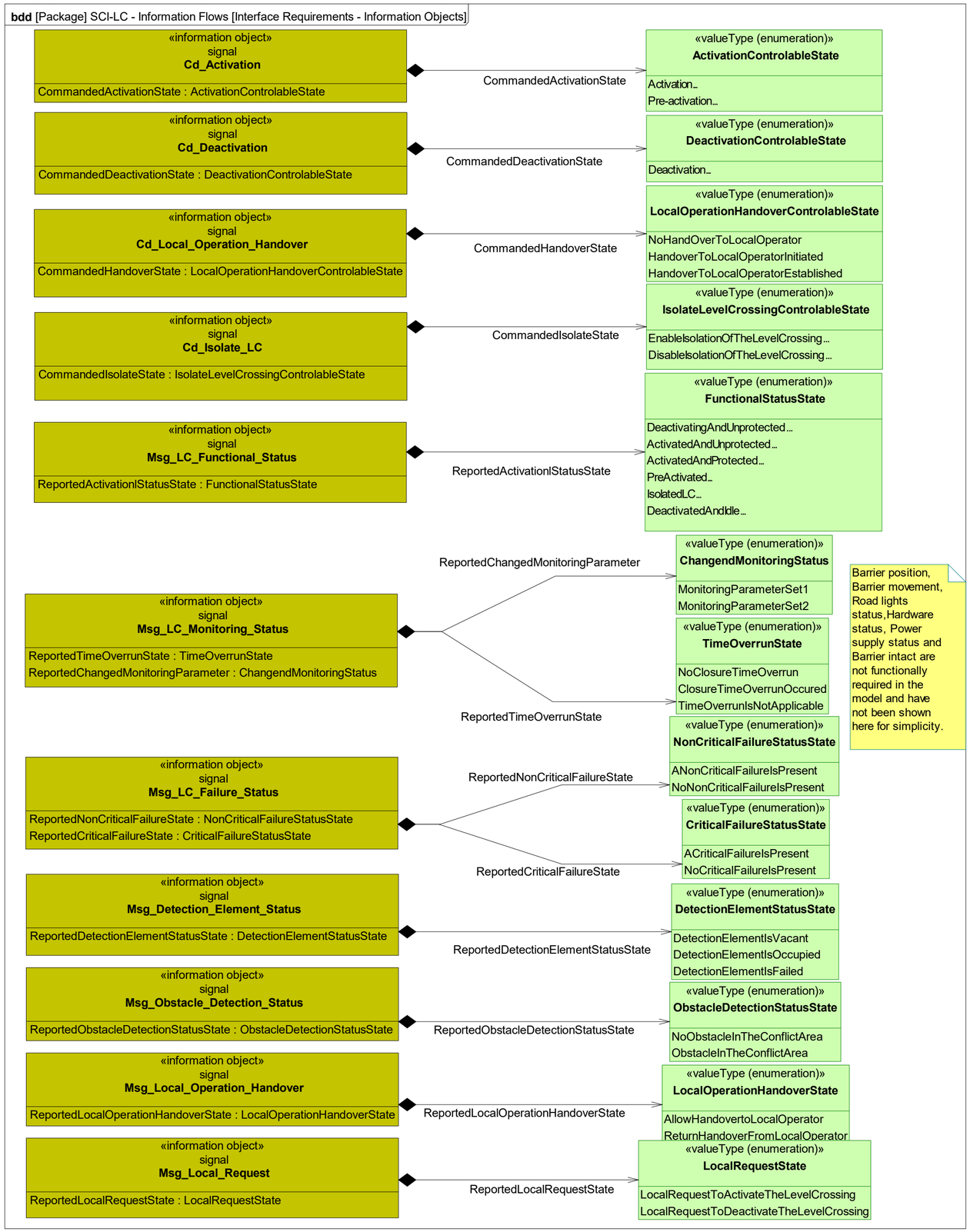
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.2537	Req	[Block] F_Observe_LCPF [Functional Viewpoint - Subsystem Requirements - Functional Entity]  		Basic LC
Eu.LC.2538	Def	D109in_Con_Use_Obstacle_Detection	The port D109in_Con_Use_Obstacle_Detection provides configuration values for the obstacle detection. true: Obstacle detection is used false: Obstacle detection is not used	Basic LC
Eu.LC.2539	Def	D30in_LCPF_Protection_State	The port D30in_LCPF_Protection_State refines the Informationflow Status_Level_Crossing_Protection_Facility.	Basic LC
Eu.LC.2540	Def	d3out_LCPF_Protection_State		Basic LC
Eu.LC.2541	Def	d4out_Obstacle_Detection_Status		Basic LC
Eu.LC.2542	Def	d74out_LCPF_Failure_State		Basic LC
Eu.LC.2543	Def	d9out_Changed_Monitoring_State		Basic LC
Eu.LC.3816	Def	D31in_Obstacle_Detection_State	The port D31in_Obstacle_Detection_State represents a detected Obstacle. true: Obstacle detected in the conflict area false: No Obstacle detected in the conflict area	Basic LC
Eu.LC.3817	Def	D32in_LCPF_Failure_State	The port D32in_LCPF_Failure_State represents a failure in the LCPF. true: Failure is present false: Failure is not present	Basic LC
Eu.LC.3818	Def	D33in_Monitoring_State	The port D33in_Monitoring_State provides values for the monitoring status. 1.: Monitoring Parameter Set 1 2.: Monitoring Parameter Set 2	Basic LC
Eu.LC.2544	Info	F_Observe_LCPF - Behaviour		Basic LC

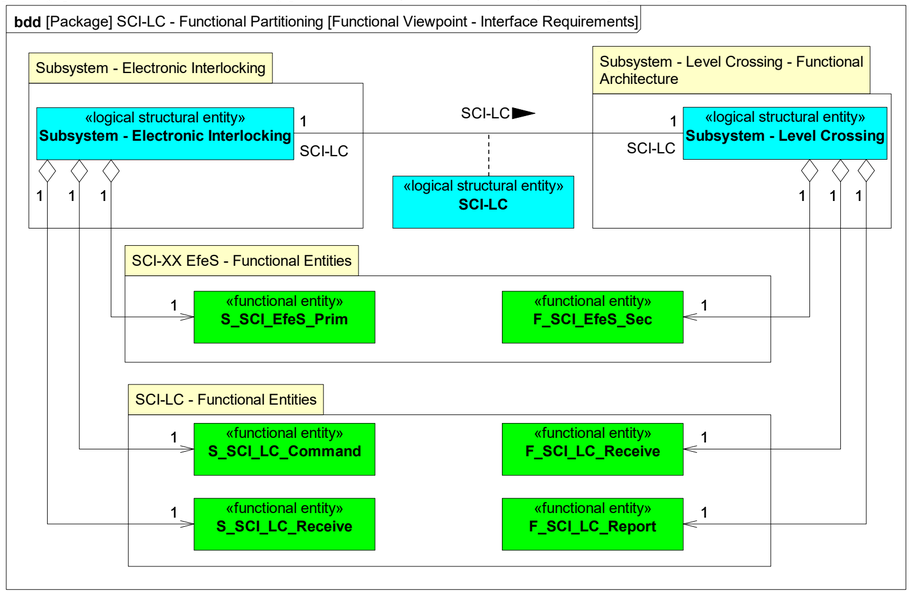
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.2545	Req	Functional Viewpoint - Subsystem Requirements - Functional Entity STD 4 	This state machine diagram describes the requirements for the following functionalities: - receive and forward the LCPF protection state - observes and forward the obstacle detection state - observes and forward the LCPF failure state - observes and forward the monitoring status	Basic LC
Eu.LC.2546	Def	Initial0		Basic LC
Eu.LC.2547	Def	/{Initial0 - OBSERVE_LCPF}		Basic LC
Eu.LC.2548	Def	OBSERVE_LCPF		Basic LC
Eu.LC.2549	Def	OBSERVE_LCPF_FAILURE_STATE		Basic LC
Eu.LC.2550	Def	FAILURE_DETECTED		Basic LC
Eu.LC.2551	Def	when(NOT D32in_LCPF_Failure_State)/{FAILURE_DETECTED - NO_FAILURE_DETECTED}		Basic LC
Eu.LC.3020	Def	entry/d74out_LCPF_Failure_State := "Failure present";{State-internal in FAILURE_DETECTED}		Basic LC

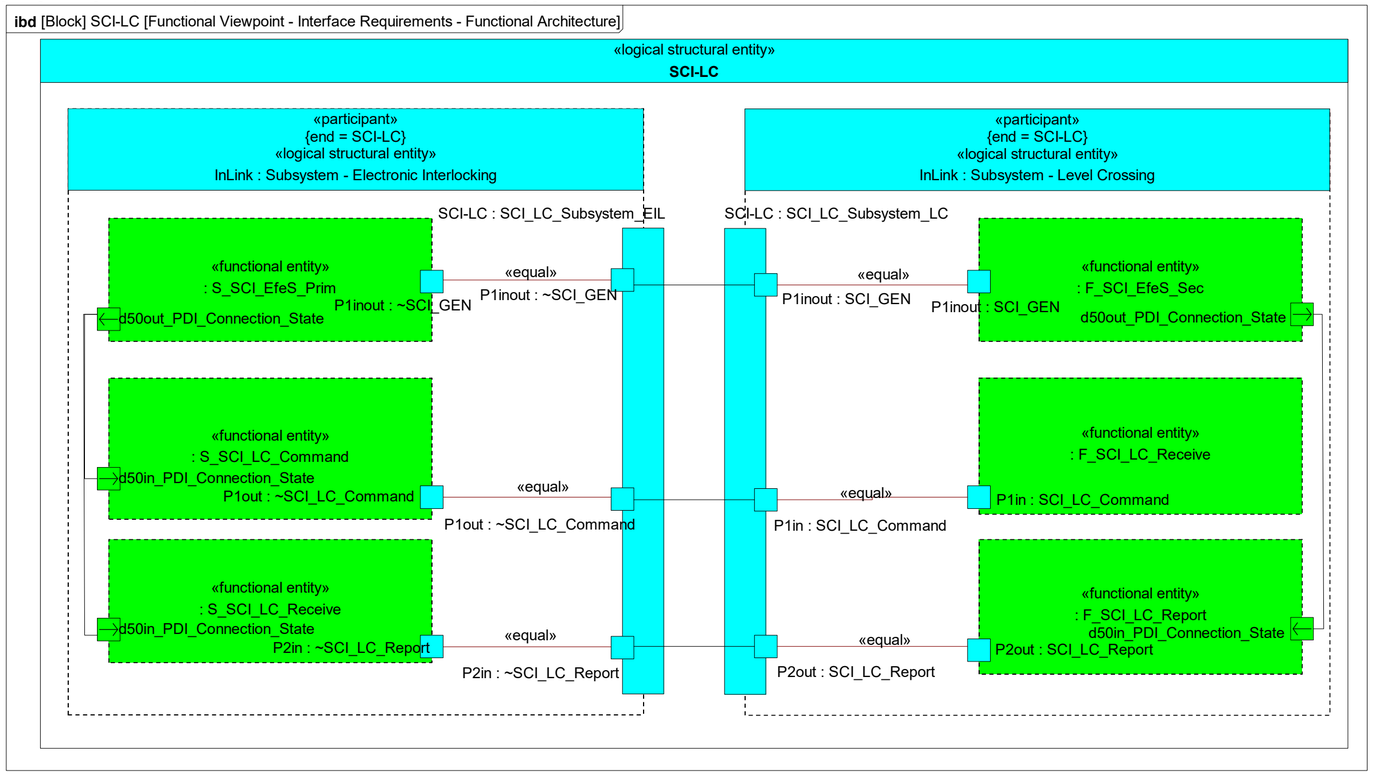
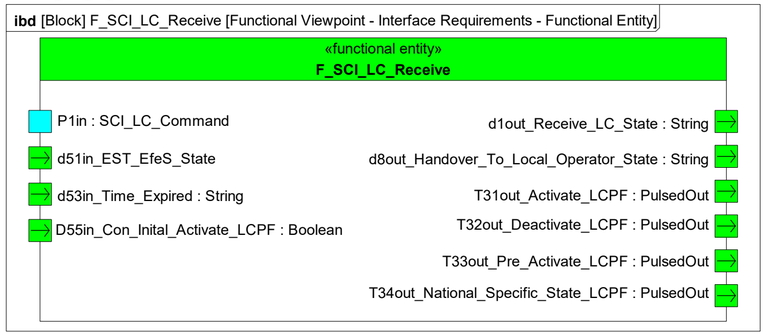
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.2552	Def	Initial3		Basic LC
Eu.LC.2553	Def	/{Initial3 - WAITING}		Basic LC
Eu.LC.2554	Def	NO_FAILURE_DETECTED		Basic LC
Eu.LC.2555	Def	when(D32in_LCPF_Failure_State)/{NO_FAILURE_DETECTED - FAILURE_DETECTED}		Basic LC
Eu.LC.3021	Def	entry/d74out_LCPF_Failure_State := "No failure present";{State-internal in NO_FAILURE_DETECTED}		Basic LC
Eu.LC.2556	Def	WAITING		Basic LC
Eu.LC.2557	Def	when(D32in_LCPF_Failure_State)/{WAITING - FAILURE_DETECTED}		Basic LC
Eu.LC.2558	Def	when(NOT D32in_LCPF_Failure_State)/{WAITING - NO_FAILURE_DETECTED}		Basic LC
Eu.LC.2559	Def	OBSERVE_LCPF_OBSTACLE_DETECTION_STATE		Basic LC
Eu.LC.2560	Def	Initial2		Basic LC
Eu.LC.2561	Def	/{Initial2 - WAITING}		Basic LC
Eu.LC.2562	Def	NO_OBSTACLE_IN_CONFLICT_AREA		Basic LC
Eu.LC.2563	Def	when(D31in_Obstacle_Detection_State)[D109in_Con_Use_Obstacle_Detection]/{NO_OBSTACLE_IN_CONFLICT_AREA - OBSTACLE_IN_CONFLICT_AREA}		Basic LC
Eu.LC.3022	Def	entry/d4out_Obstacle_Detection_Status := "No obstacle in the conflict area";{State-internal in NO_OBSTACLE_IN_CONFLICT_AREA}		Basic LC
Eu.LC.2564	Def	OBSTACLE_IN_CONFLICT_AREA		Basic LC
Eu.LC.2565	Def	when(NOT D31in_Obstacle_Detection_State)[D109in_Con_Use_Obstacle_Detection]/{OBSTACLE_IN_CONFLICT_AREA - NO_OBSTACLE_IN_CONFLICT_AREA}		Basic LC
Eu.LC.3023	Def	entry/d4out_Obstacle_Detection_Status := "Obstacle detected in the conflict area";{State-internal in OBSTACLE_IN_CONFLICT_AREA}		Basic LC
Eu.LC.2566	Def	WAITING		Basic LC
Eu.LC.2567	Def	when(NOT D31in_Obstacle_Detection_State)[D109in_Con_Use_Obstacle_Detection]/{WAITING - NO_OBSTACLE_IN_CONFLICT_AREA}		Basic LC
Eu.LC.2568	Def	when(D31in_Obstacle_Detection_State)[D109in_Con_Use_Obstacle_Detection]/{WAITING - OBSTACLE_IN_CONFLICT_AREA}		Basic LC
Eu.LC.2569	Def	OBSERVE_LCPF_PROTECTION_STATE		Basic LC
Eu.LC.2570	Def	IDLE		Basic LC
Eu.LC.2571	Def	when(D30in_LCPF_Protection_State = "Protected")/{IDLE - PROTECTED}		Basic LC
Eu.LC.2572	Def	when(D30in_LCPF_Protection_State = "Unprotected")/{IDLE - UNPROTECTED}		Basic LC
Eu.LC.3024	Def	entry/d3out_LCPF_Protection_State := "Idle";{State-internal in IDLE}		Basic LC
Eu.LC.2573	Def	Initial1		Basic LC
Eu.LC.2574	Def	/{Initial1 - WAITING}		Basic LC
Eu.LC.2575	Def	PROTECTED		Basic LC
Eu.LC.2576	Def	when(D30in_LCPF_Protection_State = "Idle")/{PROTECTED - IDLE}		Basic LC
Eu.LC.2577	Def	when(D30in_LCPF_Protection_State = "Unprotected")/{PROTECTED - UNPROTECTED}		Basic LC
Eu.LC.3025	Def	entry/d3out_LCPF_Protection_State := "Protected";{State-internal in PROTECTED}		Basic LC
Eu.LC.2578	Def	UNPROTECTED		Basic LC
Eu.LC.2579	Def	when(D30in_LCPF_Protection_State = "Idle")/{UNPROTECTED - IDLE}		Basic LC
Eu.LC.2580	Def	when(D30in_LCPF_Protection_State = "Protected")/{UNPROTECTED - PROTECTED}		Basic LC
Eu.LC.3026	Def	entry/d3out_LCPF_Protection_State := "Unprotected";{State-internal in UNPROTECTED}		Basic LC
Eu.LC.4247	Def	WAITING		Basic LC
Eu.LC.4236	Def	when(D30in_LCPF_Protection_State = "Idle")/{WAITING - IDLE}		Basic LC
Eu.LC.4237	Def	when(D30in_LCPF_Protection_State = "Protected")/{WAITING - PROTECTED}		Basic LC
Eu.LC.4238	Def	when(D30in_LCPF_Protection_State = "Unprotected")/{WAITING - UNPROTECTED}		Basic LC
Eu.LC.2581	Def	OBSERVE_MONITORING_STATUS		Basic LC
Eu.LC.2582	Def	MONITORING_PARAMETER_SET_2		Basic LC
Eu.LC.2583	Def	when(D33in_Monitoring_State = "Monitoring Parameter Set 1")/{MONITORING_PARAMETER_SET_2 - MONITORING_PARAMETER_SET_1}		Basic LC
Eu.LC.3027	Def	entry/d9out_Changed_Monitoring_State := "Monitoring Parameter Set 2";{State-internal in MONITORING_PARAMETER_SET_2}		Basic LC
Eu.LC.2584	Def	Initial4		Basic LC
Eu.LC.2585	Def	/{Initial4 - MONITORING_PARAMETER_SET_1}		Basic LC
Eu.LC.2586	Def	MONITORING_PARAMETER_SET_1		Basic LC
Eu.LC.2587	Def	when(D33in_Monitoring_State = "Monitoring Parameter Set 2")/{MONITORING_PARAMETER_SET_1 - MONITORING_PARAMETER_SET_2}		Basic LC
Eu.LC.3221	Def	entry/d9out_Changed_Monitoring_State := "Monitoring Parameter Set 1";{State-internal in MONITORING_PARAMETER_SET_1}		Basic LC

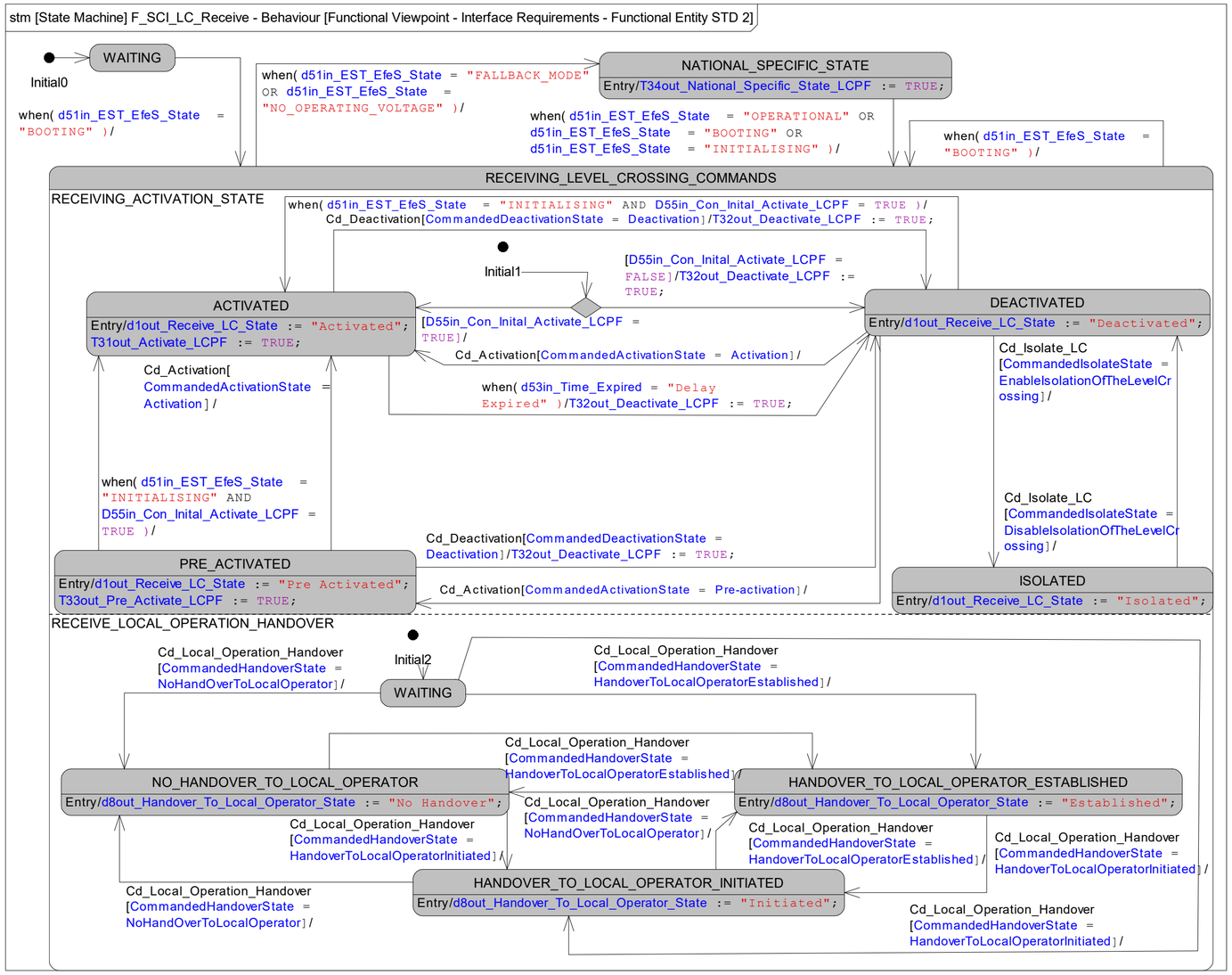
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.85	Head	**3.4 Subsystem - Level Crossing - Interfaces**		
Eu.LC.86	Head	**3.4.1 SCI-LC (Subsystem - Electronic Interlocking)**		
Eu.LC.2990	Head	**3.4.1.1 SCI-LC - Logical Viewpoint**		
Eu.LC.3262	Head	**3.4.1.1.1 SCI-LC - Logical Context**		
Eu.LC.2998	Def	[Package] SCI-LC - Logical Context [Logical Viewpoint - Interface Definition]  	Process Data Interface SCI-LC	Basic LC
Eu.LC.2755	Head	**3.4.1.2 SCI-LC - Information Flows**		
Eu.LC.87	Info	The generic commands and messages through the SCI_LC_Subsystem_EIL are specified in [Eu.Doc.119]		Basic LC
Eu.LC.2997	Def	[Package] SCI-LC - Information Flows [Interface Requirements - Direction of Information Objects]  		Basic LC

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.2756	Def	[Package] SCI-LC - Information Flows [Interface Requirements - Information Objects]		Basic LC



ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.2757	Def	Cd_Activation		Basic LC
Eu.LC.2758	Def	Cd_Deactivation		Basic LC
Eu.LC.2759	Def	Cd_Isolate_LC		Basic LC
Eu.LC.2760	Def	Cd_Local_Operation_Handover		Option LOH
Eu.LC.2761	Def	Msg_Detection_Element_Status		Basic LC
Eu.LC.2762	Def	Msg_LC_Failure_Status		Basic LC
Eu.LC.2763	Def	Msg_LC_Functional_Status		Basic LC
Eu.LC.2764	Def	Msg_LC_Monitoring_Status		Basic LC
Eu.LC.2765	Def	Msg_Local_Operation_Handover		Option LOH
Eu.LC.2766	Def	Msg_Local_Request		Option LOH
Eu.LC.2767	Def	Msg_Obstacle_Detection_Status		Basic LC
Eu.LC.2589	Head	**3.4.1.3 SCI-LC - Functional Viewpoint**		
Eu.LC.3289	Head	**3.4.1.3.1 SCI-LC - Functional Partitioning**		
Eu.LC.2991	Def	[Package] SCI-LC - Functional Partitioning [Functional Viewpoint - Interface Requirements]  	Process Data Interface SCI-LC	Basic LC
Eu.LC.2993	Head	**3.4.1.3.2 SCI-LC - Functional Architecture**		
Eu.LC.2994	Info	SCI-LC		Basic LC

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.2995	Def	[Block] SCI-LC [Functional Viewpoint - Interface Requirements - Functional Architecture]  		Basic LC
Eu.LC.2591	Head	**3.4.1.3.3 SCI-LC - Functional Entities**		
Eu.LC.2592	Info	F_SCI_LC_Receive		Basic LC
Eu.LC.3089	Req	[Block] F_SCI_LC_Receive [Functional Viewpoint - Interface Requirements - Functional Entity]  		Basic LC
Eu.LC.3282	Def	T31out_Activate_LCPF	The port T31_Activate_LCPF refines the Flow Property Activate.	Basic LC
Eu.LC.3283	Def	T32out_Deactivate_LCPF	The port T32_Deactivate_LCPF refines the Flow Property Deactivate.	Basic LC
Eu.LC.3284	Def	T33out_Pre_Activate_LCPF	The port T33_Pre_Activate_LCPF refines the Flow Property Pre-Activate.	Basic LC
Eu.LC.3285	Def	T34out_National_Specific_State_LCPF	The port T34_National_Specific_State_LCPF refines the Flow Property National_Specific_State.	Basic LC
Eu.LC.2633	Def	P1in	The port P1in exchanges information objects according to SCI_LC_Subsystem_LC.	Basic LC
Eu.LC.3276	Def	d51in_EST_EfeS_State		Basic LC
Eu.LC.2593	Def	d1out_Receive_LC_State		Basic LC
Eu.LC.2596	Def	d53in_Time_Expired		Basic LC

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.2597	Def	d8out_Handover_To_Local_Operator_State		Option LOH
Eu.LC.4081	Def	D55in_Con_Inital_Activate_LCPF	The port D55in_Con_Inital_Activate_LCPF provides configuration values for the initial activation or deactivation of the Level Crossing protection facility. true: Level Crossing protection facility is initial activated. false: Level Crossing protection facility is initial deactivated.	Basic LC
Eu.LC.2598	Info	F_SCI_LC_Receive - Behaviour		Basic LC
Eu.LC.2599	Req	Functional Viewpoint - Interface Requirements - Functional Entity STD 2  	This state machine diagram describes the requirements for the following functionalities: - receives the commanded activation state and forwards it to the internal logic of the Subsystem - Level Crossing - receives the commanded local operation handover and forwards it to the internal logic of the Subsystem - Level Crossing	Basic LC Option LOH
Eu.LC.2600	Def	Initial0		Basic LC
Eu.LC.2601	Def	/{Initial0 - WAITING}		Basic LC
Eu.LC.2602	Def	RECEIVING_LEVEL_CROSSING_COMMANDS		Basic LC

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.2603	Def	RECEIVE_LOCAL_OPERATION_HANDOVER		Option LOH
Eu.LC.2604	Def	HANDOVER_TO_LOCAL_OPERATOR_ESTABLISHED		Option LOH
Eu.LC.2605	Def	Cd_Local_Operation_Handover[CommandedHandoverState = HandoverToLocalOperatorInitiated]/{HANDOVER_TO_LOCAL_OPERATOR_ESTABLISHED - HANDOVER_TO_LOCAL_OPERATOR_INITIATED}		Option LOH
Eu.LC.2606	Def	Cd_Local_Operation_Handover[CommandedHandoverState = NoHandOverToLocalOperator]/{HANDOVER_TO_LOCAL_OPERATOR_ESTABLISHED - NO_HANDOVER_TO_LOCAL_OPERATOR}		Option LOH
Eu.LC.3028	Def	entry/d8out_Handover_To_Local_Operator_State := "Established";{State-internal in HANDOVER_TO_LOCAL_OPERATOR_ESTABLISHED}		Option LOH
Eu.LC.2607	Def	HANDOVER_TO_LOCAL_OPERATOR_INITIATED		Option LOH
Eu.LC.2608	Def	Cd_Local_Operation_Handover[CommandedHandoverState = HandoverToLocalOperatorEstablished]/{HANDOVER_TO_LOCAL_OPERATOR_INITIATED - HANDOVER_TO_LOCAL_OPERATOR_ESTABLISHED}		Option LOH
Eu.LC.2609	Def	Cd_Local_Operation_Handover[CommandedHandoverState = NoHandOverToLocalOperator]/{HANDOVER_TO_LOCAL_OPERATOR_INITIATED - NO_HANDOVER_TO_LOCAL_OPERATOR}		Option LOH
Eu.LC.3029	Def	entry/d8out_Handover_To_Local_Operator_State := "Initiated";{State-internal in HANDOVER_TO_LOCAL_OPERATOR_INITIATED}		Option LOH
Eu.LC.2610	Def	Initial2		Option LOH
Eu.LC.2611	Def	/{Initial2 - WAITING}		Option LOH
Eu.LC.2612	Def	NO_HANDOVER_TO_LOCAL_OPERATOR		Option LOH
Eu.LC.2613	Def	Cd_Local_Operation_Handover[CommandedHandoverState = HandoverToLocalOperatorEstablished]/{NO_HANDOVER_TO_LOCAL_OPERATOR - HANDOVER_TO_LOCAL_OPERATOR_ESTABLISHED}		Option LOH
Eu.LC.2614	Def	Cd_Local_Operation_Handover[CommandedHandoverState = HandoverToLocalOperatorInitiated]/{NO_HANDOVER_TO_LOCAL_OPERATOR - HANDOVER_TO_LOCAL_OPERATOR_INITIATED}		Option LOH
Eu.LC.3030	Def	entry/d8out_Handover_To_Local_Operator_State := "No Handover";{State-internal in NO_HANDOVER_TO_LOCAL_OPERATOR}		Option LOH
Eu.LC.2615	Def	WAITING		Option LOH
Eu.LC.2616	Def	Cd_Local_Operation_Handover[CommandedHandoverState = HandoverToLocalOperatorEstablished]/{WAITING - HANDOVER_TO_LOCAL_OPERATOR_ESTABLISHED}		Option LOH
Eu.LC.2617	Def	Cd_Local_Operation_Handover[CommandedHandoverState = HandoverToLocalOperatorInitiated]/{WAITING - HANDOVER_TO_LOCAL_OPERATOR_INITIATED}		Option LOH
Eu.LC.2618	Def	Cd_Local_Operation_Handover[CommandedHandoverState = NoHandOverToLocalOperator]/{WAITING - NO_HANDOVER_TO_LOCAL_OPERATOR}		Option LOH
Eu.LC.2619	Def	RECEIVING_ACTIVATION_STATE		Basic LC
Eu.LC.2620	Def	ACTIVATED		Basic LC
Eu.LC.2621	Def	Cd_Deactivation[CommandedDeactivationState = Deactivation]/T32out_Deactivate_LCPF := TRUE;{ACTIVATED - DEACTIVATED}		Basic LC
Eu.LC.2622	Def	when(d53in_Time_Expired = "Delay Expired")/T32out_Deactivate_LCPF := TRUE;{ACTIVATED - DEACTIVATED}		Basic LC
Eu.LC.3031	Def	entry/d1out_Receive_LC_State := "Activated"; T31out_Activate_LCPF := TRUE;{State-internal in ACTIVATED}		Basic LC
Eu.LC.2623	Def	DEACTIVATED		Basic LC
Eu.LC.2624	Def	Cd_Activation[CommandedActivationState = Activation]/{DEACTIVATED - ACTIVATED}		Basic LC
Eu.LC.2625	Def	Cd_Activation[CommandedActivationState = Pre-activation]/{DEACTIVATED - PRE_ACTIVATED}		Basic LC
Eu.LC.2626	Def	Cd_Isolate_LC[CommandedIsolateState = EnableIsolationOfTheLevelCrossing]/{DEACTIVATED - ISOLATED}		Basic LC
Eu.LC.3032	Def	entry/d1out_Receive_LC_State := "Deactivated";{State-internal in DEACTIVATED}		Basic LC
Eu.LC.3296	Def	when(d51in_EST_EfeS_State = "INITIALISING" AND D55in_Con_Inital_Activate_LCPF = TRUE)/{DEACTIVATED - ACTIVATED}		Basic LC
Eu.LC.2627	Def	Initial1		Basic LC
Eu.LC.2628	Def	/{Initial1 - Junction}		Basic LC
Eu.LC.2629	Def	ISOLATED		Basic LC
Eu.LC.2630	Def	Cd_Isolate_LC[CommandedIsolateState = DisableIsolationOfTheLevelCrossing]/{ISOLATED - DEACTIVATED}		Basic LC
Eu.LC.3033	Def	entry/d1out_Receive_LC_State := "Isolated";{State-internal in ISOLATED}		Basic LC
Eu.LC.2631	Def	PRE_ACTIVATED		Basic LC
Eu.LC.2632	Def	Cd_Activation[CommandedActivationState = Activation]/{PRE_ACTIVATED - ACTIVATED}		Basic LC

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.3034	Def	entry/d1out_Receive_LC_State := "Pre Activated"; T33out_Pre_Activate_LCPF := TRUE;{State-internal in PRE_ACTIVATED}		Basic LC
Eu.LC.3224	Def	Cd_Deactivation[CommandedDeactivationState = Deactivation]/T32out_Deactivate_LCPF := TRUE;{PRE_ACTIVATED - DEACTIVATED}		Basic LC
Eu.LC.3297	Def	when(d51in_EST_EfeS_State = "INITIALISING" AND D55in_Con_Inital_Activate_LCPF = TRUE)/{PRE_ACTIVATED - ACTIVATED}		Basic LC
Eu.LC.4083	Def	Junction		Basic LC
Eu.LC.4084	Def	[D55in_Con_Inital_Activate_LCPF = TRUE]/{Junction - ACTIVATED}		Basic LC
Eu.LC.4085	Def	[D55in_Con_Inital_Activate_LCPF = FALSE]/T32out_Deactivate_LCPF := TRUE;{Junction - DEACTIVATED}		Basic LC
Eu.LC.3280	Def	when(d51in_EST_EfeS_State = "BOOTING")/{RECEIVING_LEVEL_CROSSING_COMMANDS - RECEIVING_LEVEL_CROSSING_COMMANDS}		Basic LC
Eu.LC.3281	Def	when(d51in_EST_EfeS_State = "FALLBACK_MODE" OR d51in_EST_EfeS_State = "NO_OPERATING_VOLTAGE")/{RECEIVING_LEVEL_CROSSING_COMMANDS - NATIONAL_SPECIFIC_STATE}		Basic LC
Eu.LC.3277	Def	NATIONAL_SPECIFIC_STATE		Basic LC
Eu.LC.3278	Def	entry/T34out_National_Specific_State_LCPF := TRUE;{State-internal in NATIONAL_SPECIFIC_STATE}		Basic LC
Eu.LC.3279	Def	when(d51in_EST_EfeS_State = "OPERATIONAL" OR d51in_EST_EfeS_State = "BOOTING" OR d51in_EST_EfeS_State = "INITIALISING")/{NATIONAL_SPECIFIC_STATE - RECEIVING_LEVEL_CROSSING_COMMANDS}		Basic LC
Eu.LC.3298	Def	WAITING		Basic LC
Eu.LC.3299	Def	when(d51in_EST_EfeS_State = "BOOTING")/{WAITING - RECEIVING_LEVEL_CROSSING_COMMANDS}		Basic LC
Eu.LC.2634	Info	F_SCI_LC_Report		Basic LC
Eu.LC.3090	Req	[Block] F_SCI_LC_Report [Functional Viewpoint - Interface Requirements - Functional Entity] 		Basic LC

Eu.LC.3225	Def	/* cOp1_Critical_Failure */
		
		if (d61in_LC_Failure_Status <> "A critical failure is present" AND d62in_LCPF_Failure_Status <> "A critical failure is present" AND d75in_LC_Failure_Status <> "A critical failure is present") then
		return CriticalFailureStatusState.NoCriticalFailureIsPresent;
		else
		return CriticalFailureStatusState.ACriticalFailureIsPresent;
		end if
			cOp1_Critical_Failure	Basic LC

Eu.LC.3227	Def	/* cOp2_Non_Critical_Failure */
		
		if (d61in_LC_Failure_Status <> "A non critical failure is present" AND d62in_LCPF_Failure_Status <> "A non critical failure is present" AND d75in_LC_Failure_Status <> "A non critical failure is present")then
		return NonCriticalFailureStatusState.NoNonCriticalFailureIsPresent;
		else
		return NonCriticalFailureStatusState.ANonCriticalFailureIsPresent;
		end if
			cOp2_Non_Critical_Failure	Basic LC

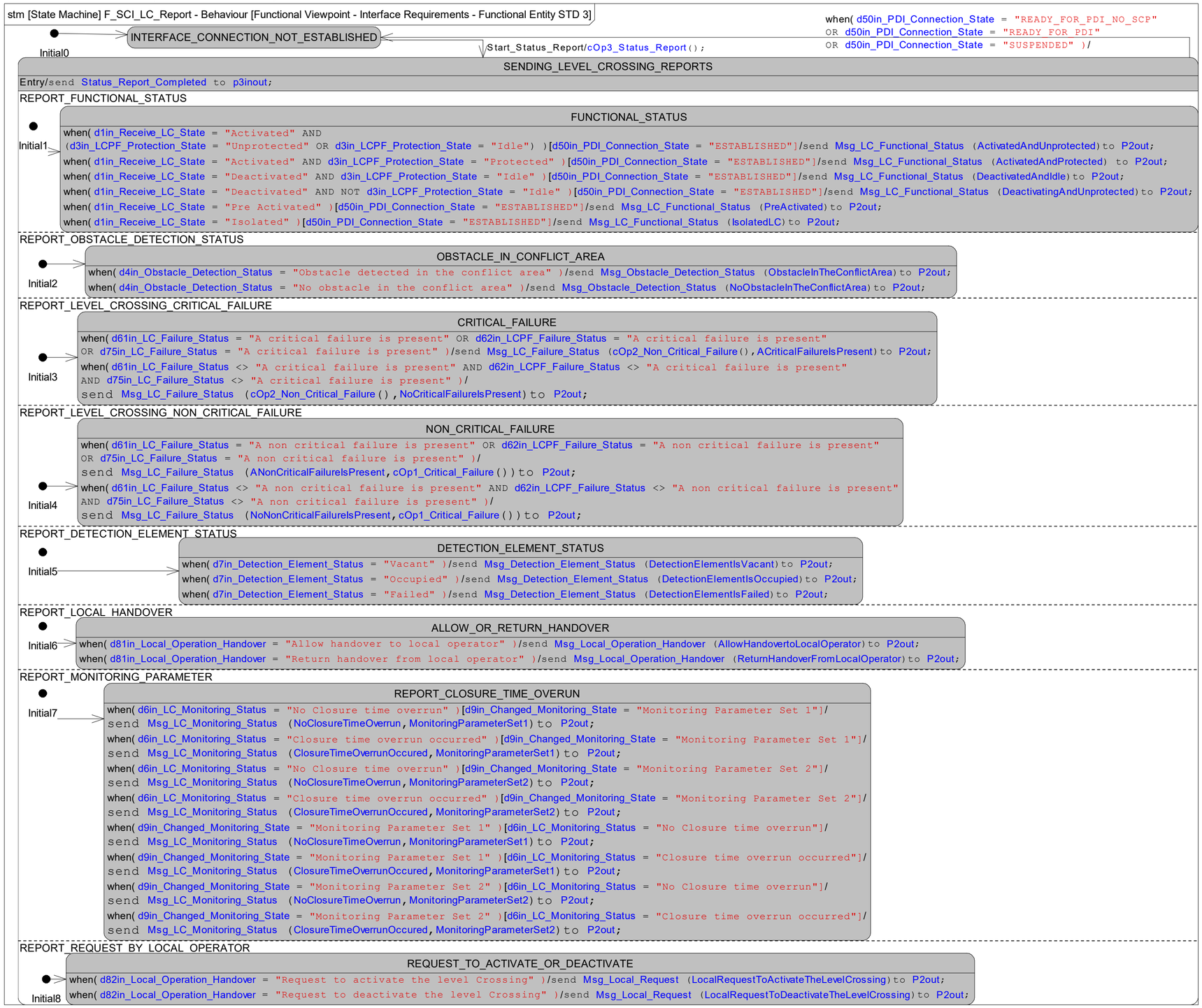
Eu.LC.3229	Def	/* cOp3_Status_Report */ if (d1in_Receive_LC_State = "Activated" AND (d3in_LCPF_Protection_State = "Unprotected" OR d3in_LCPF_Protection_State = "Idle")) then send Msg_LC_Functional_Status (FunctionalStatusState.ActivatedAndUnprotected)to P2out; elseif (d1in_Receive_LC_State = "Activated" AND d3in_LCPF_Protection_State = "Protected") then	cOp3_Status_Report	Basic LC

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.

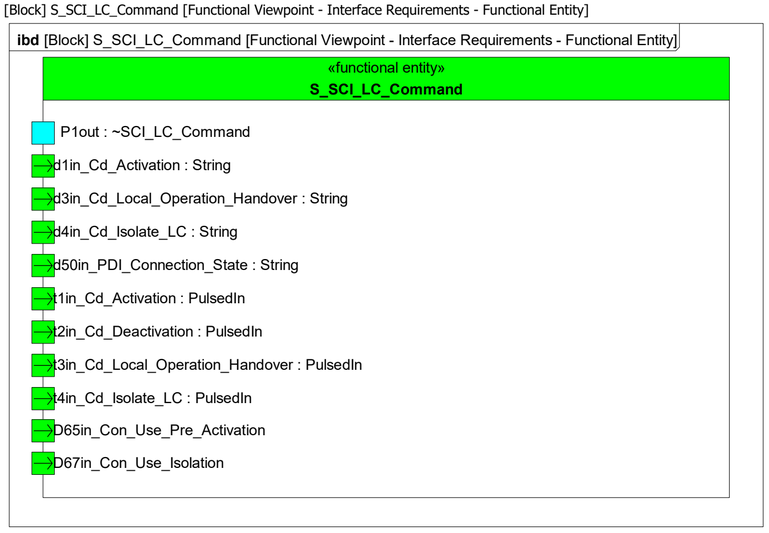
		send Msg_LC_Functional_Status (FunctionalStatusState.ActivatedAndProtected) to P2out;
		elseif (d1in_Receive_LC_State = "Isolated") then
		send Msg_LC_Functional_Status (FunctionalStatusState.IsolatedLC)to P2out;
		elseif (d1in_Receive_LC_State = "Pre Activated") then
		send Msg_LC_Functional_Status (FunctionalStatusState.PreActivated)to P2out;
		elseif (d1in_Receive_LC_State = "Deactivated" AND d3in_LCPF_Protection_State = "Idle") then
		send Msg_LC_Functional_Status (FunctionalStatusState.DeactivatedAndIdle)to P2out;
		else
		send Msg_LC_Functional_Status (FunctionalStatusState.DeactivatingAndUnprotected)to P2out;
		end if
		
		if (d4in_Obstacle_Detection_Status = "Obstacle detected in the conflict area") then
		send Msg_Obstacle_Detection_Status (ObstacleDetectionStatusState.ObstacleInTheConflictArea)to P2out;
		else
		send Msg_Obstacle_Detection_Status (ObstacleDetectionStatusState.NoObstacleInTheConflictArea)to P2out;
		end if
		
		if (d61in_LC_Failure_Status = "A critical failure is present" OR d62in_LCPF_Failure_Status = "A critical failure is present" OR d75in_LC_Failure_Status = "A critical failure is present") then
		send Msg_LC_Failure_Status (cOp2_Non_Critical_Failure(),CriticalFailureStatusState.ACriticalFailureIsPresent)to P2out;
		else
		send Msg_LC_Failure_Status (cOp2_Non_Critical_Failure(),CriticalFailureStatusState.NoCriticalFailureIsPresent)to P2out;
		end if
		
		if (d61in_LC_Failure_Status = "A non critical failure is present" OR d62in_LCPF_Failure_Status = "A non critical failure is present" OR d75in_LC_Failure_Status = "A non critical failure is present") then
		send Msg_LC_Failure_Status (NonCriticalFailureStatusState.ANonCriticalFailureIsPresent,cOp1_Critical_Failure())to P2out;
		else
		send Msg_LC_Failure_Status (NonCriticalFailureStatusState.NoNonCriticalFailureIsPresent,cOp1_Critical_Failure())to P2out;
		end if
		
		if (d6in_LC_Monitoring_Status = "No Closure time overrun" AND d9in_Changed_Monitoring_State = "Monitoring Parameter Set 1") then
		send Msg_LC_Monitoring_Status (TimeOverrunState.NoClosureTimeOverrun,ChangendMonitoringStatus.MonitoringParameterSet1)to P2out;
		elseif (d6in_LC_Monitoring_Status = "Closure time overrun occurred" AND d9in_Changed_Monitoring_State = "Monitoring Parameter Set 1") then
		send Msg_LC_Monitoring_Status (TimeOverrunState.ClosureTimeOverrunOccured,ChangendMonitoringStatus.MonitoringParameterSet1)to P2out;
		elseif (d6in_LC_Monitoring_Status = "No Closure time overrun" AND d9in_Changed_Monitoring_State <> "Monitoring Parameter Set 1") then
		send Msg_LC_Monitoring_Status (TimeOverrunState.NoClosureTimeOverrun,ChangendMonitoringStatus.MonitoringParameterSet2)to P2out;
		elseif (d6in_LC_Monitoring_Status = "Closure time overrun occurred" AND d9in_Changed_Monitoring_State <> "Monitoring Parameter Set 1") then
		send Msg_LC_Monitoring_Status (TimeOverrunState.ClosureTimeOverrunOccured,ChangendMonitoringStatus.MonitoringParameterSet2)to P2out;
		elseif (d9in_Changed_Monitoring_State = "Changed Monitoring Parameter" AND d6in_LC_Monitoring_Status = "Monitoring Parameter Set 1") then
		send Msg_LC_Monitoring_Status (TimeOverrunState.NoClosureTimeOverrun,ChangendMonitoringStatus.MonitoringParameterSet1)to P2out;
		else
		send Msg_LC_Monitoring_Status (TimeOverrunState.ClosureTimeOverrunOccured,ChangendMonitoringStatus.MonitoringParameterSet1)to P2out;
		end if
		
		if (d7in_Detection_Element_Status = "Vacant") then
		send Msg_Detection_Element_Status (DetectionElementStatusState.DetectionElementIsVacant)to P2out;
		send Status_Report_Completed to p3inout;
		elseif (d7in_Detection_Element_Status = "Occupied") then
		send Msg_Detection_Element_Status (DetectionElementStatusState.DetectionElementIsOccupied)to P2out;
		send Status_Report_Completed to p3inout;
		else
		send Msg_Detection_Element_Status (DetectionElementStatusState.DetectionElementIsFailed)to P2out;
		end if		

Eu.LC.2635	Def	d1in_Receive_LC_State		Basic LC
Eu.LC.2636	Def	d3in_LCPF_Protection_State		Basic LC
Eu.LC.2637	Def	d4in_Obstacle_Detection_Status		Basic LC
Eu.LC.2638	Def	d50in_PDI_Connection_State		Basic LC
Eu.LC.2639	Def	d61in_LC_Failure_Status		Basic LC
Eu.LC.2640	Def	d62in_LCPF_Failure_Status		Basic LC
Eu.LC.2641	Def	d6in_LC_Monitoring_Status		Basic LC
Eu.LC.2642	Def	d75in_LC_Failure_Status		Basic LC
Eu.LC.2643	Def	d7in_Detection_Element_Status		Basic LC
Eu.LC.2644	Def	d81in_Local_Operation_Handover		Basic LC
Eu.LC.2645	Def	d82in_Local_Operation_Handover		Basic LC
Eu.LC.2646	Def	d9in_Changed_Monitoring_State		Basic LC
Eu.LC.3251	Def	p3inout		Basic LC
Eu.LC.2716	Def	P2out	The port P2out exchanges information objects according to SCI_LC_Subsystem_LC.	Basic LC
Eu.LC.2647	Info	F_SCI_LC_Report - Behaviour		Basic LC

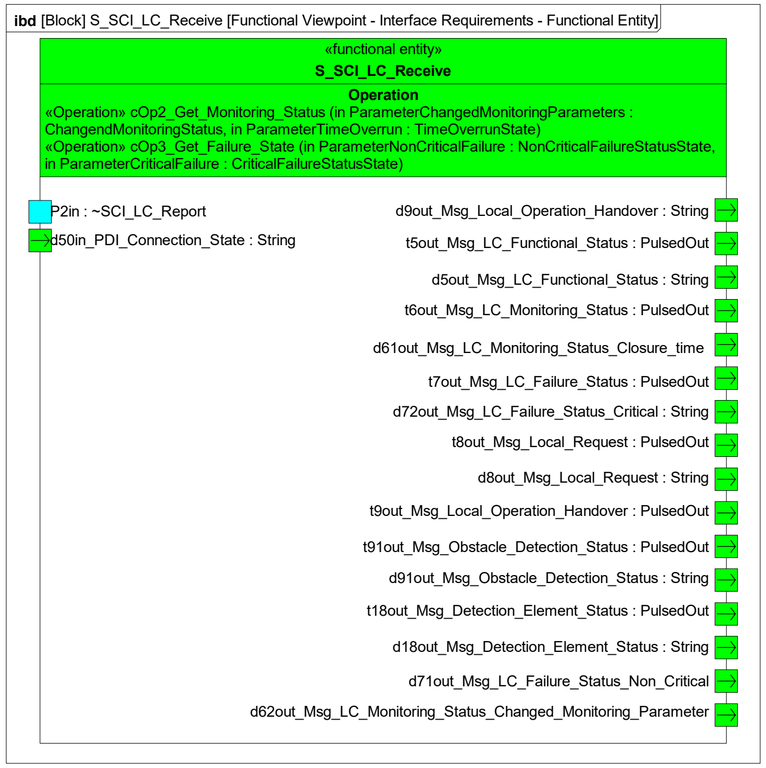
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.2648	Req	Functional Viewpoint - Interface Requirements - Functional Entity STD 3	This state machine diagram describes the requirements for the following functionalities: - receives the observed functional status from internal logic and reports this to the Subsystem - Electronic Interlocking - receives the observed obstacle detection status from internal logic and reports this to the Subsystem - Electronic Interlocking - receives the observed level crossing critical failure from internal logic and reports this to the Subsystem - Electronic Interlocking - receives the observed level crossing non critical failure from internal logic and reports this to the Subsystem - Electronic Interlocking - receives the observed detection element status from internal logic and reports this to the Subsystem - Electronic Interlocking - receives the observed local handover from internal logic and reports this to the Subsystem - Electronic Interlocking - receives the observed monitoring parameter from internal logic and reports this to the Subsystem - Electronic Interlocking - receives the observed request by local operator from internal logic and reports this to the Subsystem - Electronic Interlocking	Basic LC Option LOH



ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.2649	Def	Initial0		Basic LC
Eu.LC.2650	Def	/{Initial0 - INTERFACE_CONNECTION_NOT_ESTABLISHED}		Basic LC
Eu.LC.2651	Def	SENDING_LEVEL_CROSSING_REPORTS		Basic LC
Eu.LC.2656	Def	REPORT_DETECTION_ELEMENT_STATUS		Basic LC
Eu.LC.2657	Def	DETECTION_ELEMENT_STATUS		Basic LC
Eu.LC.3036	Def	when(d7in_Detection_Element_Status = "Failed")/send Msg_Detection_Element_Status (DetectionElementIsFailed)to P2out;{State-internal in DETECTION_ELEMENT_STATUS}		Basic LC
Eu.LC.3037	Def	when(d7in_Detection_Element_Status = "Occupied")/send Msg_Detection_Element_Status (DetectionElementIsOccupied)to P2out;{State-internal in DETECTION_ELEMENT_STATUS}		Basic LC
Eu.LC.3038	Def	when(d7in_Detection_Element_Status = "Vacant")/send Msg_Detection_Element_Status (DetectionElementIsVacant)to P2out;{State-internal in DETECTION_ELEMENT_STATUS}		Basic LC
Eu.LC.2658	Def	Initial5		Basic LC
Eu.LC.2659	Def	/{Initial5 - DETECTION_ELEMENT_STATUS}		Basic LC
Eu.LC.2660	Def	REPORT_FUNCTIONAL_STATUS		Basic LC
Eu.LC.2664	Def	FUNCTIONAL_STATUS		Basic LC
Eu.LC.3040	Def	when(d1in_Receive_LC_State = "Activated" AND (d3in_LCPF_Protection_State = "Unprotected" OR d3in_LCPF_Protection_State = "Idle"))[d50in_PDI_Connection_State = "ESTABLISHED"]/send Msg_LC_Functional_Status (ActivatedAndUnprotected)to P2out;{State-internal in FUNCTIONAL_STATUS}		Basic LC
Eu.LC.3234	Def	when(d1in_Receive_LC_State = "Activated" AND d3in_LCPF_Protection_State = "Protected")[d50in_PDI_Connection_State = "ESTABLISHED"]/send Msg_LC_Functional_Status (ActivatedAndProtected) to P2out;{State-internal in FUNCTIONAL_STATUS}		Basic LC
Eu.LC.3235	Def	when(d1in_Receive_LC_State = "Deactivated" AND d3in_LCPF_Protection_State = "Idle")[d50in_PDI_Connection_State = "ESTABLISHED"]/send Msg_LC_Functional_Status (DeactivatedAndIdle)to P2out;{State-internal in FUNCTIONAL_STATUS}		Basic LC
Eu.LC.3236	Def	when(d1in_Receive_LC_State = "Pre Activated")[d50in_PDI_Connection_State = "ESTABLISHED"]/send Msg_LC_Functional_Status (PreActivated)to P2out;{State-internal in FUNCTIONAL_STATUS}		Basic LC
Eu.LC.3237	Def	when(d1in_Receive_LC_State = "Deactivated" AND NOT d3in_LCPF_Protection_State = "Idle")[d50in_PDI_Connection_State = "ESTABLISHED"]/send Msg_LC_Functional_Status (DeactivatingAndUnprotected)to P2out;{State-internal in FUNCTIONAL_STATUS}		Basic LC
Eu.LC.4106	Def	when(d1in_Receive_LC_State = "Isolated")[d50in_PDI_Connection_State = "ESTABLISHED"]/send Msg_LC_Functional_Status (IsolatedLC)to P2out;{State-internal in FUNCTIONAL_STATUS}		Basic LC
Eu.LC.2672	Def	Initial1		Basic LC
Eu.LC.2673	Def	/{Initial1 - FUNCTIONAL_STATUS}		Basic LC
Eu.LC.2679	Def	REPORT_LEVEL_CROSSING_CRITICAL_FAILURE		Basic LC
Eu.LC.2682	Def	Initial3		Basic LC
Eu.LC.2683	Def	/{Initial3 - CRITICAL_FAILURE}		Basic LC
Eu.LC.2686	Def	CRITICAL_FAILURE		Basic LC
Eu.LC.3238	Def	when(d61in_LC_Failure_Status <> "A critical failure is present" AND d62in_LCPF_Failure_Status <> "A critical failure is present" AND d75in_LC_Failure_Status <> "A critical failure is present")/ send Msg_LC_Failure_Status (cOp2_Non_Critical_Failure(),NoCriticalFailureIsPresent)to P2out;{State-internal in CRITICAL_FAILURE}		Basic LC
Eu.LC.3239	Def	when(d61in_LC_Failure_Status = "A critical failure is present" OR d62in_LCPF_Failure_Status = "A critical failure is present" OR d75in_LC_Failure_Status = "A critical failure is present")/send Msg_LC_Failure_Status (cOp2_Non_Critical_Failure(),ACriticalFailureIsPresent)to P2out;{State-internal in CRITICAL_FAILURE}		Basic LC
Eu.LC.2689	Def	REPORT_LEVEL_CROSSING_NON_CRITICAL_FAILURE		Basic LC
Eu.LC.2690	Def	Initial4		Basic LC
Eu.LC.2691	Def	/{Initial4 - NON_CRITICAL_FAILURE}		Basic LC
Eu.LC.2696	Def	NON_CRITICAL_FAILURE		Basic LC
Eu.LC.3240	Def	when(d61in_LC_Failure_Status <> "A non critical failure is present" AND d62in_LCPF_Failure_Status <> "A non critical failure is present" AND d75in_LC_Failure_Status <> "A non critical failure is present")/ send Msg_LC_Failure_Status (NoNonCriticalFailureIsPresent,cOp1_Critical_Failure())to P2out;{State-internal in NON_CRITICAL_FAILURE}		Basic LC
Eu.LC.3241	Def	when(d61in_LC_Failure_Status = "A non critical failure is present" OR d62in_LCPF_Failure_Status = "A non critical failure is present" OR d75in_LC_Failure_Status = "A non critical failure is present")/ send Msg_LC_Failure_Status (ANonCriticalFailureIsPresent,cOp1_Critical_Failure())to P2out;{State-internal in NON_CRITICAL_FAILURE}		Basic LC
Eu.LC.2699	Def	REPORT_LOCAL_HANDOVER		Option LOH
Eu.LC.2700	Def	ALLOW_OR_RETURN_HANDOVER		Option LOH
Eu.LC.3048	Def	when(d81in_Local_Operation_Handover = "Allow handover to local operator")/send Msg_Local_Operation_Handover (AllowHandovertoLocalOperator)to P2out;{State-internal in ALLOW_OR_RETURN_HANDOVER}		Option LOH
Eu.LC.3049	Def	when(d81in_Local_Operation_Handover = "Return handover from local operator")/send Msg_Local_Operation_Handover (ReturnHandoverFromLocalOperator)to P2out;{State-internal in ALLOW_OR_RETURN_HANDOVER}		Option LOH
Eu.LC.2701	Def	Initial6		Option LOH
Eu.LC.2702	Def	/{Initial6 - ALLOW_OR_RETURN_HANDOVER}		Option LOH
Eu.LC.2703	Def	REPORT_OBSTACLE_DETECTION_STATUS		Basic LC
Eu.LC.2704	Def	Initial2		Basic LC
Eu.LC.2705	Def	/{Initial2 - OBSTACLE_IN_CONFLICT_AREA}		Basic LC
Eu.LC.2706	Def	OBSTACLE_IN_CONFLICT_AREA		Basic LC

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.3050	Def	when(d4in_Obstacle_Detection_Status = "No obstacle in the conflict area")/send Msg_Obstacle_Detection_Status (NoObstacleInTheConflictArea)to P2out;{State-internal in OBSTACLE_IN_CONFLICT_AREA}		Basic LC
Eu.LC.3051	Def	when(d4in_Obstacle_Detection_Status = "Obstacle detected in the conflict area")/send Msg_Obstacle_Detection_Status (ObstacleInTheConflictArea)to P2out;{State-internal in OBSTACLE_IN_CONFLICT_AREA}		Basic LC
Eu.LC.2707	Def	REPORT_REQUEST_BY_LOCAL_OPERATOR		Option LOH
Eu.LC.2708	Def	Initial8		Option LOH
Eu.LC.2709	Def	/{Initial8 - REQUEST_TO_ACTIVATE_OR_DEACTIVATE}		Option LOH
Eu.LC.2710	Def	REQUEST_TO_ACTIVATE_OR_DEACTIVATE		Option LOH
Eu.LC.3052	Def	when(d82in_Local_Operation_Handover = "Request to deactivate the level Crossing")/send Msg_Local_Request (LocalRequestToDeactivateTheLevelCrossing)to P2out;{State-internal in REQUEST_TO_ACTIVATE_OR_DEACTIVATE}		Option LOH
Eu.LC.3053	Def	when(d82in_Local_Operation_Handover = "Request to activate the level Crossing")/send Msg_Local_Request (LocalRequestToActivateTheLevelCrossing)to P2out;{State-internal in REQUEST_TO_ACTIVATE_OR_DEACTIVATE}		Option LOH
Eu.LC.2711	Def	REPORT_MONITORING_PARAMETER		Basic LC
Eu.LC.2712	Def	Initial7		Basic LC
Eu.LC.2713	Def	/{Initial7 - REPORT_CLOSURE_TIME_OVERUN}		Basic LC
Eu.LC.2714	Def	REPORT_CLOSURE_TIME_OVERUN		Basic LC
Eu.LC.3054	Def	when(d6in_LC_Monitoring_Status = "Closure time overrun occurred")[d9in_Changed_Monitoring_State = "Monitoring Parameter Set 1"]/ send Msg_LC_Monitoring_Status (ClosureTimeOverrunOccured,MonitoringParameterSet1)to P2out;{State-internal in REPORT_CLOSURE_TIME_OVERUN}		Basic LC
Eu.LC.3055	Def	when(d6in_LC_Monitoring_Status = "No Closure time overrun")[d9in_Changed_Monitoring_State = "Monitoring Parameter Set 1"]/ send Msg_LC_Monitoring_Status (NoClosureTimeOverrun,MonitoringParameterSet1)to P2out;{State-internal in REPORT_CLOSURE_TIME_OVERUN}		Basic LC
Eu.LC.3242	Def	when(d6in_LC_Monitoring_Status = "Closure time overrun occurred")[d9in_Changed_Monitoring_State = "Monitoring Parameter Set 2"]/ send Msg_LC_Monitoring_Status (ClosureTimeOverrunOccured,MonitoringParameterSet2)to P2out;{State-internal in REPORT_CLOSURE_TIME_OVERUN}		Basic LC
Eu.LC.3243	Def	when(d6in_LC_Monitoring_Status = "No Closure time overrun")[d9in_Changed_Monitoring_State = "Monitoring Parameter Set 2"]/ send Msg_LC_Monitoring_Status (NoClosureTimeOverrun,MonitoringParameterSet2)to P2out;{State-internal in REPORT_CLOSURE_TIME_OVERUN}		Basic LC
Eu.LC.3244	Def	when(d9in_Changed_Monitoring_State = "Monitoring Parameter Set 1")[d6in_LC_Monitoring_Status = "Closure time overrun occurred"]/ send Msg_LC_Monitoring_Status (ClosureTimeOverrunOccured,MonitoringParameterSet1)to P2out;{State-internal in REPORT_CLOSURE_TIME_OVERUN}		Basic LC
Eu.LC.3245	Def	when(d9in_Changed_Monitoring_State = "Monitoring Parameter Set 1")[d6in_LC_Monitoring_Status = "No Closure time overrun"]/ send Msg_LC_Monitoring_Status (NoClosureTimeOverrun,MonitoringParameterSet1)to P2out;{State-internal in REPORT_CLOSURE_TIME_OVERUN}		Basic LC
Eu.LC.3246	Def	when(d9in_Changed_Monitoring_State = "Monitoring Parameter Set 2")[d6in_LC_Monitoring_Status = "Closure time overrun occurred"]/ send Msg_LC_Monitoring_Status (ClosureTimeOverrunOccured,MonitoringParameterSet2)to P2out;{State-internal in REPORT_CLOSURE_TIME_OVERUN}		Basic LC
Eu.LC.3247	Def	when(d9in_Changed_Monitoring_State = "Monitoring Parameter Set 2")[d6in_LC_Monitoring_Status = "No Closure time overrun"]/ send Msg_LC_Monitoring_Status (NoClosureTimeOverrun,MonitoringParameterSet2)to P2out;{State-internal in REPORT_CLOSURE_TIME_OVERUN}		Basic LC
Eu.LC.2715	Def	when(d50in_PDI_Connection_State = "READY_FOR_PDI_NO_SCP" OR d50in_PDI_Connection_State = "READY_FOR_PDI" OR d50in_PDI_Connection_State = "SUSPENDED")/{SENDING_LEVEL_CROSSING_REPORTS - INTERFACE_CONNECTION_NOT_ESTABLISHED}		Basic LC
Eu.LC.4174	Def	entry/send Status_Report_Completed to p3inout;{State-internal in SENDING_LEVEL_CROSSING_REPORTS}		Basic LC
Eu.LC.3286	Def	INTERFACE_CONNECTION_NOT_ESTABLISHED		Basic LC
Eu.LC.3288	Def	Start_Status_Report/cOp3_Status_Report();{INTERFACE_CONNECTION_NOT_ESTABLISHED - SENDING_LEVEL_CROSSING_REPORTS}		Basic LC
Eu.LC.2717	Info	S_SCI_LC_Command		Basic LC
Eu.LC.3091	Req	[Block] S_SCI_LC_Command [Functional Viewpoint - Interface Requirements - Functional Entity]  		Basic LC

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.2718	Def	d1in_Cd_Activation	The FlowPort d1in_Cd_Activation belongs to T1 _Cd_Activation.	Basic LC
Eu.LC.2719	Def	d3in_Cd_Local_Operation_Handover	The FlowPort d3in_Cd_Local_Operation_Handover belongs to T3_Cd_Local_Operation_Handover.	Basic LC
Eu.LC.2720	Def	d4in_Cd_Isolate_LC	The FlowPort d4in_Cd_Isolate_LC belongs to T4 _Cd_Isolate_LC.	Basic LC
Eu.LC.2721	Def	d50in_PDI_Connection_State		Basic LC
Eu.LC.2728	Def	t1in_Cd_Activation	The FlowPort T1_Cd_Activation refines the Flow Property Cd_Activation.	Basic LC
Eu.LC.2729	Def	t2in_Cd_Deactivation	The FlowPort T2_Cd_Deactivation refines the Flow Property Cd_Deactivation.	Basic LC
Eu.LC.2730	Def	t3in_Cd_Local_Operation_Handover	The FlowPort T3_Cd_Local_Operation_Handover refines the Flow Property Cd_Local_Operation_Handover.	Option LOH
Eu.LC.2731	Def	t4in_Cd_Isolate_LC	The FlowPort T4_Cd_Isolate_LC refines the Flow Property Cd_Isolate_LC.	Basic LC
Eu.LC.3252	Def	D65in_Con_Use_Pre_Activation	The port D65in_Con_Use_Pre_Activation provides configuration values for Pre-Activation. true: Pre-Activation is used. false: Pre-Activation is not used.	Basic LC
Eu.LC.3253	Def	D67in_Con_Use_Isolation	The port D67in_Con_Use_Isolation provides configuration values for Cd_Isolate_LC. true: Cd_Isolate_LC is used. false: Cd_Isolate_LC is not used.	Basic LC
Eu.LC.2722	Def	P1out	The port P1out exchanges information objects according to SCI_LC_Subsystem_EIL.	Basic LC
Eu.LC.2723	Info	S_SCI_LC_Command - Behaviour		Basic LC
Eu.LC.2724	Req	Functional Viewpoint - Interface Requirements - Functional Entity STD 0 stm [State Machine] S_SCI_LC_Command - Behaviour [Functional Viewpoint - Interface Requirements - Functional Entity STD 0] Initial0 SENDING_COMMANDS when(t1in_Cd_Activation)[d1in_Cd_Activation = "Activation" AND d50in_PDI_Connection_State = "ESTABLISHED"]/ send Cd_Activation (Activation) to P1out; when(t1in_Cd_Activation)[d1in_Cd_Activation = "Pre-Activation" AND d50in_PDI_Connection_State = "ESTABLISHED" AND D65in_Con_Use_Pre_Activation = TRUE]/ send Cd_Activation (Pre-activation) to P1out; when(t4in_Cd_Isolate_LC)[d4in_Cd_Isolate_LC = "Isolate LC enable" AND d50in_PDI_Connection_State = "ESTABLISHED" AND D67in_Con_Use_Isolation = TRUE]/ send Cd_Isolate_LC (EnableIsolationOfTheLevelCrossing) to P1out; when(t2in_Cd_Deactivation)[d50in_PDI_Connection_State = "ESTABLISHED"]/ send Cd_Deactivation (Deactivation) to P1out; when(t3in_Cd_Local_Operation_Handover)[d3in_Cd_Local_Operation_Handover = "Handover to local operator initiated" AND d50in_PDI_Connection_State = "ESTABLISHED"]/ send Cd_Local_Operation_Handover (HandoverToLocalOperatorInitiated) to P1out; when(t3in_Cd_Local_Operation_Handover)[d3in_Cd_Local_Operation_Handover = "Handover to local operator established" AND d50in_PDI_Connection_State = "ESTABLISHED"]/ send Cd_Local_Operation_Handover (HandoverToLocalOperatorEstablished) to P1out; when(t3in_Cd_Local_Operation_Handover)[d3in_Cd_Local_Operation_Handover = "No handover to local operator" AND d50in_PDI_Connection_State = "ESTABLISHED"]/ send Cd_Local_Operation_Handover (NoHandOverToLocalOperator) to P1out; when(t4in_Cd_Isolate_LC)[d4in_Cd_Isolate_LC = "Isolate LC disable" AND d50in_PDI_Connection_State = "ESTABLISHED" AND D67in_Con_Use_Isolation = TRUE]/ send Cd_Isolate_LC (DisableIsolationOfTheLevelCrossing) to P1out;	This state machine diagram describes the requirements for the following functionalities: - receives the activation state to be set from internal logic and commands this to the Subsystem - Level Crossing - receives the local operation handover to be set from internal logic and commands this to the Subsystem - Level Crossing	Basic LC Option LOH
Eu.LC.2725	Def	Initial0		Basic LC
Eu.LC.2726	Def	/{Initial0 - SENDING_COMMANDS}		Basic LC
Eu.LC.2727	Def	SENDING_COMMANDS		Basic LC
Eu.LC.3056	Def	when(t1in_Cd_Activation)[d1in_Cd_Activation = "Pre-Activation" AND d50in_PDI_Connection_State = "ESTABLISHED" AND D65in_Con_Use_Pre_Activation = TRUE]/ send Cd_Activation (Pre-activation)to P1out;{State-internal in SENDING_COMMANDS}		Basic LC
Eu.LC.3057	Def	when(t1in_Cd_Activation)[d1in_Cd_Activation = "Activation" AND d50in_PDI_Connection_State = "ESTABLISHED"]/ send Cd_Activation (Activation)to P1out;{State-internal in SENDING_COMMANDS}		Basic LC
Eu.LC.3058	Def	when(t2in_Cd_Deactivation)[d50in_PDI_Connection_State = "ESTABLISHED"]/ send Cd_Deactivation (Deactivation)to P1out;{State-internal in SENDING_COMMANDS}		Basic LC

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.3059	Def	when(t3in_Cd_Local_Operation_Handover)[d3in_Cd_Local_Operation_Handover = "Handover to local operator initiated" AND d50in_PDI_Connection_State = "ESTABLISHED"]/ send Cd_Local_Operation_Handover (HandoverToLocalOperatorInitiated)to P1out;{State-internal in SENDING_COMMANDS}		Option LOH
Eu.LC.3060	Def	when(t3in_Cd_Local_Operation_Handover)[d3in_Cd_Local_Operation_Handover = "Handover to local operator established" AND d50in_PDI_Connection_State = "ESTABLISHED"]/ send Cd_Local_Operation_Handover (HandoverToLocalOperatorEstablished)to P1out;{State-internal in SENDING_COMMANDS}		Option LOH
Eu.LC.3061	Def	when(t3in_Cd_Local_Operation_Handover)[d3in_Cd_Local_Operation_Handover = "No handover to local operator" AND d50in_PDI_Connection_State = "ESTABLISHED"]/ send Cd_Local_Operation_Handover (NoHandOverToLocalOperator)to P1out;{State-internal in SENDING_COMMANDS}		Option LOH
Eu.LC.3062	Def	when(t4in_Cd_Isolate_LC)[d4in_Cd_Isolate_LC = "Isolate LC enable" AND d50in_PDI_Connection_State = "ESTABLISHED" AND D67in_Con_Use_Isolation = TRUE]/ send Cd_Isolate_LC (EnableIsolationOfTheLevelCrossing)to P1out;{State-internal in SENDING_COMMANDS}		Basic LC
Eu.LC.3254	Def	when(t4in_Cd_Isolate_LC)[d4in_Cd_Isolate_LC = "Isolate LC disable" AND d50in_PDI_Connection_State = "ESTABLISHED" AND D67in_Con_Use_Isolation = TRUE]/ send Cd_Isolate_LC (DisableIsolationOfTheLevelCrossing)to P1out;{State-internal in SENDING_COMMANDS}		Basic LC
Eu.LC.2732	Info	S_SCI_LC_Receive		Basic LC
Eu.LC.3092	Req	[Block] S_SCI_LC_Receive [Functional Viewpoint - Interface Requirements - Functional Entity]  		Basic LC

Eu.LC.3255	Def	/* cOp2_Get_Monitoring_Status */
		
		if ParameterChangedMonitoringParameters = ChangendMonitoringStatus.MonitoringParameterSet1 then
		d62out_Msg_LC_Monitoring_Status_Changed_Monitoring_Parameter := "Monitoring Parameter Set 1";
		elseif ParameterChangedMonitoringParameters = ChangendMonitoringStatus.MonitoringParameterSet2 then
		d62out_Msg_LC_Monitoring_Status_Changed_Monitoring_Parameter := "Monitoring Parameter Set 2";
		end if
		if ParameterTimeOverrun = TimeOverrunState.ClosureTimeOverrunOccured then
		d61out_Msg_LC_Monitoring_Status_Closure_time := "Closure time overrun occurred";
		elseif ParameterTimeOverrun = TimeOverrunState.NoClosureTimeOverrun then
		d61out_Msg_LC_Monitoring_Status_Closure_time := "No Closure time overrun";
		end if
			cOp2_Get_Monitoring_Status	Basic LC

Eu.LC.3257	Def	/* cOp3_Get_Failure_State */
		
		if ParameterCriticalFailure = CriticalFailureStatusState.ACriticalFailureIsPresent then
		d72out_Msg_LC_Failure_Status_Critical := "A critical failure is present";
		elseif ParameterCriticalFailure = CriticalFailureStatusState.NoCriticalFailureIsPresent then
		d72out_Msg_LC_Failure_Status_Critical := "No critical failure is present";
		end if
		if ParameterNonCriticalFailure = NonCriticalFailureStatusState.ANonCriticalFailureIsPresent then
		d71out_Msg_LC_Failure_Status_Non_Critical := "A non critical failure is present";
		elseif ParameterNonCriticalFailure = NonCriticalFailureStatusState.NoNonCriticalFailureIsPresent then
		d71out_Msg_LC_Failure_Status_Non_Critical := "No non critical failure is present";
		end if
			cOp3_Get_Failure_State	Basic LC

Eu.LC.2733	Def	d18out_Msg_Detection_Element_Status	The FlowPort d18out_Msg_Detection_Element_Status belongs to T18_Msg_Detection_Element_Status.	Basic LC
Eu.LC.2734	Def	d50in_PDI_Connection_State		Basic LC

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.2735	Def	d5out_Msg_LC_Functional_Status	The FlowPort d5out_Msg_LC_Functional_Status belongs to T5_Msg_LC_Functional_Status.	Basic LC
Eu.LC.2736	Def	d61out_Msg_LC_Monitoring_Status_Closure_time	The FlowPort d61out_Msg_LC_Monitoring_Status_Closure_time belongs to T6_Msg_LC_Monitoring_Status.	Basic LC
Eu.LC.2737	Def	d72out_Msg_LC_Failure_Status_Critical	The FlowPort d72out_Msg_LC_Failure_Status_Critical belongs to T7_Msg_LC_Failure_Status.	Basic LC
Eu.LC.2738	Def	d8out_Msg_Local_Request	The FlowPort d8out_Msg_Local_Request belongs to T8_Msg_Local_Request.	Option LOH
Eu.LC.2739	Def	d91out_Msg_Obstacle_Detection_Status	The FlowPort d91out_Msg_Obstacle_Detection_Status belongs to T91_Msg_Obstacle_Detection_Status.	Basic LC
Eu.LC.2740	Def	d9out_Msg_Local_Operation_Handover	The FlowPort d9out_Msg_Local_Operation_Handover belongs to T9_Msg_Local_Operation_Handover.	Option LOH
Eu.LC.3259	Def	d62out_Msg_LC_Monitoring_Status_Changed_Monitoring_Parameter	The FlowPort d62out_Msg_LC_Monitoring_Status_Changed_Monitoring_Parameter belongs to T6 _Msg_LC_Monitoring_Status.	Basic LC
Eu.LC.3260	Def	d71out_Msg_LC_Failure_Status_Non_Critical	The FlowPort d71out_Msg_LC_Failure_Status_Non_Critical belongs to T7_Msg_LC_Failure_Status.	Basic LC
Eu.LC.2747	Def	t18out_Msg_Detection_Element_Status	The FlowPort t18out_Msg_Detection_Element_Status refines the Flow Properties Vacated_Detection_Element, Occupied_Detection_Element or Failed_Detection_Element.	Basic LC
Eu.LC.2748	Def	t5out_Msg_LC_Functional_Status	The FlowPort t5out_Msg_LC_Functional_Status refines the Flow Property Msg_LC_Functional_Status.	Basic LC
Eu.LC.2749	Def	t6out_Msg_LC_Monitoring_Status	The FlowPort t6out_Msg_LC_Monitoring_Status refines the Flow Property Msg_LC_Monitoring_Status.	Basic LC
Eu.LC.2750	Def	t7out_Msg_LC_Failure_Status	The FlowPort t7out_Msg_LC_Failure_Status refines the Flow Property Msg_LC_Failure_Status.	Basic LC
Eu.LC.2751	Def	t8out_Msg_Local_Request	The FlowPort T8_Msg_Local_Request refines the Flow Properties Activate and Deactivate.	Option LOH
Eu.LC.2752	Def	t91out_Msg_Obstacle_Detection_Status	The FlowPort t91out_Msg_Obstacle_Detection_Status refines the Flow Properties Status_Level_Crossing_Protection_Facility.	Basic LC
Eu.LC.2753	Def	t9out_Msg_Local_Operation_Handover	The FlowPort t9out_Msg_Local_Operation_Handover refines the Flow Property Msg_Local_Operation_Handover.	Option LOH
Eu.LC.2741	Def	P2in	The port P2in exchanges information objects according to SCI_LC_Subsystem_EIL.	Basic LC
Eu.LC.2742	Info	S_SCI_LC_Receive - Behaviour		Basic LC

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.

Eu.LC.2743	Req	Functional Viewpoint - Interface Requirements - Functional Entity STD 1
		stm [State Machine] S_SCI_LC_Receive - Behaviour [Functional Viewpoint - Interface Requirements - Functional Entity STD 1]
		Initial0
		RECEIVING_LC_REPORTS
		Msg_LC_Functional_Status[ReportedActivationlStatusState = DeactivatingAndUnprotected]/d5out_Msg_LC_Functional_Status := "Deactivating and unprotected";
		t5out_Msg_LC_Functional_Status := TRUE;
		Msg_LC_Functional_Status[ReportedActivationlStatusState = ActivatedAndUnprotected]/d5out_Msg_LC_Functional_Status := "Activated and unprotected";
		t5out_Msg_LC_Functional_Status := TRUE;
		Msg_LC_Functional_Status[ReportedActivationlStatusState = ActivatedAndProtected]/d5out_Msg_LC_Functional_Status := "Activated and protected";
		t5out_Msg_LC_Functional_Status := TRUE;
		Msg_LC_Functional_Status[ReportedActivationlStatusState = PreActivated]/d5out_Msg_LC_Functional_Status := "Pre-Activated";
		t5out_Msg_LC_Functional_Status := TRUE;
		Msg_LC_Functional_Status[ReportedActivationlStatusState = IsolatedLC]/d5out_Msg_LC_Functional_Status := "Isolated LC";
		t5out_Msg_LC_Functional_Status := TRUE;
		Msg_Obstacle_Detection[ReportedObstacleDetectionStatusState = ObstacleInTheConflictArea]/d91out_Msg_Obstacle_Detection_Status := "Obstacle detected in the conflict area";
		t91out_Msg_Obstacle_Detection_Status := TRUE;
		Msg_Obstacle_Detection[ReportedObstacleDetectionStatusState = NoObstacleInTheConflictArea]/d91out_Msg_Obstacle_Detection_Status := "No obstacle in the conflict area";
		t91out_Msg_Obstacle_Detection_Status := TRUE;
		Msg_LC_Failure_Status/cOp3_Get_Failure_State (ReportedNonCriticalFailureState, ReportedCriticalFailureState);
		t7out_Msg_LC_Failure_Status := TRUE;
		Msg_Detection_Element_Status[ReportedDetectionElementStatusState = DetectionElementIsVacant]/d18out_Msg_Detection_Element_Status := "Vacant";
		t18out_Msg_Detection_Element_Status := TRUE;
		Msg_Detection_Element_Status[ReportedDetectionElementStatusState = DetectionElementIsOccupied]/d18out_Msg_Detection_Element_Status := "Occupied";
		t18out_Msg_Detection_Element_Status := TRUE;
		Msg_Detection_Element_Status[ReportedDetectionElementStatusState = DetectionElementIsFailed]/d18out_Msg_Detection_Element_Status := "Failed";
		t18out_Msg_Detection_Element_Status := TRUE;
		Msg_Local_Operation_Handover[ReportedLocalOperationHandoverState = AllowHandovertoLocalOperator]/d9out_Msg_Local_Operation_Handover := "Allow handover to local operator";
		t9out_Msg_Local_Operation_Handover := TRUE;
		Msg_Local_Operation_Handover[ReportedLocalOperationHandoverState = ReturnHandoverFromLocalOperator]/d9out_Msg_Local_Operation_Handover := "Return handover from local operator";
		t9out_Msg_Local_Operation_Handover := TRUE;
		Msg_LC_Monitoring_Status/cOp2_Get_Monitoring_Status (ReportedChangedMonitoringParameter,ReportedTimeOverrunState);
		t6out_Msg_LC_Monitoring_Status := TRUE;
		Msg_Local_Request[ReportedLocalRequestState = LocalRequestToActivateTheLevelCrossing]/d8out_Msg_Local_Request := "Local request to activate the level crossing";
		t8out_Msg_Local_Request := TRUE;
		Msg_Local_Request[ReportedLocalRequestState = LocalRequestToDeactivateTheLevelCrossing]/d8out_Msg_Local_Request := "Local request to deactivate the level crossing";
		t8out_Msg_Local_Request := TRUE;
		Msg_LC_Functional_Status[ReportedActivationlStatusState = DeactivatedAndIdle]/d5out_Msg_LC_Functional_Status := "Deactivated and idle";
		t5out_Msg_LC_Functional_Status := TRUE;
			This state machine diagram describes the requirements for the following functionalities:
			
			- receives the reported functional status and forwards it to the internal logic
			- receives the reported obstacle detection status and forwards it to the internal logic
			- receives the reported level crossing critical failure and forwards it to the internal logic
			- receives the reported level crossing non critical failure and forwards it to the internal logic
			- receives the reported detection element status and forwards it to the internal logic
			- receives the reported local handover and forwards it to the internal logic
			- receives the reported monitoring parameter and forwards it to the internal logic
			- receives the reported request by local operator and forwards it to the internal logic
				Basic LC Option LOH

Eu.LC.2744	Def	Initial0		Basic LC
Eu.LC.2745	Def	/{Initial0 - RECEIVING_LC_REPORTS}		Basic LC
Eu.LC.2746	Def	RECEIVING_LC_REPORTS		Basic LC
Eu.LC.3063	Def	Msg_Detection_Element_Status[ReportedDetectionElementStatusState = DetectionElementIsOccupied]/d18out_Msg_Detection_Element_Status := "Occupied"; t18out_Msg_Detection_Element_Status := TRUE;{State-internal in RECEIVING_LC_REPORTS}		Basic LC
Eu.LC.3064	Def	Msg_LC_Functional_Status[ReportedActivationlStatusState = DeactivatingAndUnprotected]/d5out_Msg_LC_Functional_Status := "Deactivating and unprotected"; t5out_Msg_LC_Functional_Status := TRUE;{State-internal in RECEIVING_LC_REPORTS}		Basic LC
Eu.LC.3065	Def	Msg_LC_Functional_Status[ReportedActivationlStatusState = PreActivated]/d5out_Msg_LC_Functional_Status := "Pre-Activated"; t5out_Msg_LC_Functional_Status := TRUE;{State-internal in RECEIVING_LC_REPORTS}		Basic LC
Eu.LC.3066	Def	Msg_LC_Functional_Status[ReportedActivationlStatusState = ActivatedAndUnprotected]/d5out_Msg_LC_Functional_Status := "Activated and unprotected"; t5out_Msg_LC_Functional_Status := TRUE;{State-internal in RECEIVING_LC_REPORTS}		Basic LC
Eu.LC.3067	Def	Msg_LC_Monitoring_Status/cOp2_Get_Monitoring_Status (ReportedChangedMonitoringParameter,ReportedTimeOverrunState); t6out_Msg_LC_Monitoring_Status := TRUE;{State-internal in RECEIVING_LC_REPORTS}		Basic LC
Eu.LC.3070	Def	Msg_Local_Operation_Handover[ReportedLocalOperationHandoverState = AllowHandovertoLocalOperator]/d9out_Msg_Local_Operation_Handover := "Allow handover to local operator"; t9out_Msg_Local_Operation_Handover := TRUE;{State-internal in RECEIVING_LC_REPORTS}		Option LOH
Eu.LC.3071	Def	Msg_Local_Operation_Handover[ReportedLocalOperationHandoverState = ReturnHandoverFromLocalOperator]/d9out_Msg_Local_Operation_Handover := "Return handover from local operator"; t9out_Msg_Local_Operation_Handover := TRUE;{State-internal in RECEIVING_LC_REPORTS}		Option LOH

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.3072	Def	Msg_Local_Request[ReportedLocalRequestState = LocalRequestToActivateTheLevelCrossing]/d8out_Msg_Local_Request := "Local request to activate the level crossing"; t8out_Msg_Local_Request := TRUE;{State-internal in RECEIVING_LC_REPORTS}		Option LOH
Eu.LC.3073	Def	Msg_Local_Request[ReportedLocalRequestState = LocalRequestToDeactivateTheLevelCrossing]/d8out_Msg_Local_Request := "Local request to deactivate the level crossing"; t8out_Msg_Local_Request := TRUE;{State-internal in RECEIVING_LC_REPORTS}		Option LOH
Eu.LC.3074	Def	Msg_Detection_Element_Status[ReportedDetectionElementStatusState = DetectionElementIsVacant]/d18out_Msg_Detection_Element_Status := "Vacant"; t18out_Msg_Detection_Element_Status := TRUE;{State-internal in RECEIVING_LC_REPORTS}		Basic LC
Eu.LC.3075	Def	Msg_Obstacle_Detection[ReportedObstacleDetectionStatusState = NoObstacleInTheConflictArea]/d91out_Msg_Obstacle_Detection_Status := "No obstacle in the conflict area"; t91out_Msg_Obstacle_Detection_Status := TRUE;{State-internal in RECEIVING_LC_REPORTS}		Basic LC
Eu.LC.3076	Def	Msg_Obstacle_Detection[ReportedObstacleDetectionStatusState = ObstacleInTheConflictArea]/d91out_Msg_Obstacle_Detection_Status := "Obstacle detected in the conflict area"; t91out_Msg_Obstacle_Detection_Status := TRUE;{State-internal in RECEIVING_LC_REPORTS}		Basic LC
Eu.LC.3077	Def	Msg_Detection_Element_Status[ReportedDetectionElementStatusState = DetectionElementIsFailed]/d18out_Msg_Detection_Element_Status := "Failed"; t18out_Msg_Detection_Element_Status := TRUE;{State-internal in RECEIVING_LC_REPORTS}		Basic LC
Eu.LC.3078	Def	Msg_LC_Failure_Status/cOp3_Get_Failure_State (ReportedNonCriticalFailureState, ReportedCriticalFailureState); t7out_Msg_LC_Failure_Status := TRUE;{State-internal in RECEIVING_LC_REPORTS}		Basic LC
Eu.LC.3082	Def	Msg_LC_Functional_Status[ReportedActivationlStatusState = ActivatedAndProtected]/d5out_Msg_LC_Functional_Status := "Activated and protected"; t5out_Msg_LC_Functional_Status := TRUE;{State-internal in RECEIVING_LC_REPORTS}		Basic LC
Eu.LC.3083	Def	Msg_LC_Functional_Status[ReportedActivationlStatusState = IsolatedLC]/d5out_Msg_LC_Functional_Status := "Isolated LC"; t5out_Msg_LC_Functional_Status := TRUE;{State-internal in RECEIVING_LC_REPORTS}		Basic LC
Eu.LC.4107	Def	Msg_LC_Functional_Status[ReportedActivationlStatusState = DeactivatedAndIdle]/d5out_Msg_LC_Functional_Status := "Deactivated and idle"; t5out_Msg_LC_Functional_Status := TRUE;{State-internal in RECEIVING_LC_REPORTS}		Basic LC
Eu.LC.117	Head	**3.4.2 SMI-LC (Subsystem - Maintenance and Data Management)**		
Eu.LC.118	Info	The generic InformationFlows and the related FlowProperties through the SMI-LC are specified in [Eu.Doc.120].		Basic LC
Eu.LC.119	Head	**3.4.3 SDI-LC (Subsystem - Maintenance and Data Management)**		
Eu.LC.120	Info	The generic data points through the SDI-LC are specified in [Eu.Doc.94]. The specific data points through the SDI-LC are specified in [Eu.Doc.78].		Basic LC
Eu.LC.3292	Head	**3.4.4 SSI-LC (Subsystem - Security Services Platform)**		
Eu.LC.3293	Info	The generic content through SSI-LC is specified in [Eu.Doc.117]. Note: In future phases, the EULYNX security specifications will be replaced by harmonised specifications published by the EU-Rail System Pillar Cyber Security domain.		Basic LC
Eu.LC.136	Head	**3.4.5 LC1 (Basic Data identifier)**		
Eu.LC.137	Info	The generic FlowSpecification and the related FlowProperties through LC1 are specified in [Eu.Doc.20].		Basic LC
Eu.LC.138	Head	**3.4.6 LC2 (Maintainer)**		
Eu.LC.139	Info	The generic FlowProperties through LC2 are specified in [Eu.Doc.20].		Basic LC
Eu.LC.3223	Info	Maintainer		Basic LC
Eu.LC.148	Head	**3.4.7 LC5 (Detection element)**		
Eu.LC.149	Info	Detection_element	Definition of the InformationFlow (by FlowSpecification) for Control Interface LC5 (Detection element).	Basic LC
Eu.LC.150	Def	Occupied_Detection_Element	The Subsystem - Level Crossing detects that the Detection element is occupied.	Basic LC
Eu.LC.151	Def	Vacated_Detection_Element	The Subsystem - Level Crossing detects that the Detection element is vacant.	Basic LC
Eu.LC.152	Def	Failed_Detection_Element	The Subsystem - Level Crossing detects that the Detection element is failed.	Basic LC
Eu.LC.153	Head	**3.4.8 LC6 (Local operator)**		
Eu.LC.154	Info	Local_operator	Definition of the InformationFlow (by FlowSpecification) for Control and Display Interface LC6 (Local operator).	Option LOH
Eu.LC.155	Def	Activate	The Subsystem - Level Crossing detects the local activation of the Level Crossing protection facility from the Local operator.	Option LOH
Eu.LC.156	Def	Deactivate	The Subsystem - Level Crossing detects the local deactivation of the Level Crossing protection facility from the Local operator.	Option LOH
Eu.LC.159	Def	Input_Allow_Handover_To_Local_Operator	The Subsystem - Level Crossing detects that the Local operator confirms a handover of the local operations.	Option LOH
Eu.LC.160	Def	Input_Return_Handover_To_Local_Operator	The Subsystem - Level Crossing detects that the Local operator requests to return the handover of the local operations.	Option LOH

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.161	Def	Output_Established_Handover_To_Local_Operator	The Subsystem - Level Crossing reports to the Local operator that the handover of the local operations is established.	Option LOH
Eu.LC.162	Def	Output_No_Handover_To_Local_Operator	The Subsystem - Level Crossing reports to the Local operator that there is no handover of the local operations is initiated.	Option LOH
Eu.LC.163	Def	Output_Initiated_Handover_To_Local_Operator	The Subsystem - Level Crossing reports to the Local operator that the handover of the local operations is initiated.	Option LOH
Eu.LC.164	Head	**3.4.9 LC4 (Level Crossing protection facility)**		
Eu.LC.165	Info	Level_Crossing_protection_facility	Definition of the InformationFlow (by FlowSpecification) for Control and Display Interface LC4 (Level Crossing protection facility).	Basic LC
Eu.LC.166	Def	Activate	The Subsystem - Level Crossing request the Level Crossing protection facility to activate the Level Crossing protection facility.	Basic LC
Eu.LC.167	Def	Deactivate	The Subsystem - Level Crossing request the Level Crossing protection facility to deactivate the Level Crossing protection facility.	Basic LC
Eu.LC.168	Def	National_Specific_State	The Subsystem - Level Crossing request the Level Crossing protection facility to change to a national specific state.	Basic LC
Eu.LC.169	Def	Pre-Activate	The Subsystem - Level Crossing request the Level Crossing protection facility to pre-activate the Level Crossing protection facility.	Basic LC
Eu.LC.170	Def	Status_Level_Crossing_Protection_Facility	The Level Crossing protection facility reports its new status to the Subsystem - Level Crossing.	Basic LC
Eu.LC.1254	Head	**4 RAMSS requirements**		
Eu.LC.1255	Info	The requirements for reliability, availability, maintainability, safety and security are specified in [Eu.Doc.20].		Basic LC
Eu.LC.1256	Head	**5 Technical requirements**		
Eu.LC.1257	Info	The generic technical requirements are specified in [Eu.Doc.20].		Basic LC
Eu.LC.1258	Head	**5.1 Specific technical interface requirements**		
Eu.LC.1259	Head	**5.1.1 Interface to the Point of Service - Signalling (PoS - Signalling)**		
Eu.LC.1260	Req	Via the technical interface **PoS-Signalling** the data of the functional interface "SCI-LC" shall be exchanged with the Subsystem - Electronic Interlocking as specified in [Eu.Doc.92].		Basic LC
Eu.LC.1261	Req	Via the technical interface **PoS-Signalling** the data of the functional interface "SMI-LC" shall be exchanged with the Subsystem - Maintenance and Data Management as specified in [Eu.Doc.76].		Basic LC
Eu.LC.1262	Req	Via the technical interface **PoS-Signalling** the data of the functional interface "SDI-LC" shall be exchanged with the Subsystem - Maintenance and Data Management as specified in [Eu.Doc.77].		Basic LC
Eu.LC.3265	Req	Via the technical interface **PoS-Signalling** the data of the functional interface "SSI-LC" shall be exchanged with the Subsystem - Security Services Platform as specified in [Eu.Doc.117]. Note: In future phases, the EULYNX security specifications will be replaced by harmonised specifications published by the EU-Rail System Pillar Cyber Security domain.		Basic LC
Eu.LC.1265	Head	**5.1.2 Interface to the Detection element**		
Eu.LC.1266	Info	These requirements shall be defined by national specification. Note: In future phases of the System Pillar, national specifications will be replaced by harmonised specifications.		Basic LC
Eu.LC.1267	Head	**5.1.3 Interface to the Local operator**		
Eu.LC.1268	Info	These requirements shall be defined by national specification. Note: In future phases of the System Pillar, national specifications will be replaced by harmonised specifications.		Basic LC
Eu.LC.1311	Head	**5.1.4 Interface to the Level Crossing protection facility**		Basic LC
Eu.LC.1312	Info	These requirements shall be defined by national specification. Note: In future phases of the System Pillar, national specifications will be replaced by harmonised specifications.		Basic LC
Eu.LC.1313	Info	The Status_Level_Crossing_Protection_Facility message via LC4 includes the following information: • LCPF_Monitoring_Status_Barrier_Position • LCPF_Monitoring_Status_Barrier_Movement • LCPF_Monitoring_Status_Barrier_Intact • LCFP_Monitoring_Status_Road_Lights • LCFP_Monitoring_Status_Hardware • LCFP_Monitoring_Status_Power • LCPF_Failure_Status • LCPF_Functional_Status_Idle • LCPF_Functional_Status_Unprotected • LCPF_Functional_Status_Protected • Obstacle_Detection_Status		Basic LC
Eu.LC.1314	Info	The LC4 interface is defined as a functional interface, physical properties are not currently defined. This specification is based upon the following assumptions on the properties of the LC4 interface.		Basic LC

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.1317	Info	General assumptions:		Basic LC
Eu.LC.1315	Info	• Obstacle detectors are connected to the LCPF. The obstacle detection status is reported to the Subsystem – Level Crossing via LC4.		Basic LC
Eu.LC.1318	Info	• The LCPF may be operated independent of LC4 interface according to national specifications. For example, this can be a local switch on the LCPF to directly operate road signals and barriers. This can be used even when the subsystem LC is not operational or has no connection to the electronic interlocking. Note: In future phases of the System Pillar, national specifications will be replaced by harmonised specifications.		Basic LC
Eu.LC.1319	Info	• In case the LCPF is operated independent of LC4, national operational rules must be in place to avoid conflicts with activation requests from the interlocking.		Basic LC
Eu.LC.1320	Info	• The LCPF always reports its functional and monitoring status on LC4, regardless whether it is operated via LC4 or according to national specifications. Note: In future phases of the System Pillar, national specifications will be replaced by harmonised specifications.		Basic LC
Eu.LC.1269	Head	**5.2 Time behaviour**		
Eu.LC.1270	Req	The time values defined in the chapter Functional requirements specification (Eu.LC.172) shall be configured for the operation of the Subsystem - Level Crossing.		Basic LC
Eu.LC.3919	Info	**5.2.1 Response times**		
Eu.LC.3920	Req	The Subsystem - Level Crossing shall send the corresponding message telegram to the Subsystem - Electronic Interlocking within 700 ms after the detection of a passing wheel.		Basic LC
Eu.LC.3923	Req	The Subsystem - Level Crossing shall activate the Level Crossing Protection Facility within 600 ms after receipt of a command telegram.		Basic LC
Eu.LC.3922	Req	The Subsystem - Level Crossing shall deactivate the Level Crossing Protection Facility within 600 ms after receipt of a command telegram.		Basic LC
Eu.LC.3921	Req	The Subsystem - Level Crossing shall send the corresponding message telegram to the Subsystem - Electronic Interlocking within 600 ms after detecting a Local Operator Input.		Option LOH
Eu.LC.3924	Req	The Subsystem - Level Crossing shall set the Local Operator Output within 500 ms after receipt of a command telegram.		Option LOH
Eu.LC.4142	Req	The Subsystem - Level Crossing shall send the corresponding message telegram to the Subsystem - Electronic Interlocking within 600 ms after detecting at the Level Crossing Protection Facility a change of the functional status to unprotected caused by a barrier failure.		Basic LC
Eu.LC.4145	Req	The Subsystem - Level Crossing shall send the corresponding message telegram to the Subsystem - Electronic Interlocking within 800 ms after detecting at the Level Crossing Protection Facility a change of the functional status to unprotected caused by a road light failure.		Basic LC
Eu.LC.4144	Req	The Subsystem - Level Crossing shall send the corresponding message telegram to the Subsystem - Electronic Interlocking within 700 ms after detecting at the Level Crossing Protection Facility a change of the functional status to protected.		Basic LC
Eu.LC.4143	Req	The Subsystem - Level Crossing shall send the corresponding message telegram to the Subsystem - Electronic Interlocking within 600 ms after detecting at the Level Crossing Protection Facility a change of the obstacle detection status.		Basic LC
Eu.LC.4147	Req	The Subsystem - Level Crossing shall send the corresponding message telegram to the Subsystem - Electronic Interlocking within 1000 ms after detecting at the Level Crossing Protection Facility a change of the monitoring status (incl. barrier position, barrier movement, barrier intact, road lights, hardware, power).		Basic LC
Eu.LC.4146	Req	The Subsystem - Level Crossing shall send the corresponding message telegram to the Subsystem - Electronic Interlocking within 1000 ms after detecting at the Level Crossing Protection Facility a change of the failure status.		Basic LC
Eu.LC.1271	Head	**5.3 Configuration and engineering data**		
Eu.LC.1272	Head	**5.3.1 Specific data**		
Eu.LC.1273	Req	The specific configuration and engineering data for the Subsystem - Level Crossing shall include as a minimum the following information:		Basic LC
Eu.LC.1275	Req	• The applicable time values defined in chapter Definition of time values (Eu.LC.172).		Basic LC
Eu.LC.1321	Req	• The usage of the Closure Time.		Basic LC
Eu.LC.1323	Req	• The usage of the PDI Loss Deactivation Time.		Basic LC
Eu.LC.1280	Req	• The usage of the activation type Pre-Activation.		Basic LC
Eu.LC.1322	Req	• The usage of Detection elements.		Basic LC
Eu.LC.1278	Req	• The number of Detection elements.		Basic LC
Eu.LC.1340	Req	• The index of Detection elements.		Basic LC
Eu.LC.1325	Req	• The usage of obstacle detection.		Basic LC
Eu.LC.1326	Req	• The usage of LC isolation function.		Basic LC
Eu.LC.1324	Req	• List of triggers resulting in a critical failure.		Basic LC
Eu.LC.1341	Req	• List of triggers resulting in a non-critical failure.		Basic LC
Eu.LC.1336	Req	• The presence of local operation handover.		Option LOH
Eu.LC.1337	Req	• The index of local operations handovers.		Option LOH
Eu.LC.1338	Req	• The presence of local (de)activation requests.		Option LOH
Eu.LC.1339	Req	• The index of local (de)activation requests.		Option LOH
Eu.LC.1288	Req	Two different data sections can be loaded which are the safety-relevant data and the non safety-relevant data. The following definitions apply to the assignment of the sections:		Basic LC
Eu.LC.1290	Req	• The configuration data, such as the IP addresses of the Subsystem - Electronic Interlocking, the value of the diagnostic data points with attribute type 'configuration', is non safety-relevant. This data shall be used to calculate the CSNS.		Basic LC
Eu.LC.1291	Req	• The remaining configuration data is currently categorised as safety-relevant. This data shall be used to calculate the CSS.		Basic LC

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.1292	Req	• The engineering data is safety-relevant. This data shall be used to calculate the CSS.		Basic LC
Eu.LC.3947	Head	**5.3.2 Value configuration**		
Eu.LC.3948	Req	**Con_tmax_Closure_Time** The time value shall be configured in accordance with: Configurable resolution: steps of 1 s Configurable range: from 0 s up to 1800 s Con_tmax_Closure_Time is defined in Eu.LC.175.		Basic LC
Eu.LC.3949	Req	**Con_t_PDI_Loss_Deactivation_Delay** The time value shall be configured in accordance with: Configurable resolution: steps of 1 s Configurable range: from 0 s up to 1800 s Con_t_PDI_Loss_Deactivation_Delay is defined in Eu.LC.177.		Basic LC