



EULYNX Initiative



Europe's Rail Joint Undertaking

Requirements specification for subsystem Level Crossing

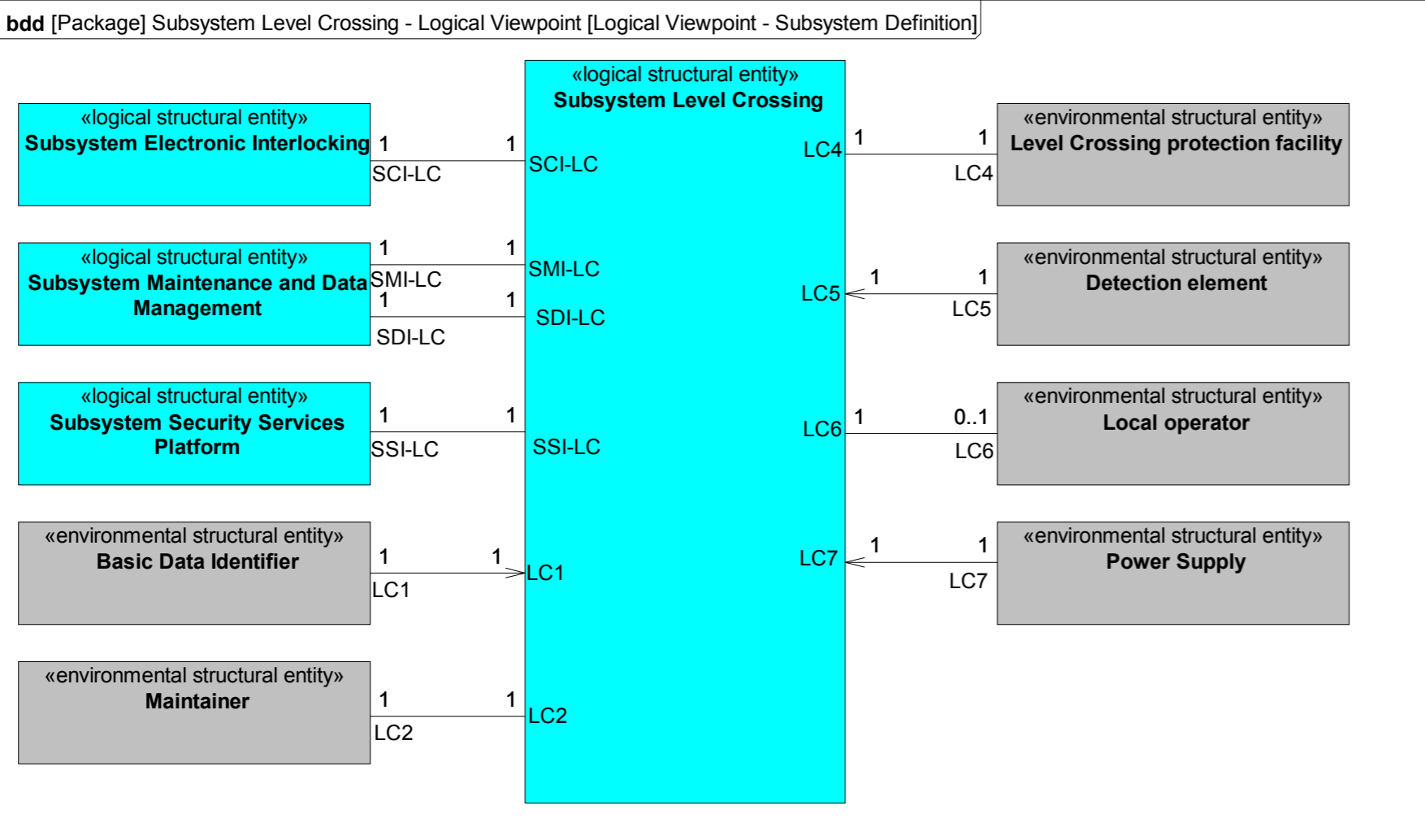
Document number: Eu.Doc.108
Version: 2.2 (0.A)

Contents

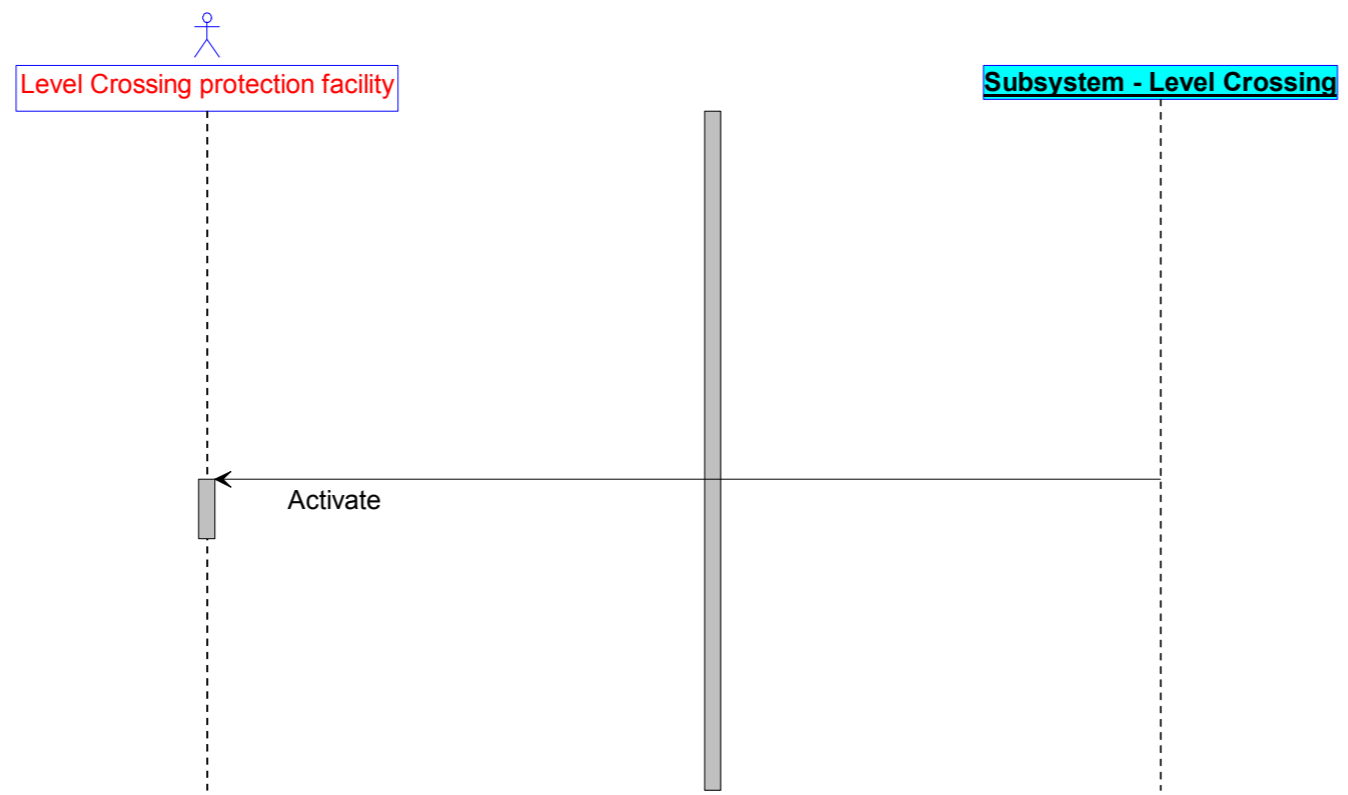
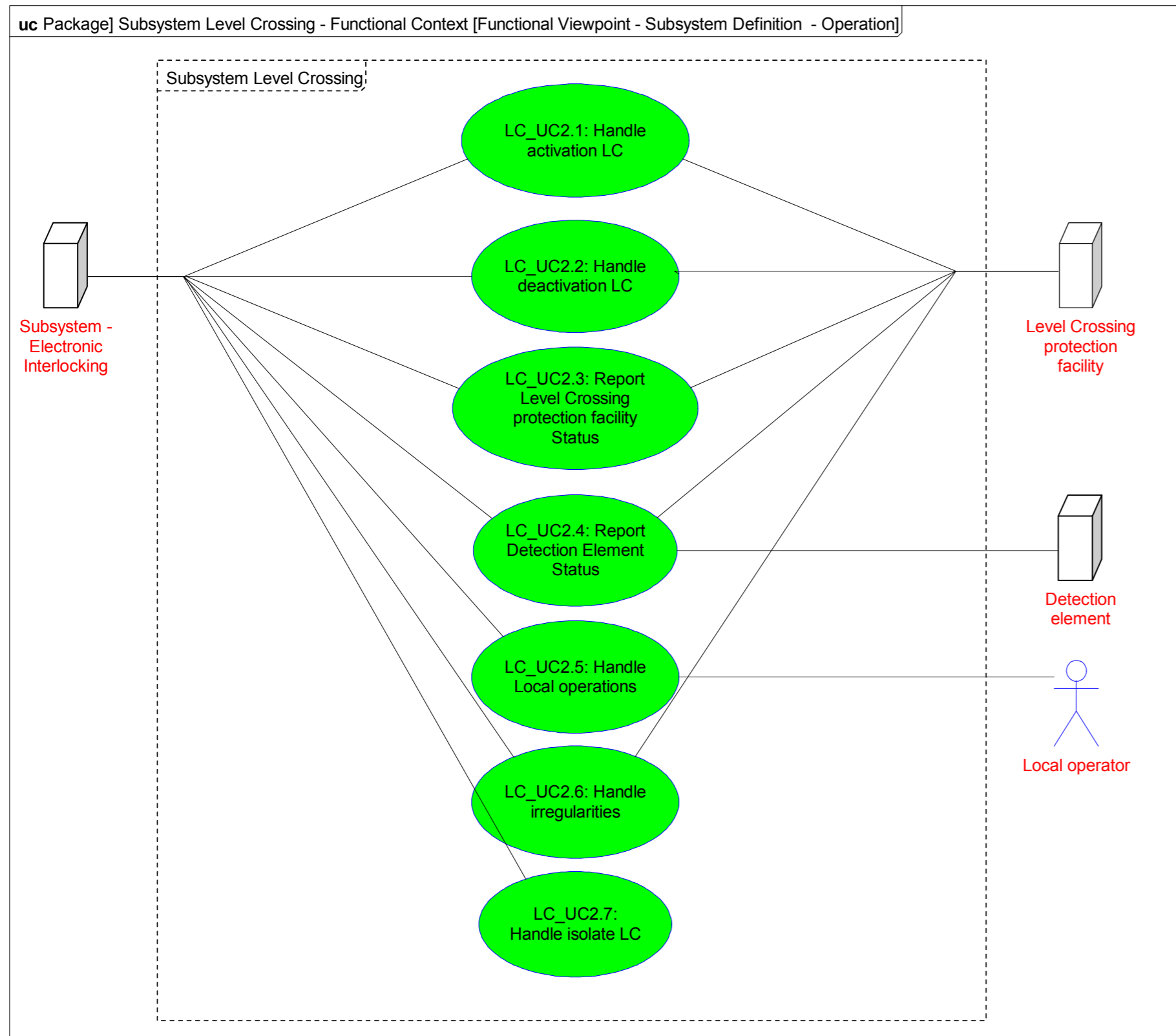
1	Introduction	1
1.1	Release information	1
1.2	Impressum	1
1.3	Purpose	1
1.4	Applicable standards and regulations	1
1.5	Applicable documents	2
1.6	Terms and abbreviations	2
1.7	Variability management	2
1.8	Definition of object types	2
1.9	Modelling	2
2	Conditions of use	2
2.1	Functional packages	2
3	Functional requirements specification	2
3.1	Subsystem Level Crossing - General Infos and Assumptions	2
3.2	Subsystem Level Crossing - Logical Viewpoint	2
3.2.1	Subsystem Level Crossing - Logical Context	2
3.3	Subsystem Level Crossing - Functional Viewpoint	3
3.3.1	Definition of time values	3
3.3.2	Subsystem Level Crossing - Functional Context	3
3.3.3	Subsystem Level Crossing - Functional Partitioning	18
3.3.4	Subsystem Level Crossing - Functional Architecture	19
3.3.5	Subsystem Level Crossing - Functional Entities	21
3.4	Subsystem Level Crossing - Interfaces	33
3.4.1	SCI-LC (Subsystem - Electronic Interlocking)	33
3.4.1.1	SCI-LC - Logical Viewpoint	33
3.4.1.1.1	SCI-LC - Logical Context	33
3.4.1.2	SCI-LC - Information Flows	34
3.4.1.3	SCI-LC - Functional Viewpoint	36
3.4.1.3.1	SCI-LC - Functional Partitioning	36
3.4.1.3.2	SCI-LC - Functional Architecture	36
3.4.1.3.3	SCI-LC - Functional Entities	37
3.4.2	SMI-LC (Subsystem - Maintenance and Data Management)	50
3.4.3	SDI-LC (Subsystem - Maintenance and Data Management)	50
3.4.4	SSI-LC (Subsystem - Security Services Platform)	50
3.4.5	LC1 (Basic Data identifier)	50
3.4.6	LC2 (Maintainer)	50
3.4.7	LC5 (Detection element)	50
3.4.8	LC6 (Local operator)	50
3.4.9	LC4 (Level Crossing protection facility)	50
4	RAMSS requirements	51
5	Technical requirements	51
5.1	Specific technical interface requirements	51
5.1.1	Interface to the Point of Service - Signalling (PoS - Signalling)	51
5.1.2	Interface to the Detection element	51
5.1.3	Interface to the Local operator	51
5.1.4	Interface to the Level Crossing protection facility	51
5.2	Time behaviour	51
5.2.1	Response times	51
5.3	Configuration and engineering data	51
5.3.1	Specific data	51

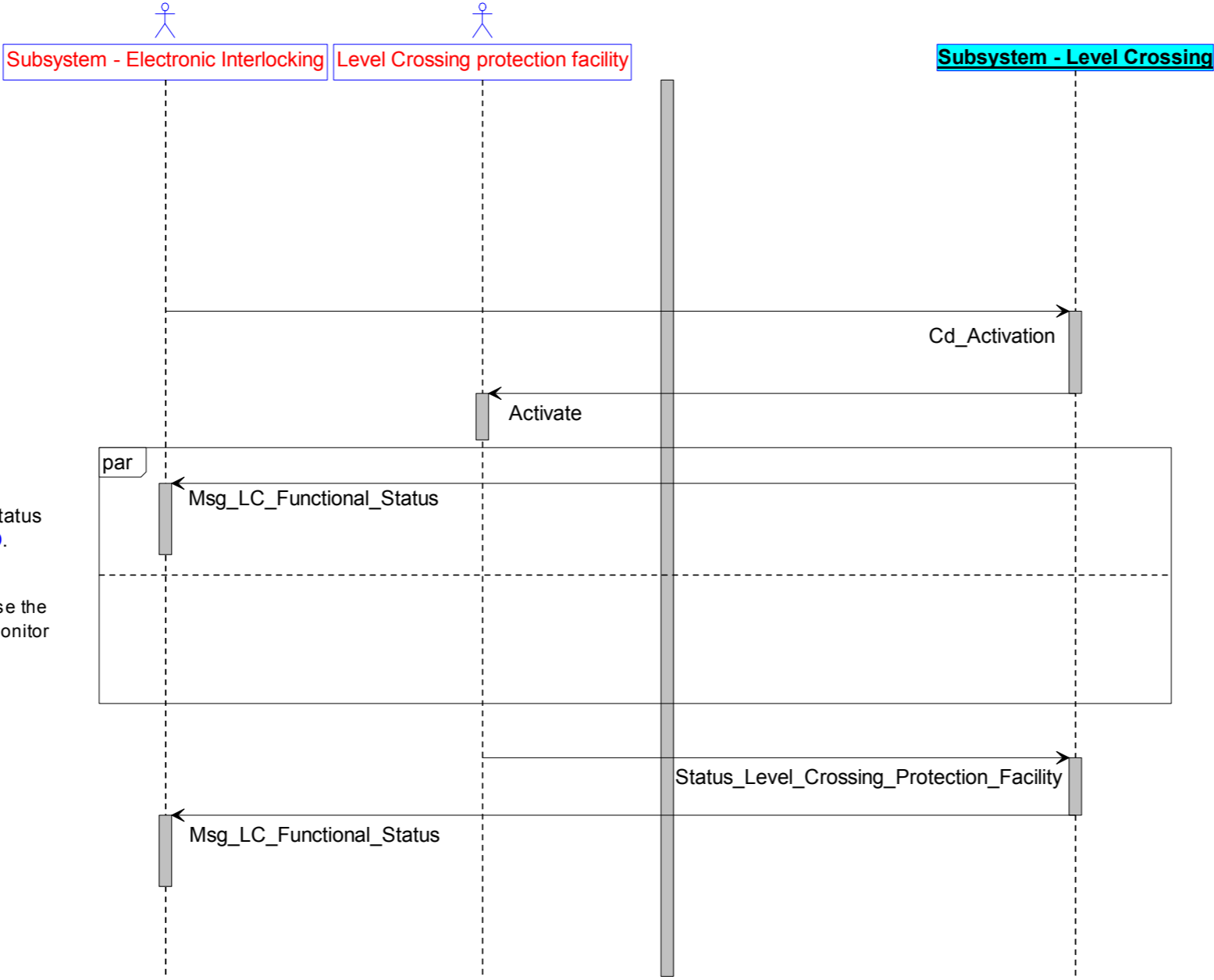
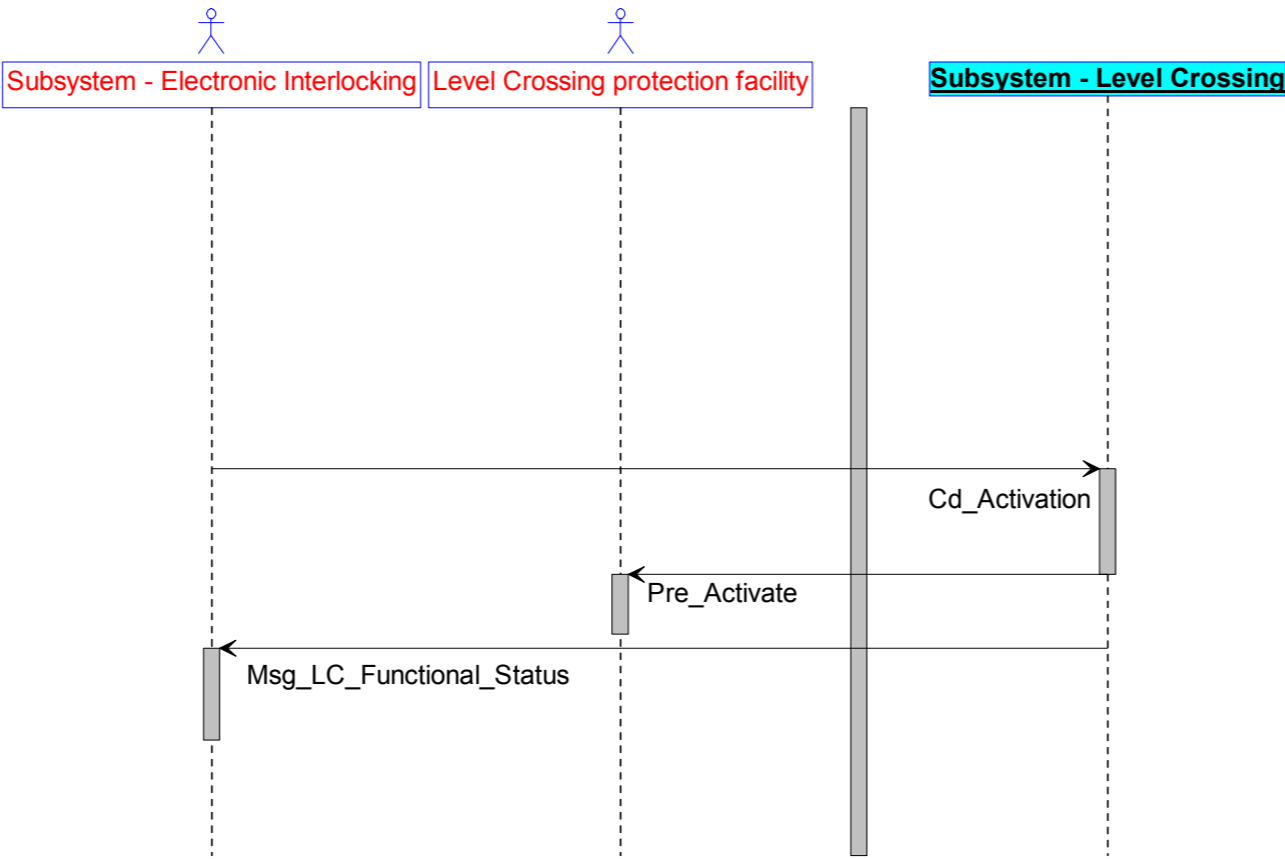
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.1	Head	1 Introduction		
Eu.LC.2	Head	1.1 Release information		
Eu.LC.3	Info	[Eu.Doc.108] Requirements specification for subsystem Level crossing CENELEC Phase: 4 Version: 2.2 (0.A) Approval date: 15.06.2023		
Eu.LC.4	Info	Version history		
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.32	Head	1.2 Impressum		
Eu.LC.33	Info	Publisher: Europe's Rail Joint Undertaking https://rail-research.europa.eu/ EULYNX Initiative A full list of the EULYNX Partners can be found on www.eulynx.eu/index.php/members		
Eu.LC.34	Info	Responsible for this document: EU-Rail System Pillar Trackside Assets Control and Supervision domain		
Eu.LC.35	Info	Copyright EULYNX Partners All information included or disclosed in this document is licensed under the European Union Public Licence EUPL, Version 1.2 or later.		
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: <ul style="list-style-type: none"> • 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].		

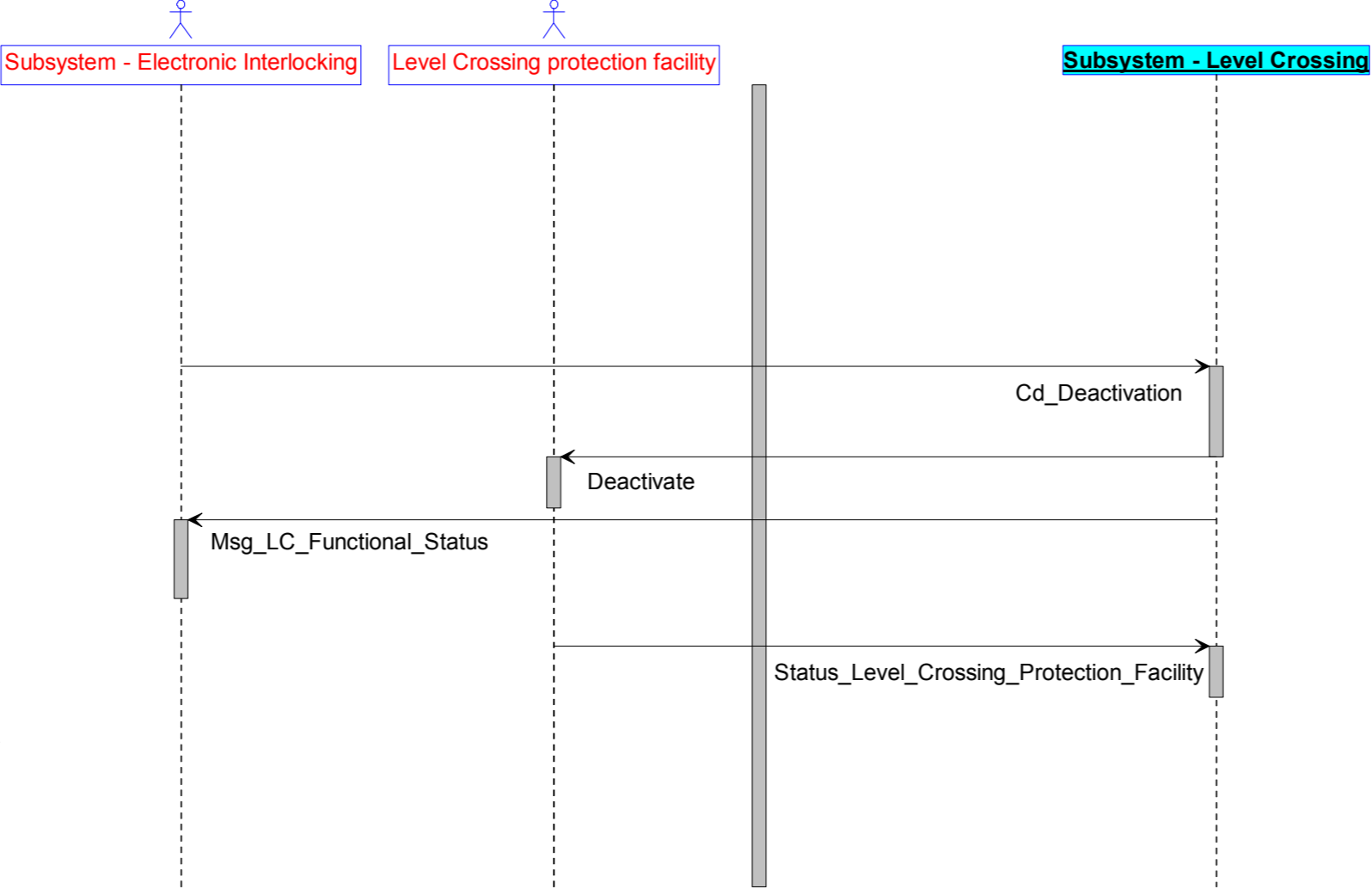
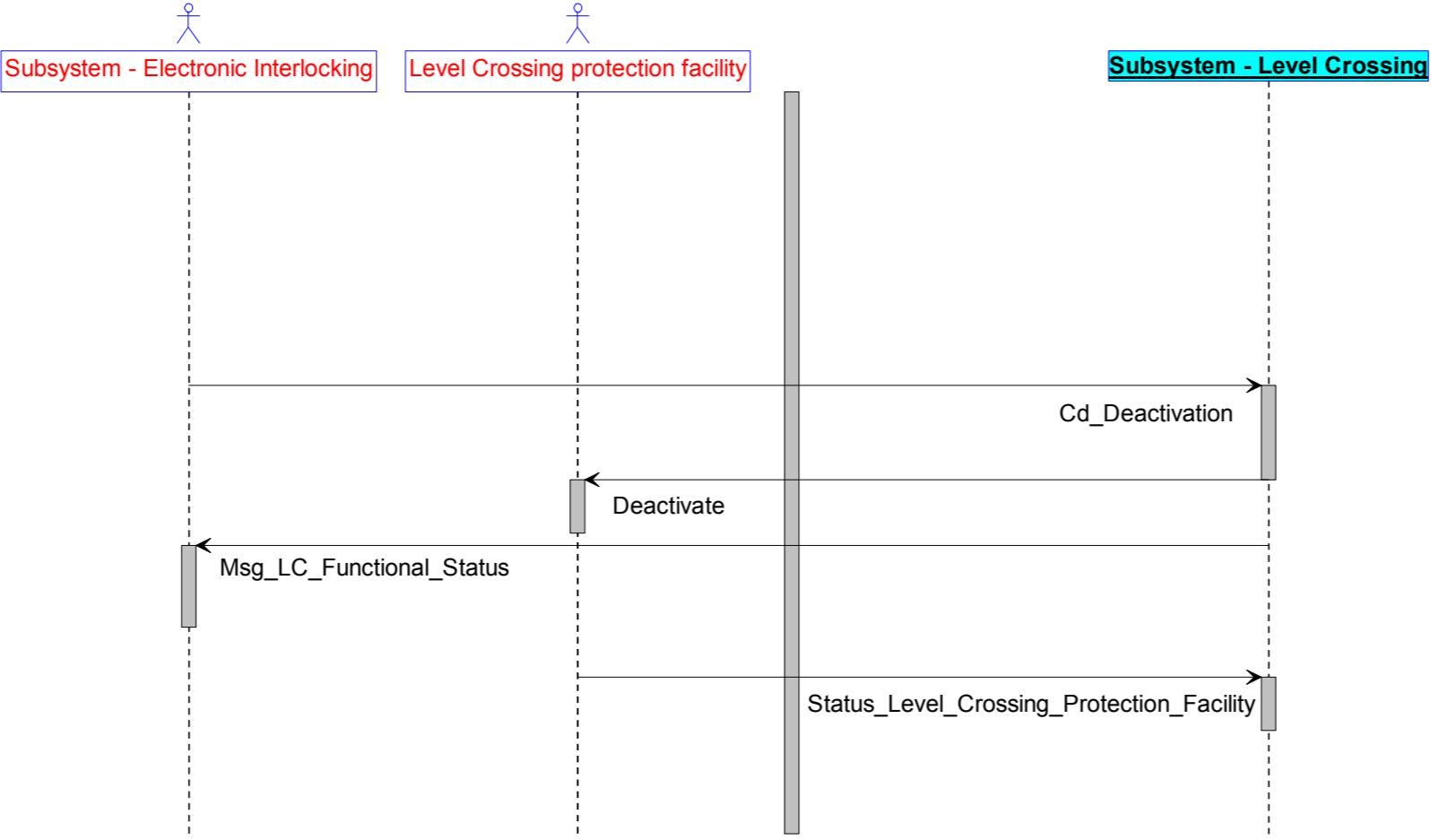
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.43	Head	1.5 Applicable documents		
Eu.LC.44	Info	The current versions of documents used as input or related to this document are listed in the EULYNX Documentation Plan [Eu.Doc.11]. The relationships between the documents are displayed in 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	<ul style="list-style-type: none"> "Req" - This denotes a mandatory requirement. 		
Eu.LC.53	Info	<ul style="list-style-type: none"> "Info" - This denotes additional information to help understand the specification. These objects do not specify any additional requirements. 		
Eu.LC.54	Info	<ul style="list-style-type: none"> "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.		
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.63	Head	2 Conditions of use		
Eu.LC.2371	Req	All references to Eu.Doc.20 refer to version 4.0 (3.A) of that document.		
Eu.LC.3266	Req	All references to Eu.Doc.119 refer to version 1.0 (3.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.2775	Head	3.2 Subsystem Level Crossing - Logical Viewpoint		
Eu.LC.3263	Head	3.2.1 Subsystem Level Crossing - Logical Context		

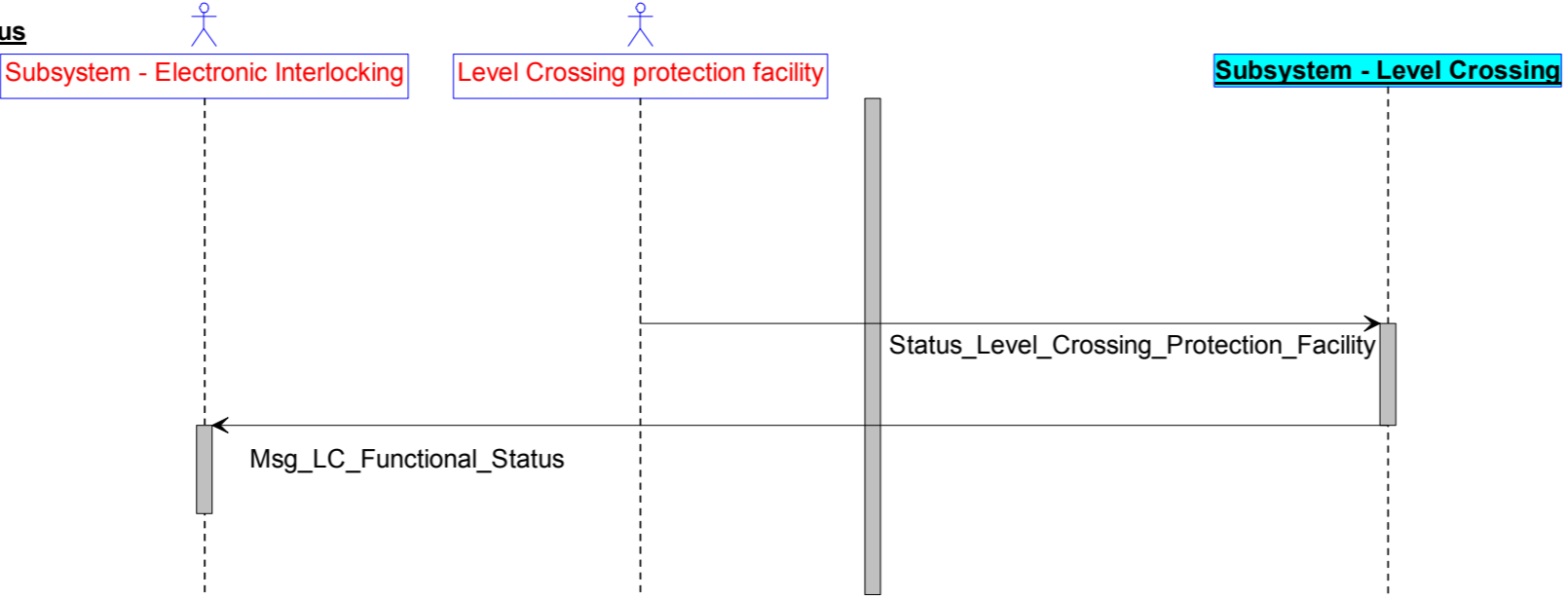
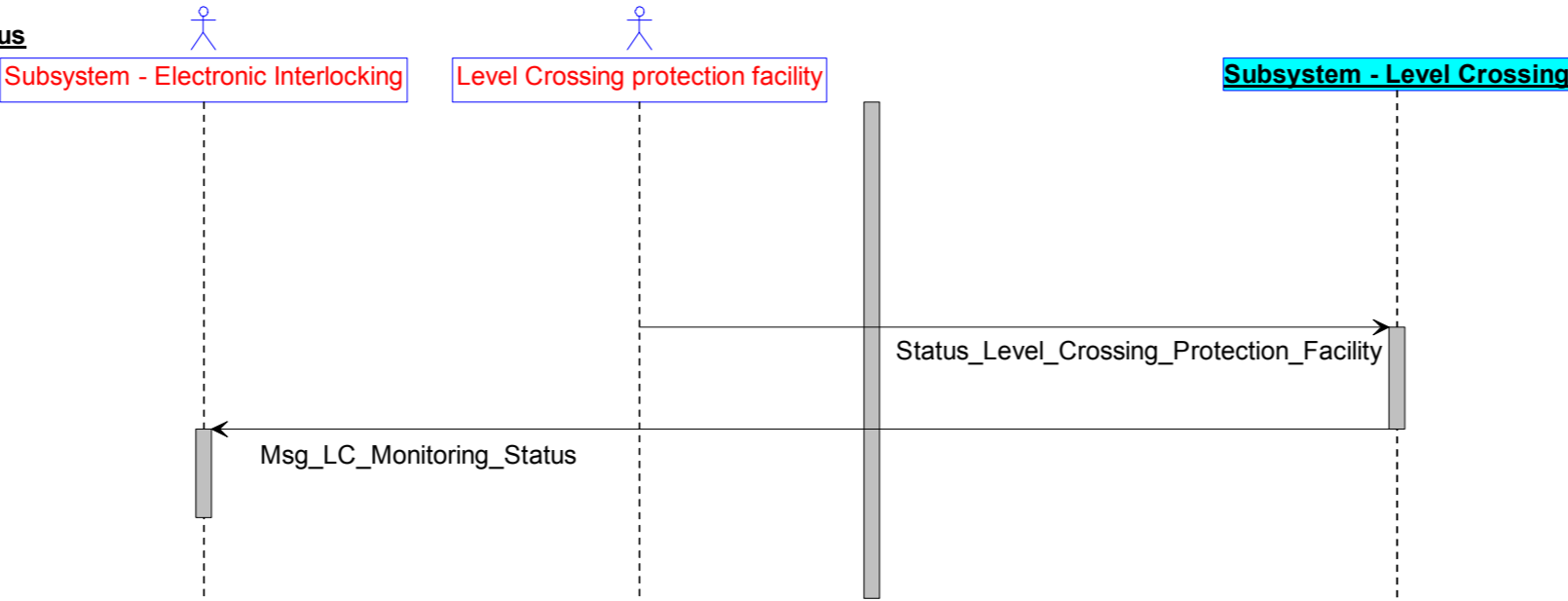
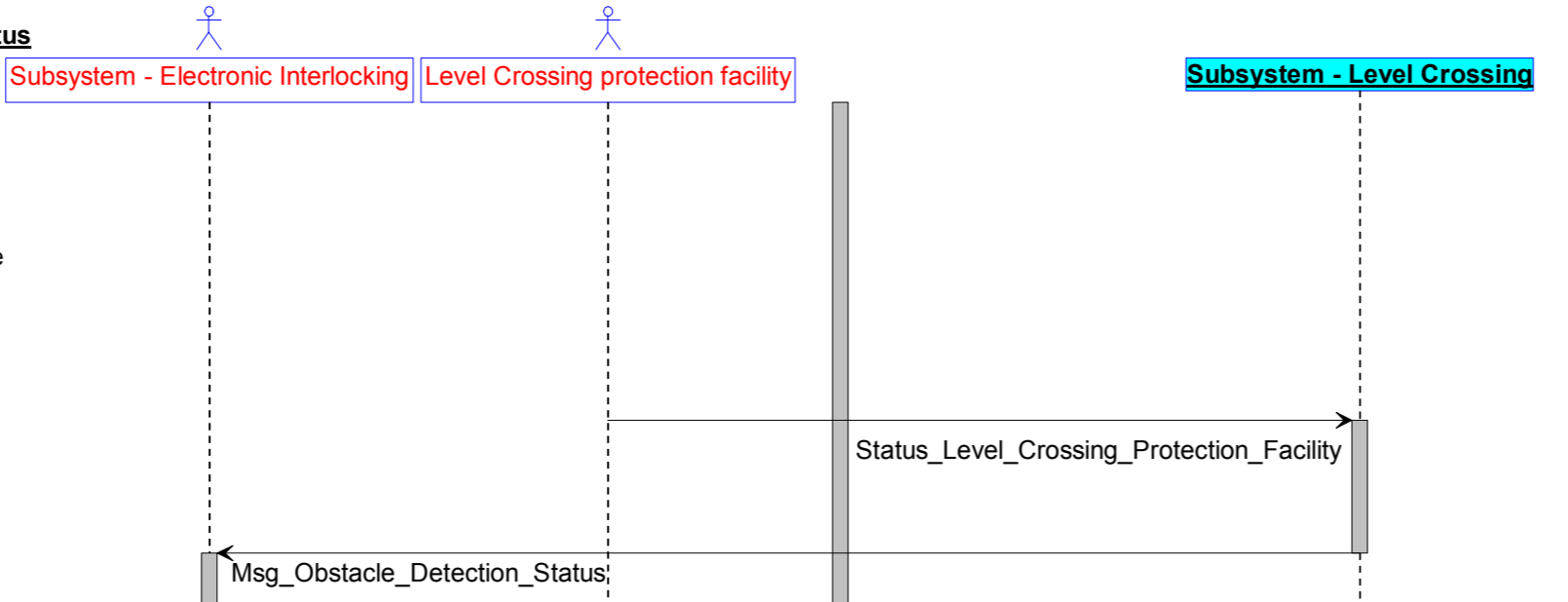
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.2776	Info	<p>[Package] Subsystem Level Crossing- Logical Viewpoint [Subsystem Definition - Logical Context]</p> <p>bdd [Package] Subsystem Level Crossing - Logical Viewpoint [Logical Viewpoint - Subsystem Definition]</p> 		Basic LC
Eu.LC.2373	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	Req	Con_tmax_Closure	"Con_tmax_Closure" is a configurable time value which is specific to a location. The time value provides the permissible activation of the level crossing. Configurable resolution: steps of 1 s Configurable range: from 0 s up to 1800 s	Basic LC
Eu.LC.177	Req	Con_t_PDI_Loss_Deactivation	"Con_t_PDI_Loss_Deactivation" is a configurable time value which is specific to a location. The time value provides the duration between a activation caused by a interrupted Safe communication protocol connection and a deactivation of the Level Crossing protection facility. Configurable resolution: steps of 1 s. Configurable range: from 0 s up to 1800 s.	Basic LC
Eu.LC.2376	Head	3.3.2 Subsystem Level Crossing - Functional Context		

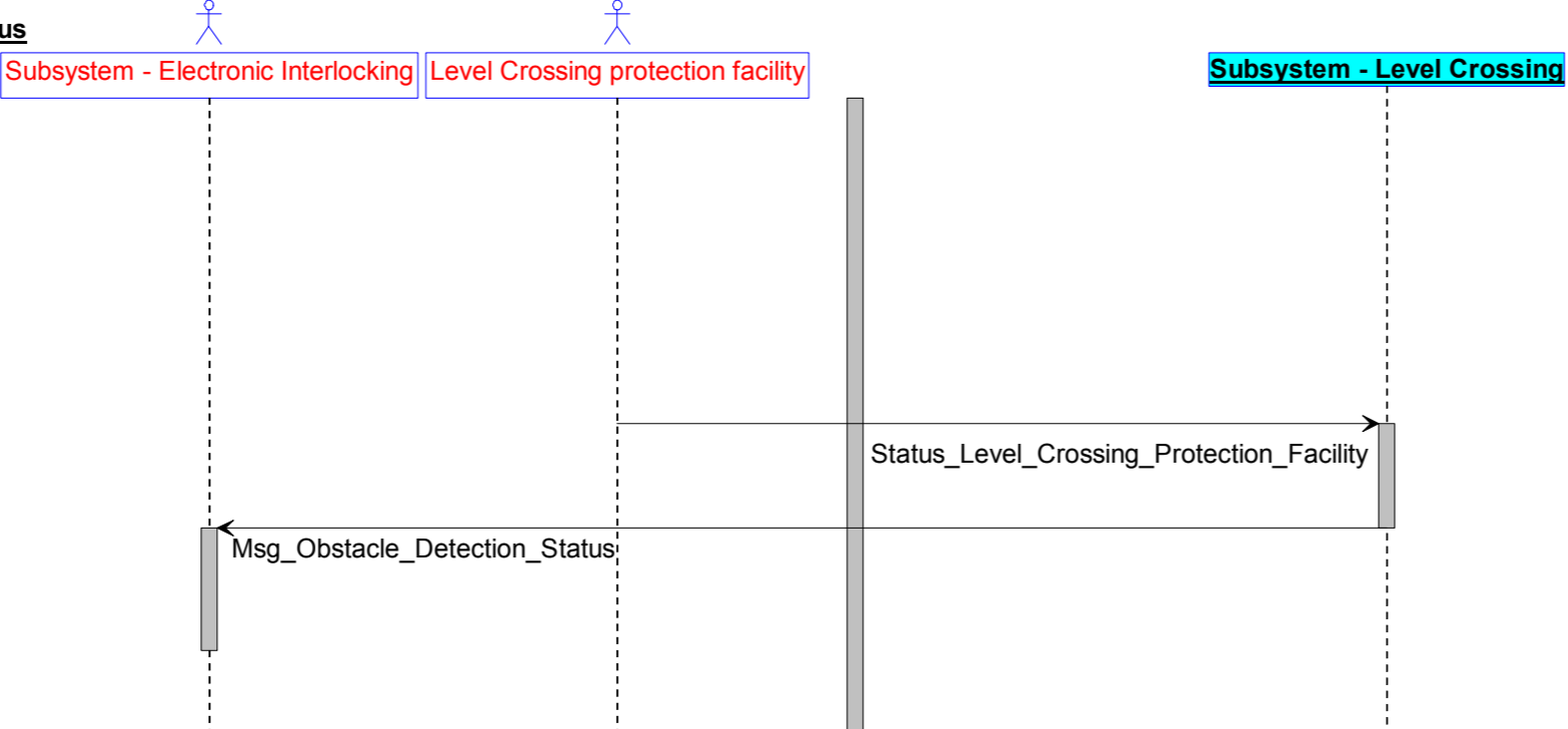
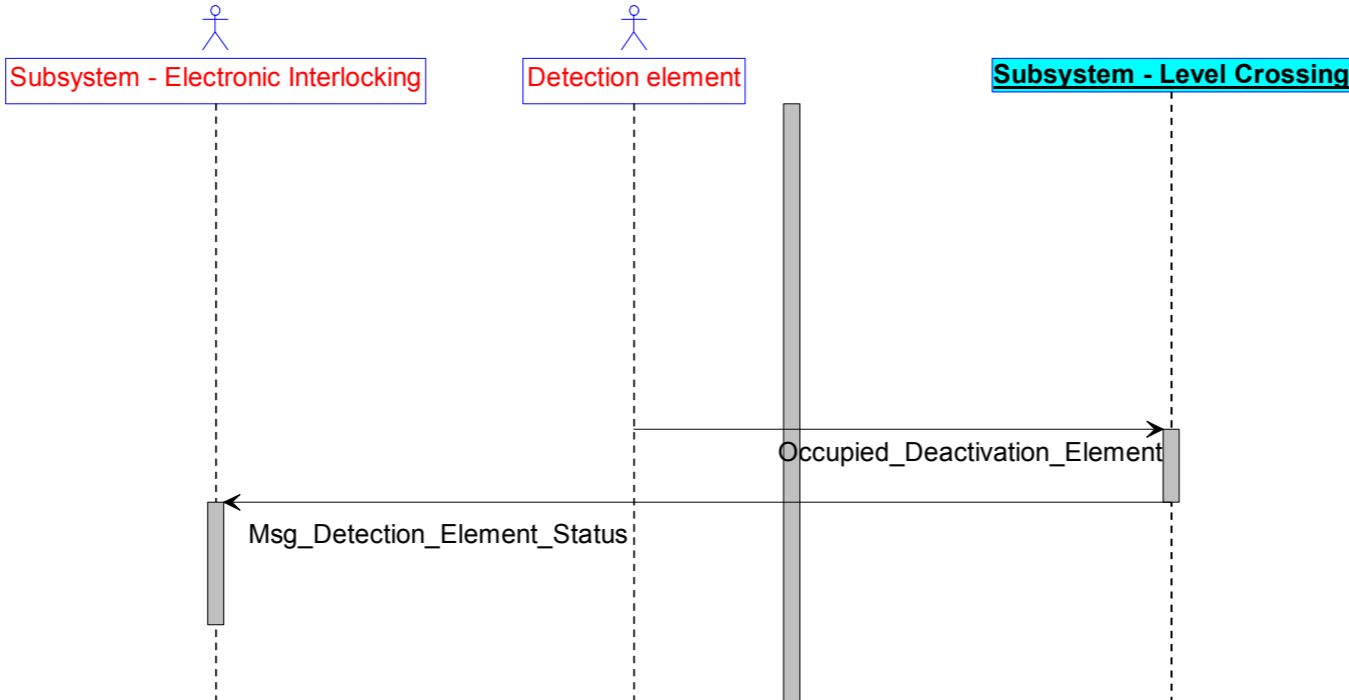
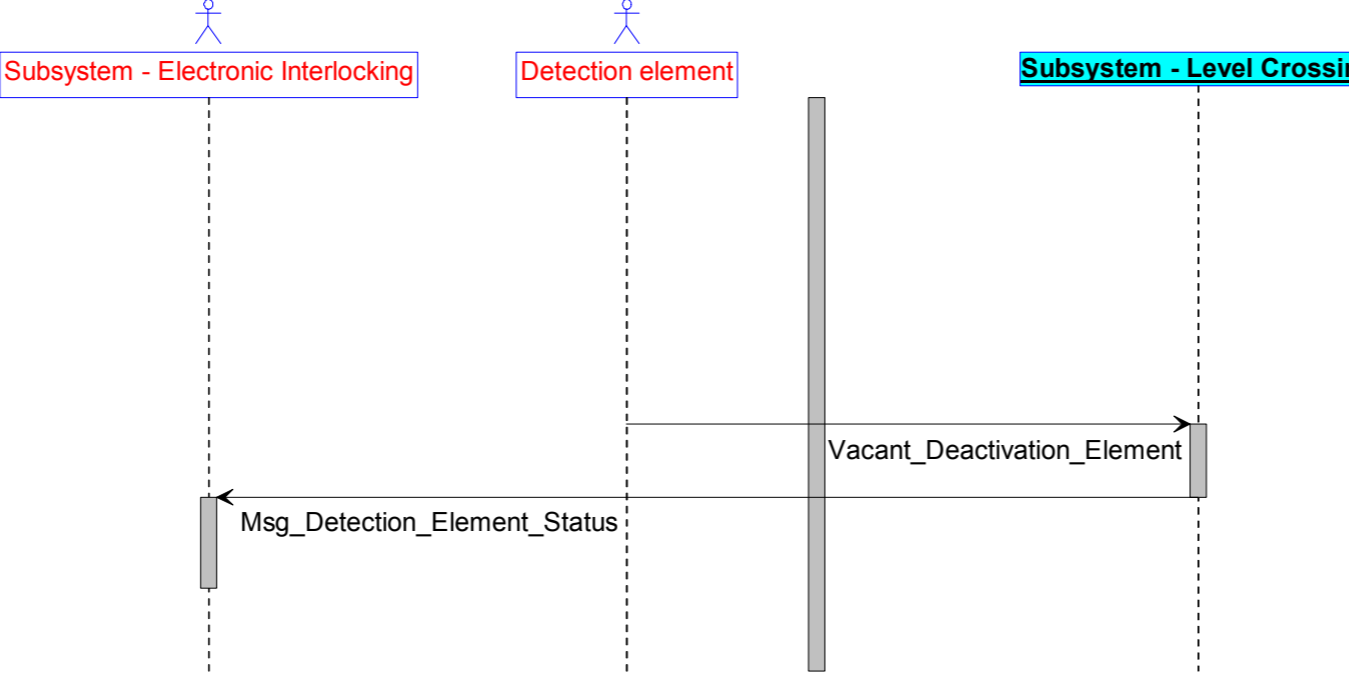
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.189	Info	<p>[Package] Subsystem Level Crossing - Functional Context [UseCase Definition Initialisation]</p> <p>uc [Package] Subsystem Level Crossing - Functional Context [Functional Viewpoint - Subsystem Definition - Initialisation]</p>		Basic LC
Eu.LC.188	Info	<p>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.</p>		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
Eu.LC.191	Info	<p>LC SD 1.3.1</p> <p>LC UC1.3: Report status</p> <p>Main Success Scenario: Report status [LC SD 1.3.1]</p> <p>par</p> <p>1.a1 The Subsystem - Level Crossing reports the current functional status to the Subsystem - Electronic Interlocking.</p> <p>also par</p> <p>1.b1 The Subsystem - Level Crossing reports the current monitoring status to the Subsystem - Electronic Interlocking.</p> <p>also par</p> <p>1.c1 The Subsystem - Level Crossing reports the current failure status to the Subsystem - Electronic Interlocking.</p> <p>also par</p> <p>1.d1 The Subsystem - Level Crossing reports the current status of the Detection element to the Subsystem - Electronic Interlocking.</p> <p>also par</p> <p>1.e1 The Subsystem - Level Crossing reports the current status of the Obstacle detector to the Subsystem - Electronic Interlocking.</p> <p>end par</p>		Basic LC

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
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	<p>LC SD 1.4.1</p> <p>LC UC1.4: Establish initial state of outputs</p> <p>Main Success Scenario: Establish initial state of outputs [LC SD 1.4.1]</p> <p>Precondition: The Subsystem - Level Crossing is in the state BOOTING or INITIALISING. The Initial State Of Outputs has not been established.</p> <p>Interaction 1.4.1.A:</p> <ol style="list-style-type: none"> 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. If the Subsystem - Level Crossing is configured to use the Closure Timer the Subsystem - Level Crossing starts to monitor the time period Con_tmax_Closure. <p>Postcondition: The Subsystem - Level Crossing is in the state ACTIVATED AND UNPROTECTED. Initial State Of Outputs established.</p>  <p>The diagram shows two lifelines: 'Level Crossing protection facility' (actor) and 'Subsystem - Level Crossing' (system). A message arrow labeled 'Activate' points from the Subsystem - Level Crossing to the Level Crossing protection facility.</p>		Basic LC
Eu.LC.225	Info	<p>[Package] Subsystem Level Crossing - Functional Context [UseCase Definition - Operation]</p> <p>uc Package] Subsystem Level Crossing - Functional Context [Functional Viewpoint - Subsystem Definition - Operation]</p>  <p>The diagram shows a central dashed box labeled 'Subsystem Level Crossing' containing seven use cases: LC_UC2.1: Handle activation LC, LC_UC2.2: Handle deactivation LC, LC_UC2.3: Report Level Crossing protection facility Status, LC_UC2.4: Report Detection Element Status, LC_UC2.5: Handle Local operations, LC_UC2.6: Handle irregularities, and LC_UC2.7: Handle isolate LC. On the left, a box labeled 'Subsystem - Electronic Interlocking' is connected to all use cases. On the right, three external entities are connected: 'Level Crossing protection facility' (connected to LC_UC2.1, 2, 3), 'Detection element' (connected to LC_UC2.4), and 'Local operator' (connected to LC_UC2.5).</p>		Basic LC

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
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
Eu.LC.228	Info	<p>LC SD 2.1.1</p> <p>LC UC2.1: Handle activation LC</p> <p>Main Success Scenario: Activate [LC SD 2.1.1]</p> <p>Precondition: The Subsystem - Level Crossing is in the state OPERATIONAL. The Subsystem - Level Crossing is either in the state DEACTIVATED AND UNPROTECTED or PRE-ACTIVATED.</p> <p>Interaction 2.1.1.A:</p> <ol style="list-style-type: none"> - The Subsystem - Level Crossing receives from the Subsystem - Electronic Interlocking the command to activate the Level Crossing protection facility. - The Subsystem - Level Crossing activates the Level Crossing protection facility. <p>par</p> <ol style="list-style-type: none"> 3.a1 The Subsystem - Level Crossing reports to the Subsystem - Electronic Interlocking that the functional status has been changed to ACTIVATED AND UNPROTECTED. <p>also par</p> <ol style="list-style-type: none"> 3.b1 If the Subsystem - Level Crossing is configured to use the Closure Timer the Subsystem - Level Crossing starts to monitor the time period Con_tmax_Closure. <p>end par</p> <p>Interaction 2.1.1.B:</p> <ol style="list-style-type: none"> - The Level Crossing protection facility reports the new status protected to the Subsystem - Level Crossing. - The Subsystem - Level Crossing reports to the Subsystem - Electronic Interlocking that the functional status has been changed to ACTIVATED AND PROTECTED. <p>Postcondition: The Subsystem - Level Crossing is in the state ACTIVATED AND PROTECTED.</p>	 <pre> sequenceDiagram actor Actor1 actor Actor2 participant SIE as Subsystem - Electronic Interlocking participant LCPF as Level Crossing protection facility participant SLC as Subsystem - Level Crossing SIE->>SLC: Cd_Activation activate SLC SLC->>LCPF: Activate activate LCPF LCPF->>SIE: Msg_LC_Functional_Status deactivate LCPF SLC->>SIE: Msg_LC_Functional_Status deactivate SLC LCPF->>SLC: Status_Level_Crossing_Protection_Facility activate LCPF deactivate LCPF </pre>	Basic LC
Eu.LC.321	Info	<p>LC SD 2.1.2</p> <p>LC UC2.1: Handle activation LC</p> <p>Alternative Scenario: Pre-activate [LC SD 2.1.2]</p> <p>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 UNPROTECTED.</p> <p>Interaction 2.1.2.A:</p> <ol style="list-style-type: none"> - The Subsystem - Level Crossing receives from the Subsystem - Electronic Interlocking the command to pre-activate the Level Crossing protection facility. - The Subsystem - Level Crossing pre-activates the Level Crossing protection facility. - The Subsystem - Level Crossing reports to the Subsystem - Electronic Interlocking that the functional status has been changed to PRE-ACTIVATED. <p>Postcondition: The Subsystem - Level Crossing is in state PRE-ACTIVATED.</p>	 <pre> sequenceDiagram actor Actor1 actor Actor2 participant SIE as Subsystem - Electronic Interlocking participant LCPF as Level Crossing protection facility participant SLC as Subsystem - Level Crossing SIE->>SLC: Cd_Activation activate SLC SLC->>LCPF: Pre_Activate activate LCPF LCPF->>SIE: Msg_LC_Functional_Status deactivate LCPF SLC->>SIE: Msg_LC_Functional_Status deactivate SLC </pre>	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

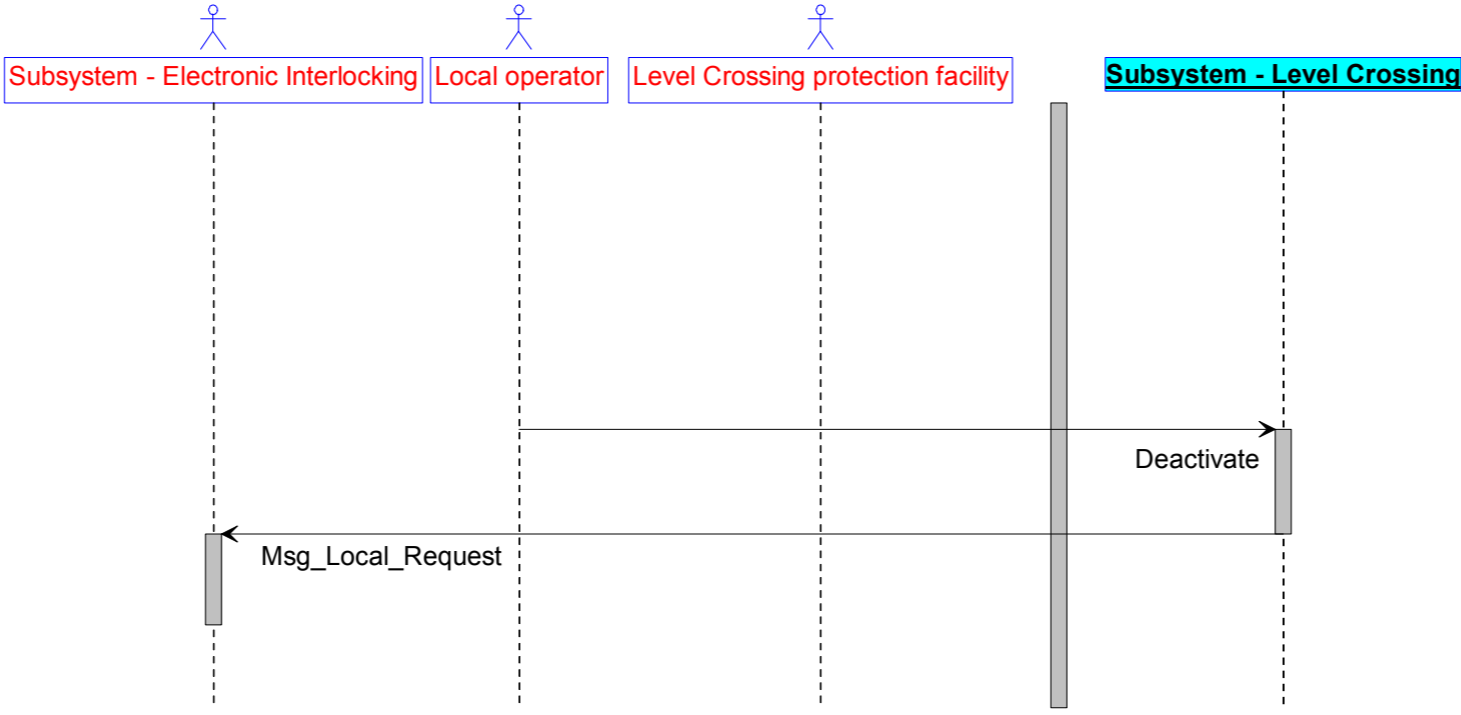
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.447	Info	<p>LC SD 2.2.1</p> <p>LC UC2.2: Handle deactivation LC</p> <p>Main Success Scenario: Deactivate after Activation [LC SD 2.2.1]</p> <p>Precondition: The Subsystem - Level Crossing is in the state OPERATIONAL. The Subsystem - Level Crossing is in the state ACTIVATED AND UNPROTECTED or ACTIVATED AND PROTECTED.</p> <p>Interaction 2.2.1.A:</p> <ol style="list-style-type: none"> - The Subsystem - Level Crossing receives from the Subsystem - Electronic Interlocking the command to deactivate the Level Crossing protection facility. - The Subsystem - Level Crossing deactivates the Level Crossing protection facility. - The Subsystem - Level Crossing reports to the Subsystem - Electronic Interlocking that the functional status has been to DEACTIVATED AND UNPROTECTED. <p>Interaction 2.2.1.B:</p> <ol style="list-style-type: none"> - The Level Crossing protection facility reports the new status idle to the Subsystem - Level Crossing. - If the Subsystem - Level Crossing is configured to use the Closure Timer the Subsystem - Level Crossing stops to monitor the time period Con_tmax_Closure. <p>Postcondition: The Subsystem - Level Crossing is in state DEACTIVATED AND UNPROTECTED.</p> 		Basic LC
Eu.LC.3865		<p>LC SD 2.2.2</p> <p>LC UC2.2: Handle deactivation LC</p> <p>Alternative Scenario: Deactivate after Pre-Activation [LC SD 2.2.2]</p> <p>Precondition: The Subsystem - Level Crossing is in the state OPERATIONAL. The Subsystem - Level Crossing is in the state PRE-ACTIVATED.</p> <p>Interaction 2.2.1.A:</p> <ol style="list-style-type: none"> - The Subsystem - Level Crossing receives from the Subsystem - Electronic Interlocking the command to deactivate the Level Crossing protection facility. - The Subsystem - Level Crossing deactivates the Level Crossing protection facility. - The Subsystem - Level Crossing reports to the Subsystem - Electronic Interlocking that the functional status has been to DEACTIVATED AND UNPROTECTED. <p>Interaction 2.2.1.B:</p> <ol style="list-style-type: none"> - The Level Crossing protection facility reports the new status idle to the Subsystem - Level Crossing. <p>Postcondition: The Subsystem - Level Crossing is in state DEACTIVATED AND UNPROTECTED.</p> 		
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	<p>LC SD 2.3.1</p> <p>LC UC2.3: Report Level Crossing protection facility Status</p>  <p>Alternative Scenario: Report Functional Status [LC SD 2.3.1]</p> <p>Precondition: The Subsystem - Level Crossing is in the state OPERATIONAL.</p> <p>Interaction 2.3.1.A:</p> <ol style="list-style-type: none"> - The Level Crossing protection facility reports to the Subsystem - Level Crossing a change in the functional parameters of the Level Crossing protection facility. - The Subsystem - Level Crossing reports to the Subsystem - Electronic Interlocking the new functional status of the Level Crossing protection facility. <p>Postcondition: ---</p>		Basic LC
Eu.LC.571	Info	<p>LC SD 2.3.2</p> <p>LC UC2.3: Report Level Crossing protection facility Status</p>  <p>Alternative Scenario: Report Monitoring Status [LC SD 2.3.2]</p> <p>Precondition: The Subsystem - Level Crossing is in the state OPERATIONAL.</p> <p>Interaction 2.3.2.A:</p> <ol style="list-style-type: none"> - The Level Crossing protection facility reports to the Subsystem - Level Crossing a change in the monitoring parameters of the Level Crossing protection facility. - The Subsystem - Level Crossing reports to the Subsystem - Electronic Interlocking the new monitoring status of the Level Crossing protection facility. <p>Postcondition: ---</p>		Basic LC
Eu.LC.604	Info	<p>LC SD 2.3.3</p> <p>LC UC2.3: Report Level Crossing protection facility Status</p>  <p>Alternative Scenario: Detect obstacle in the Level Crossing area [LC SD 2.3.3]</p> <p>Precondition: The Subsystem - Level Crossing is configured to use obstacle detection. The Subsystem - Level Crossing is in the state OPERATIONAL.</p> <p>Interaction 2.3.3.A:</p> <ol style="list-style-type: none"> - The Level Crossing protection facility reports to the Subsystem - Level Crossing that an obstacle is detected in the Level Crossing area. The conditions for activation are not fulfilled. - 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. <p>Postcondition: ---</p>	The Level Crossing protection facility monitors whether the Obstacle detector shall be activated or not.	Basic LC

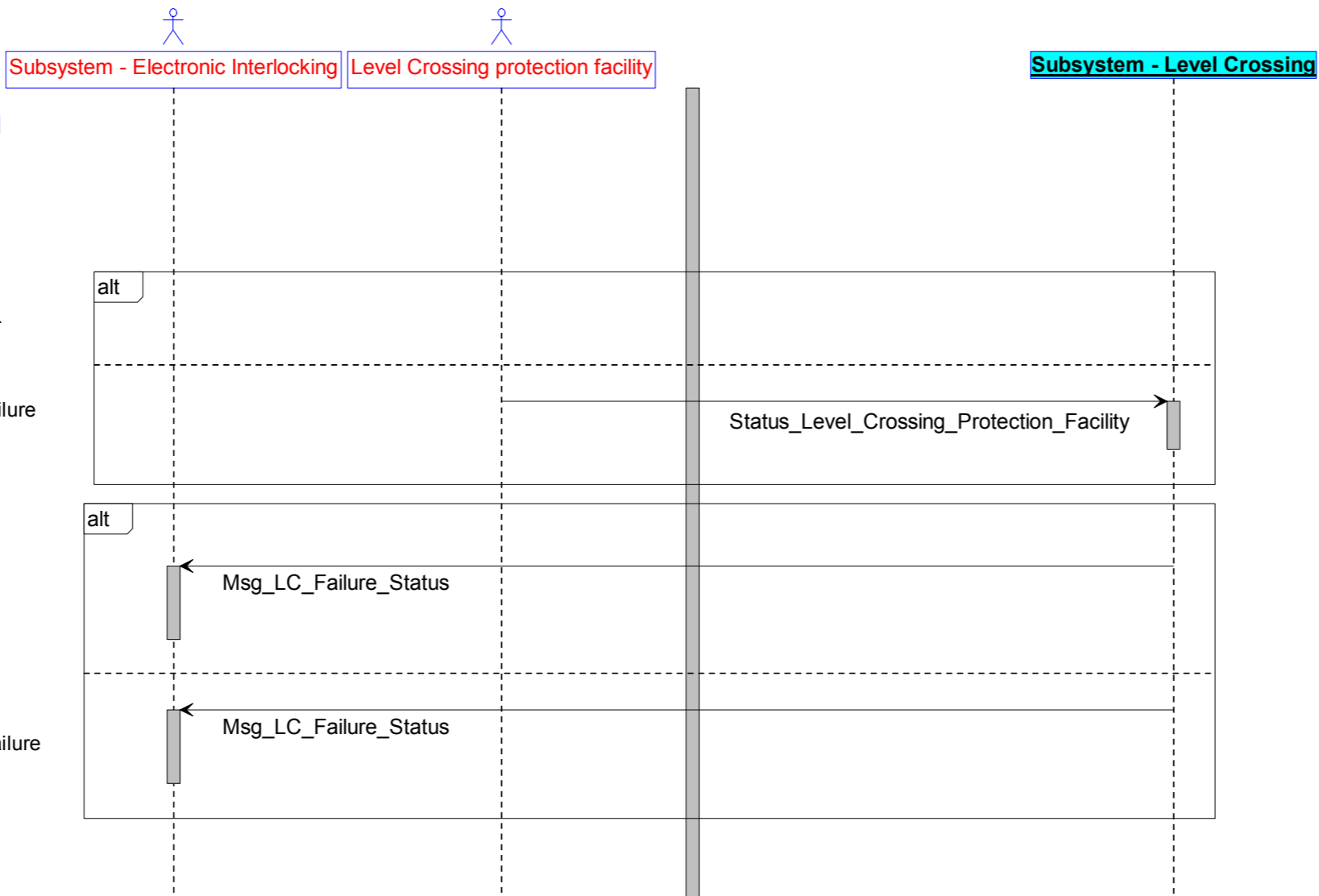
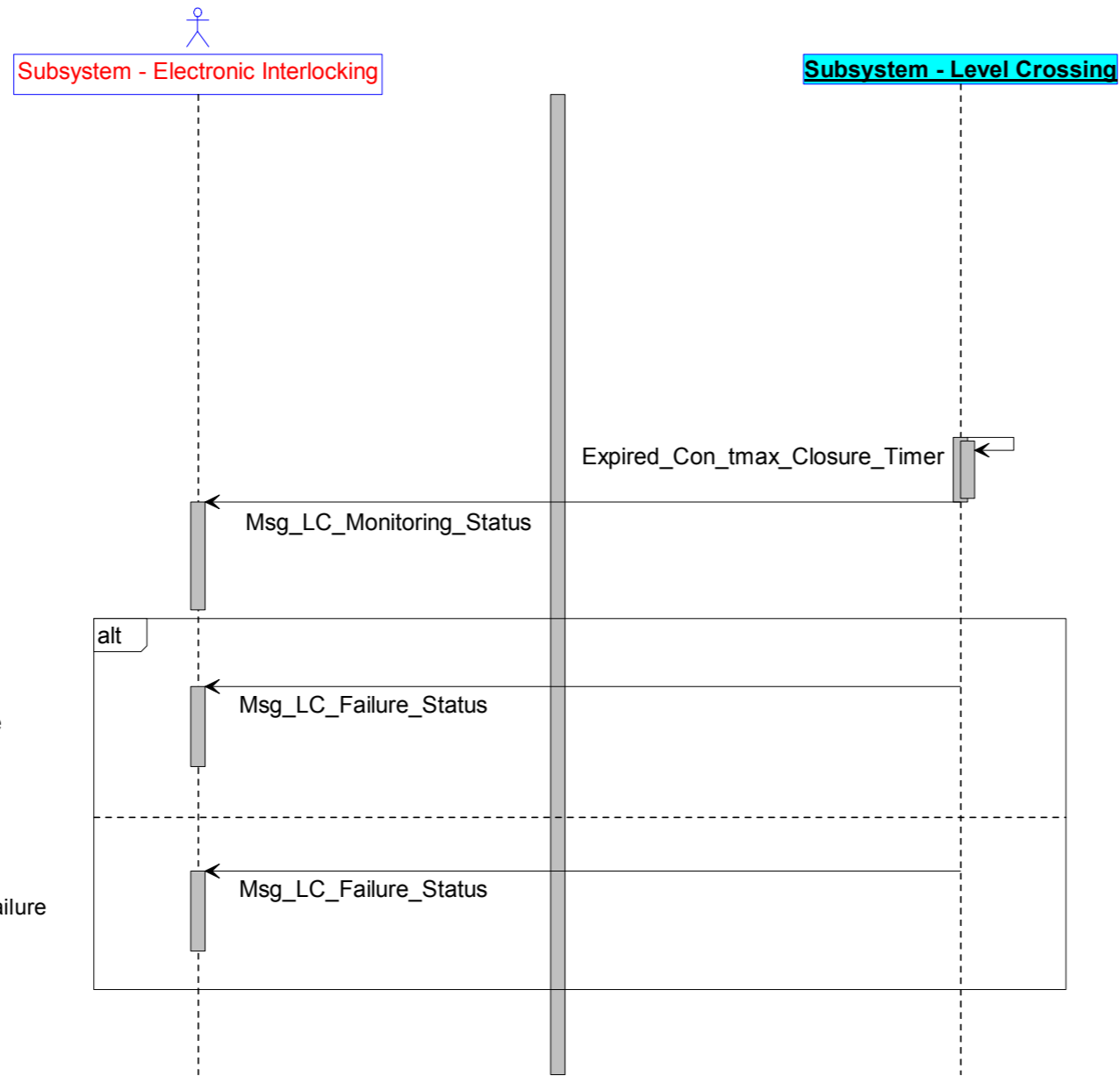
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.3304	Info	<p>LC SD 2.3.4</p> <p>LC UC2.3: Report Level Crossing protection facility Status</p>  <p>Alternative Scenario: Detect no obstacle in the Level Crossing area [LC SD 2.3.4]</p> <p>Precondition: The Subsystem - Level Crossing is configured to use obstacle detection. The Subsystem - Level Crossing is in the state OPERATIONAL.</p> <p>Interaction 2.3.4.A:</p> <ol style="list-style-type: none"> - The Level Crossing protection facility reports to the Subsystem - Level Crossing that no obstacle is detected in the Level Crossing area. - 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. <p>Postcondition: ---</p>	<p>The Level Crossing protection facility monitors whether the Obstacle detector shall be activated or not.</p>	Basic LC
Eu.LC.615	Info	<p>LC_UC2.4: Report Detection Element Status</p>	<p>The Subsystem-UseCase LC_UC2.4: Report Detection Element Status defines reporting of the statuses of the Detection element to the Subsystem - Level Crossing.</p>	Basic LC
Eu.LC.616	Info	<p>LC SD 2.4.1</p> <p>LC UC2.4: Report Detection Element Status</p>  <p>Alternative Scenario: Report occupied Detection element [LC SD 2.4.1]</p> <p>Precondition: The Subsystem - Level Crossing is in state the OPERATIONAL. The Subsystem - Level Crossing is configured to use the Detection element.</p> <p>Interaction 2.4.1.A:</p> <ol style="list-style-type: none"> - The Subsystem - Level Crossing detects that the Detection element is occupied. - 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. <p>Postcondition: ---</p>		Basic LC
Eu.LC.625	Info	<p>LC SD 2.4.2</p> <p>LC UC2.4: Report Detection Element Status</p>  <p>Alternative Scenario: Report vacant Detection element [LC SD 2.4.2]</p> <p>Precondition: The Subsystem - Level Crossing is in state the OPERATIONAL. The Subsystem - Level Crossing is configured to use the Detection element.</p> <p>Interaction 2.4.2.A:</p> <ol style="list-style-type: none"> - The Subsystem - Level Crossing detects that the Detection element is vacant. - 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. <p>Postcondition: ---</p>		Basic LC

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.634	Info	<p>LC SD 2.4.3</p> <p>LC UC2.4: Report Detection Element Status</p> <p>Alternative Scenario: Report failed detection element [LC SD 2.4.3]</p> <p>Precondition: The Subsystem - Level Crossing is in the state OPERATIONAL. The Subsystem - Level Crossing is configured to use the Detection element.</p> <p>Interaction 2.4.3.A:</p> <ol style="list-style-type: none"> - The Subsystem - Level Crossing detects a technical failure for the detection element related deactivation of the Level Crossing protection facility. - 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. <p>Postcondition: --</p>		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	<p>LC SD 2.5.1</p> <p>LC UC2.5: Handle Local operations</p> <p>Alternative Scenario: Receiving a command that the handover of local operations is initiated [LC SD 2.5.1]</p> <p>Precondition: The Subsystem - Level Crossing is configured to use Local Operation. The Subsystem - Level Crossing is in the state OPERATIONAL.</p> <p>Interaction 2.5.1.A:</p> <ol style="list-style-type: none"> - 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. - 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). <p>Postcondition: --</p>		Option LOH
Eu.LC.653	Info	<p>LC SD 2.5.2</p> <p>LC UC2.5: Handle Local operations</p> <p>Alternative Scenario: Receiving an input to allow the handover of local operations by Local operator [LC SD 2.5.2]</p> <p>Precondition: The Subsystem - Level Crossing is configured to use Local Operation. The Subsystem - Level Crossing is in the state OPERATIONAL.</p> <p>Interaction 2.5.2.A:</p> <ol style="list-style-type: none"> - The Subsystem - Level Crossing receives the input from the Local operator to allow handover of local operation for a specific index (e.g. track). - The Subsystem - Level Crossing reports to the Subsystem - Electronic Interlocking that the Local operator has allowed the handover of local operations. <p>Postcondition: --</p>		Option LOH

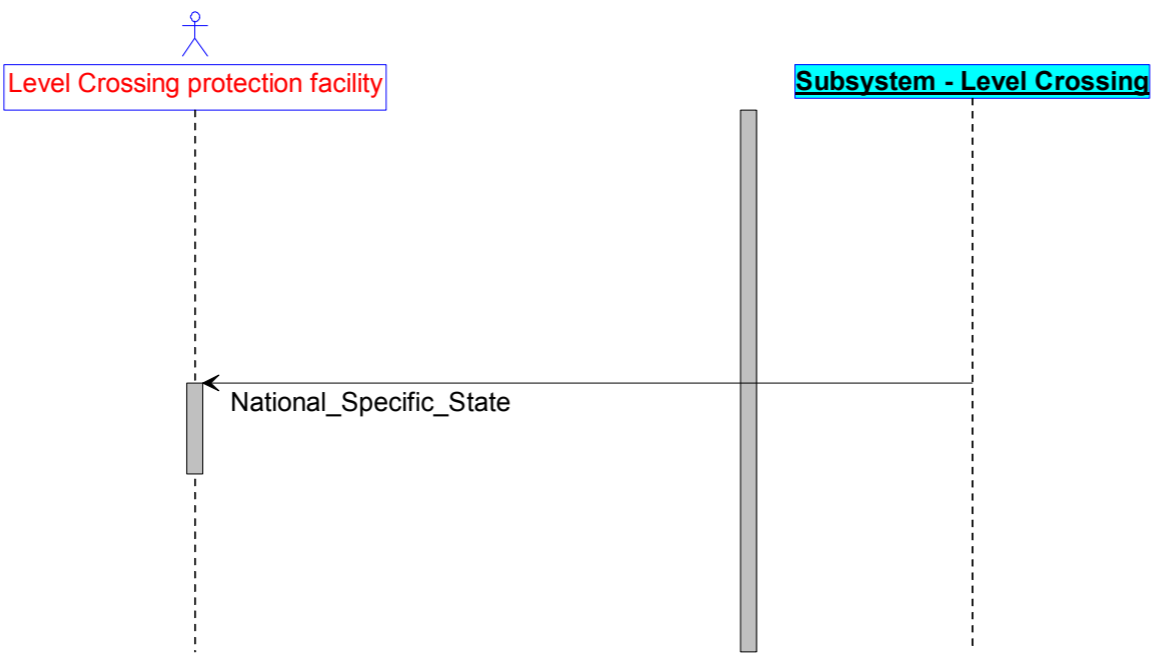
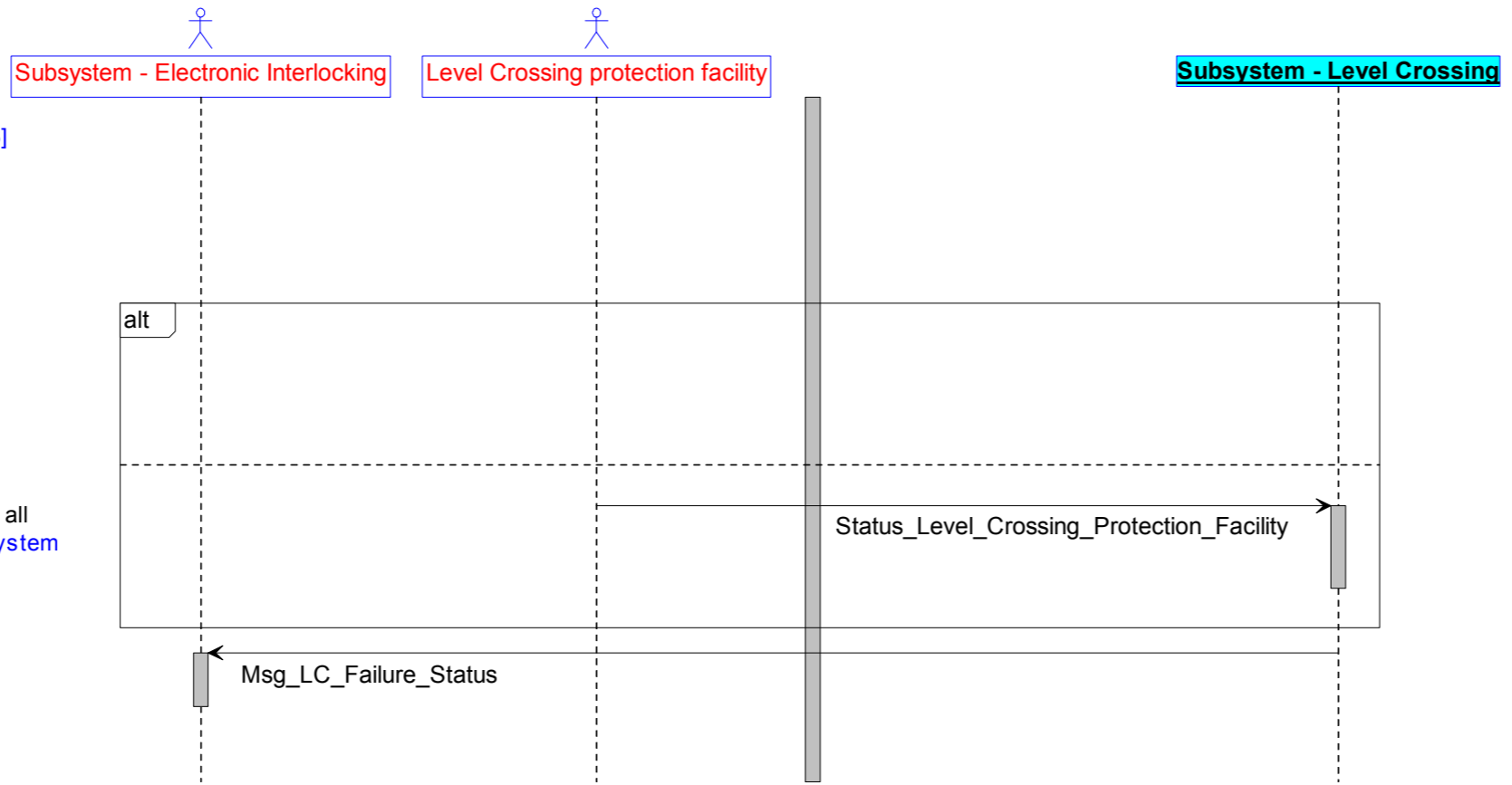
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.664	Info	<p>LC SD 2.5.3</p> <p>LC UC2.5: Handle Local operations</p> <p>Alternative Scenario: Receiving a command that the handover of local operations is established [LC SD 2.5.3]</p> <p>Precondition: The Subsystem - Level Crossing is configured to use Local Operation. The Subsystem - Level Crossing is in the state OPERATIONAL.</p> <p>Interaction 2.5.3.A:</p> <ol style="list-style-type: none"> - 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. - 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). <p>Postcondition: --</p>		Option LOH
Eu.LC.673	Info	<p>LC SD 2.5.4</p> <p>LC UC2.5: Handle Local operations</p> <p>Alternative Scenario: Receiving an input to return the handover of local operations by Local operator [LC SD 2.5.4]</p> <p>Precondition: The Subsystem - Level Crossing is configured to use Local Operation. The Subsystem - Level Crossing is in the state OPERATIONAL.</p> <p>Interaction 2.5.4.A:</p> <ol style="list-style-type: none"> - 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). - The Subsystem - Level Crossing reports to the Subsystem - Electronic Interlocking that the Local operator has returned the handover of local operations. <p>Postcondition: --</p>		Option LOH
Eu.LC.704	Info	<p>LC SD 2.5.5</p> <p>LC UC2.5: Handle Local operations</p> <p>Alternative Scenario: Receiving a command that the handover of local operations is returned [LC SD 2.5.5]</p> <p>Precondition: The Subsystem - Level Crossing is configured to use Local Operation. The Subsystem - Level Crossing is in the state OPERATIONAL.</p> <p>Interaction 2.5.5.A:</p> <ol style="list-style-type: none"> - 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. - 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). <p>Postcondition: --</p>		Option LOH

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.741	Info	<p>LC SD 2.5.6</p> <p>LC UC2.5: Handle Local operations</p> <p>Alternative Scenario: Activate the level crossing via local operations [LC SD 2.5.6]</p> <p>Precondition: The Subsystem - Level Crossing is configured to use Local Operation. The Subsystem - Level Crossing is in the state OPERATIONAL.</p> <p>Interaction 2.5.6.A:</p> <ol style="list-style-type: none"> - The Subsystem - Level Crossing detects a request for an activation from the Local operator for activating the Level Crossing protection facility. - The Subsystem - Level Crossing reports to the Subsystem - Electronic Interlocking that a local operator has requested a local activation. <p>Postcondition: ---</p> 	<p>Note: If the conditions for a local operation are fulfilled the Main Success Scenario: Activate LC SD 2.1.1 will be started.</p>	Option LOH
Eu.LC.753	Info	<p>LC SD 2.5.7</p> <p>LC UC2.5: Handle Local operations</p> <p>Alternative Scenario: Deactivate the level crossing via local operations [LC SD 2.5.7]</p> <p>Precondition: The Subsystem - Level Crossing is configured to use Local Operation. The Subsystem - Level Crossing is in the state OPERATIONAL.</p> <p>Interaction 2.5.7.A:</p> <ol style="list-style-type: none"> - The Subsystem - Level Crossing detects a request for a deactivation from the Local operator for activating the Level Crossing protection facility. - The Subsystem - Level Crossing reports to the Subsystem - Electronic Interlocking that a local operator has requested a local deactivation. <p>Postcondition: ---</p> 	<p>Note: If the conditions for a local operation are fulfilled the Main Success Scenario: Deactivation LC SD 2.2.1 will be started.</p>	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

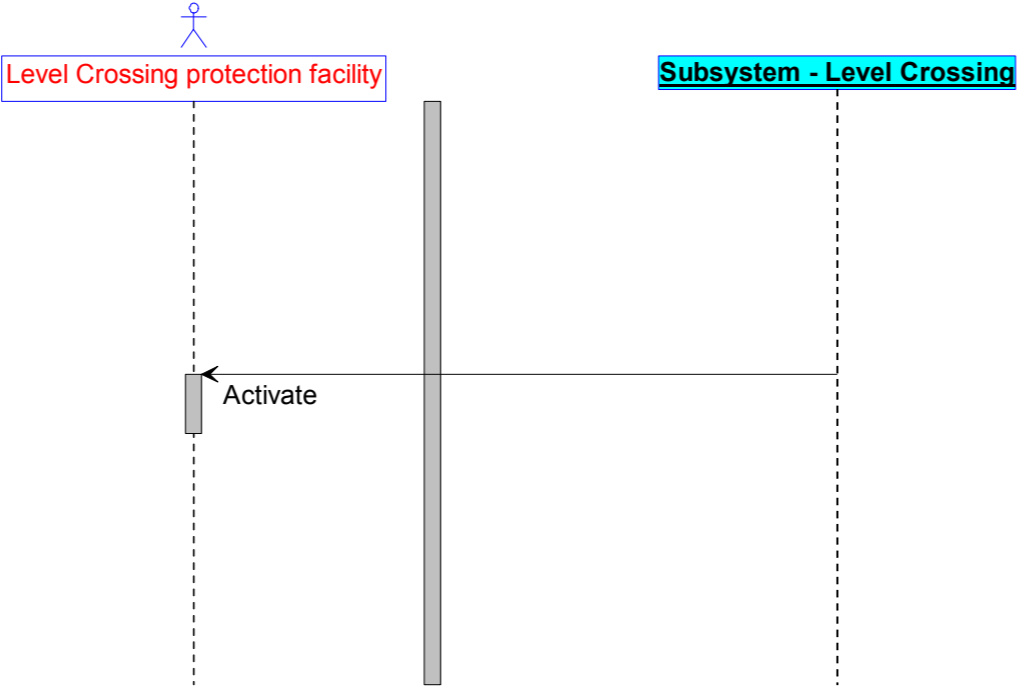
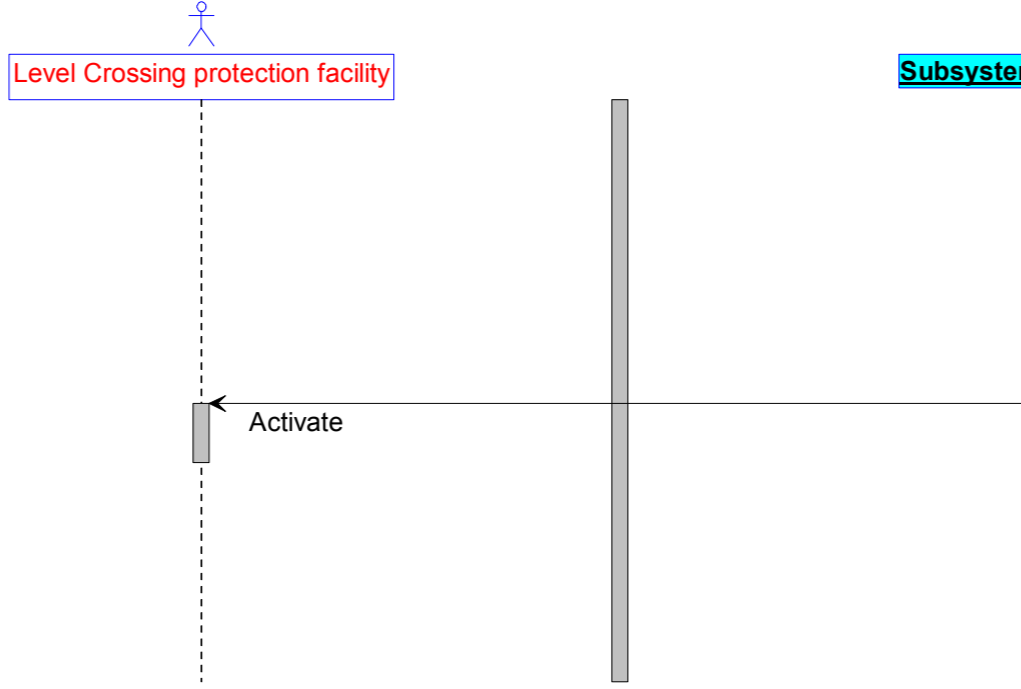
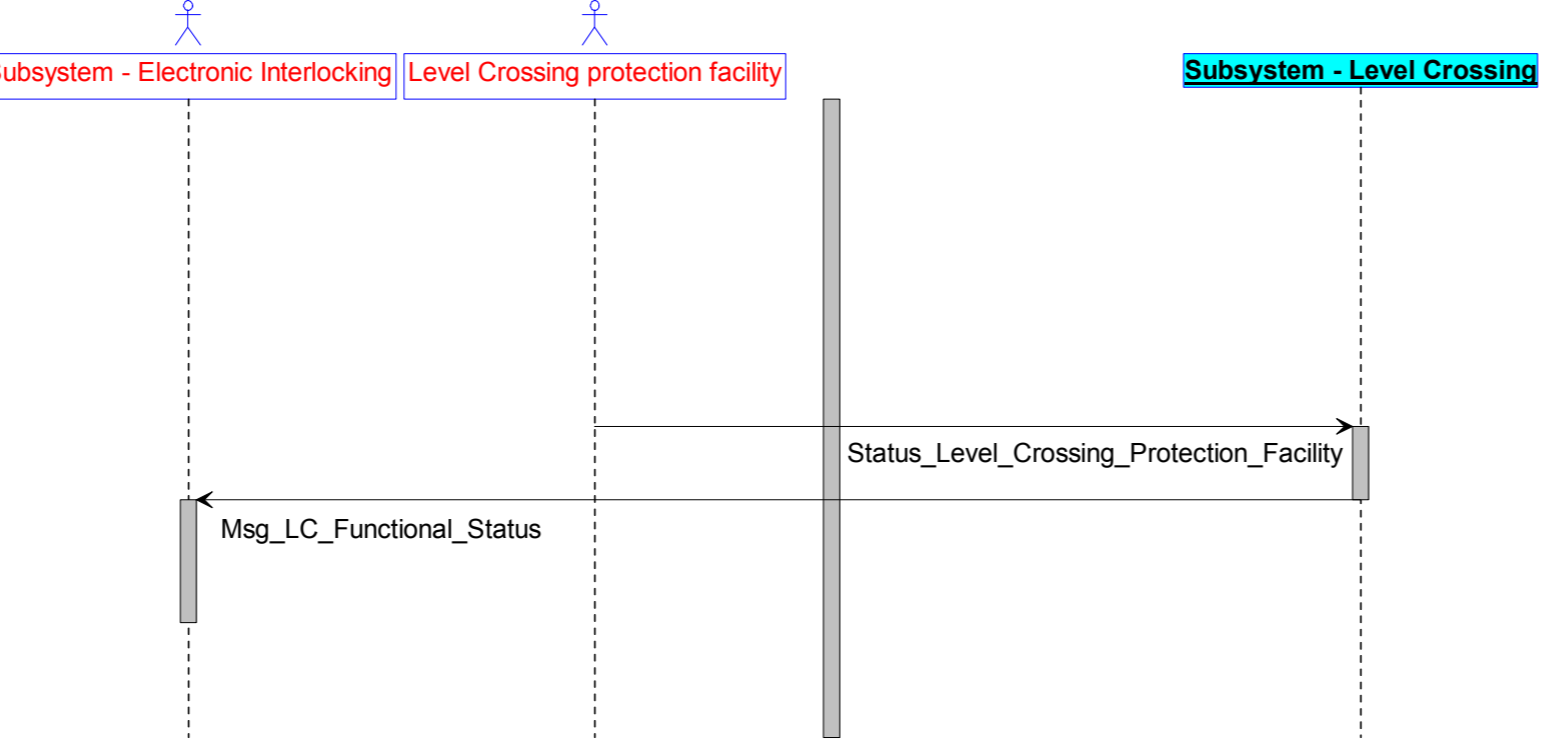
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.964	Info	<p>LC SD 2.6.1</p> <p>LC UC2.6: Handle irregularities</p> <p>Alternative Scenario: Closure time value exceeded [LC SD 2.6.1]</p> <p>Precondition: The Subsystem - Level Crossing is configured to use the Closure Timer. The Subsystem - Level Crossing is in the state OPERATIONAL. The Subsystem - Level Crossing is in the state ACTIVATED AND UNPROTECTED or ACTIVATED AND PROTECTED.</p> <p>Interaction 2.6.1.A:</p> <ol style="list-style-type: none"> - The Subsystem - Level Crossing detects that the time period Con_tmax_Closure has exceeded. 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. <p>alt [The Subsystem - Level Crossing is configured to report a critical failure after a Closure time value has exceeded]</p> <ol style="list-style-type: none"> 3.a1 The Subsystem - Level Crossing reports to the Subsystem - Electronic Interlocking that a critical failure occurred. <p>else alt [The Subsystem - Level Crossing is configured to report a non-critical failure after a Closure time value has exceeded]</p> <ol style="list-style-type: none"> 3.a2 The Subsystem - Level Crossing reports to the Subsystem - Electronic Interlocking that a non-critical failure occurred. <p>end alt</p> <p>Postcondition: —</p>		Basic LC
Eu.LC.978	Info	<p>LC SD 2.6.2</p> <p>LC UC2.6: Handle irregularities</p> <p>Alternative Scenario: Report an occurred failure [LC SD 2.6.2]</p> <p>Precondition: The Subsystem - Level Crossing is in the state OPERATIONAL.</p> <p>Interaction 2.6.2.A:</p> <p>alt [There is a failure inside the Subsystem - Level Crossing]</p> <ol style="list-style-type: none"> 1.a1 - The Subsystem - Level Crossing detects a failure. <p>else alt [There is a failure inside the Level Crossing protection facility]</p> <ol style="list-style-type: none"> 1.b1 - The Level Crossing protection facility detects a failure and reports it to the Subsystem - Level Crossing. <p>end alt</p> <p>alt [The Subsystem - Level Crossing is configured to report a critical failure after a failure was detected]</p> <ol style="list-style-type: none"> 2.a1 The Subsystem - Level Crossing reports to the Subsystem - Electronic Interlocking that a critical failure occurred. <p>else alt [The Subsystem - Level Crossing is configured to report a non-critical failure after a failure was detected]</p> <ol style="list-style-type: none"> 2.b1 The Subsystem - Level Crossing reports to the Subsystem - Electronic Interlocking that a non-critical failure occurred. <p>end alt</p> <p>Postcondition: —</p>		Basic LC

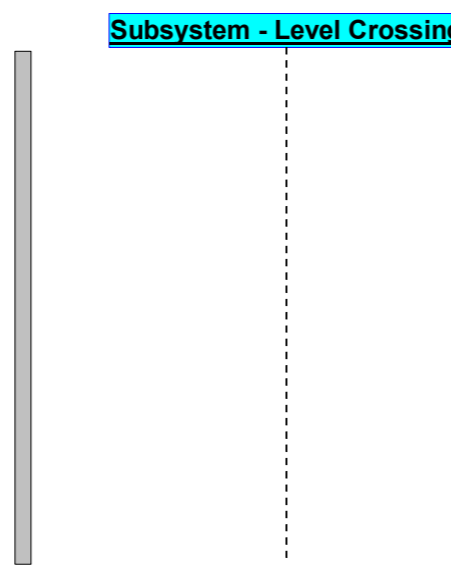
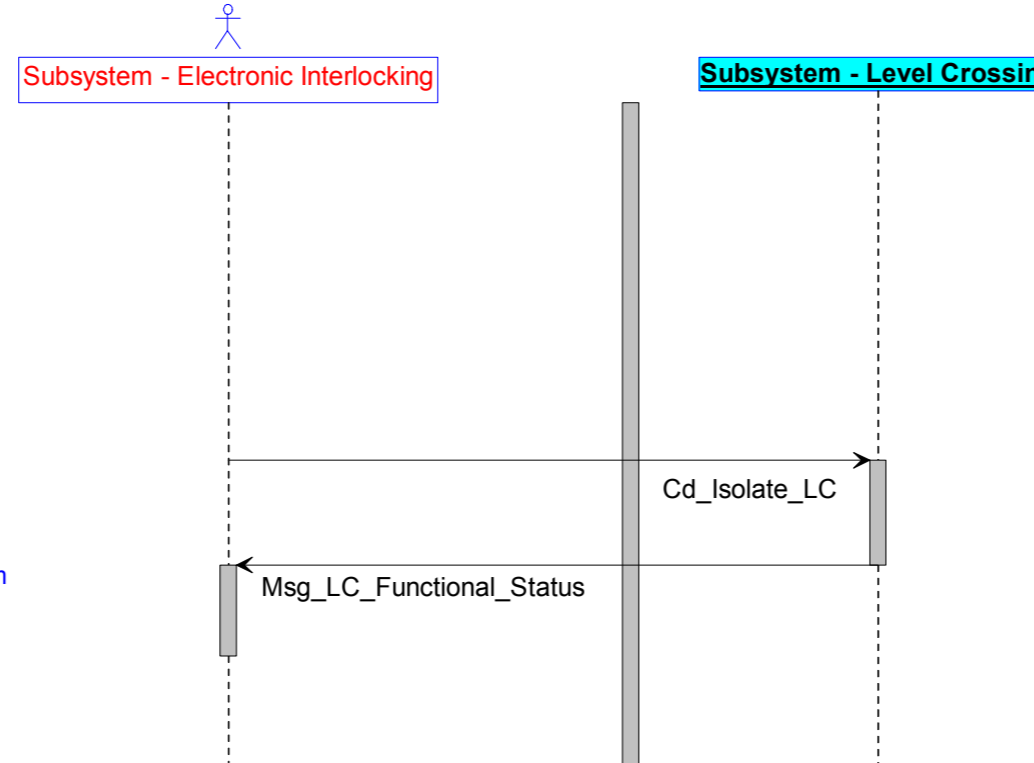
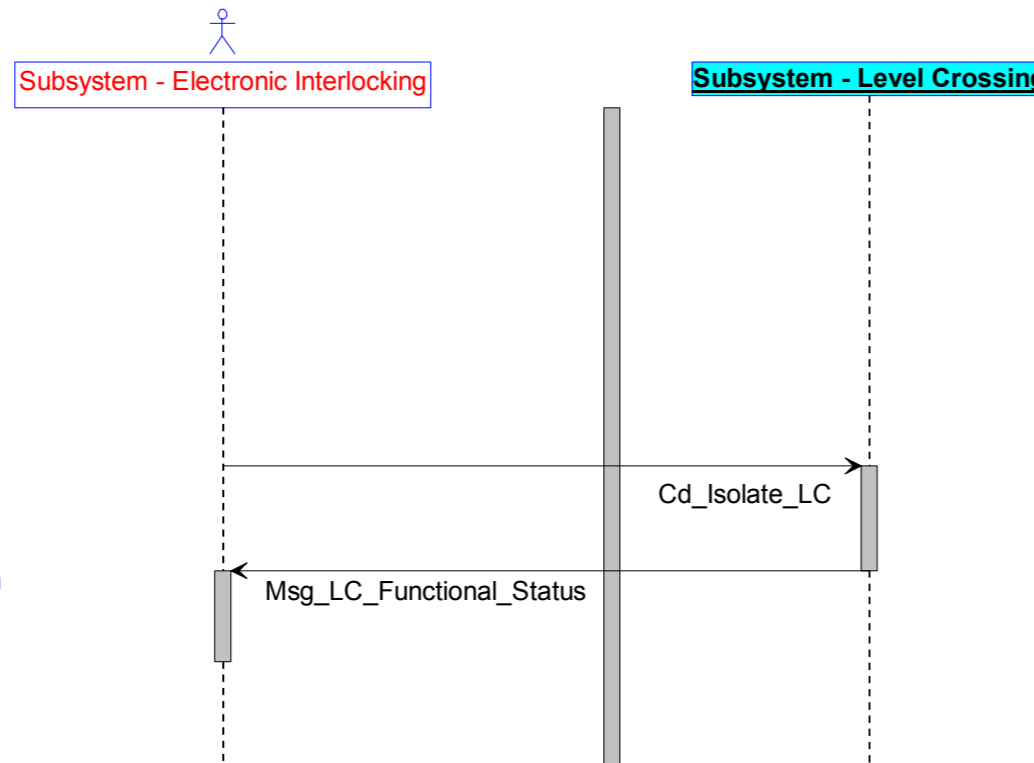


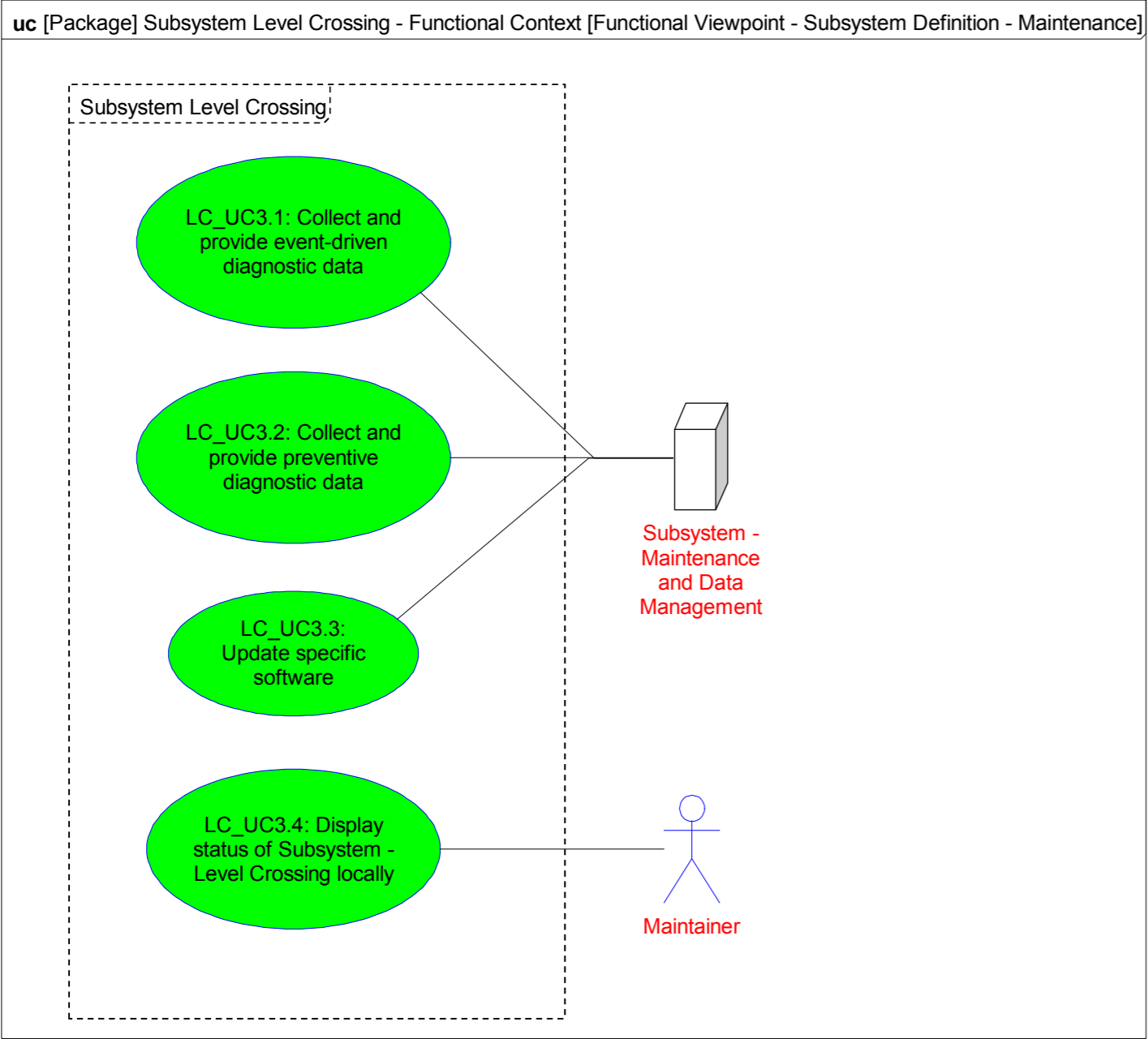
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.1000	Info	<p>LC SD 2.6.3</p> <p>LC UC2.6: Handle irregularities</p> <p>Alternative Scenario: Report removed all failures [LC SD 2.6.3]</p> <p>Precondition: The Subsystem - Level Crossing is in the state OPERATIONAL.</p> <p>Interaction 2.6.3.A:</p> <p>alt [There was a failure inside the Subsystem - Level Crossing]</p> <p>1.a1 - The Subsystem - Level Crossing detects that all occurred failures are removed.</p> <p>else alt [There was a failure inside the Level Crossing protection facility]</p> <p>1.b1 - The Level Crossing protection facility detects that all occurred failures are removed and reports it to the Subsystem - Level Crossing.</p> <p>end alt</p> <p>2. The Subsystem - Level Crossing reports to the Subsystem - Electronic Interlocking that no failures are present.</p> <p>Postcondition: —</p>		Basic LC
Eu.LC.1013	Info	<p>LC SD 2.6.4</p> <p>LC UC2.6: Handle irregularities</p> <p>Alternative Scenario: Perform fallback operation [LC SD 2.6.4]</p> <p>Precondition: —</p> <p>Interaction 2.6.4.A:</p> <p>1. - The Subsystem - Level Crossing enters the state FALLBACK_MODE.</p> <p>2. The Subsystem - Level Crossing requests the Level Crossing protection facility to change to the most safe national specific state.</p> <p>Postcondition: The Subsystem - Level Crossing is in the state FALLBACK_MODE. The Subsystem - Level Crossing is in the state according to the national requirements.</p>		Basic LC



ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.1023	Info	<p>LC SD 2.6.5</p> <p>LC UC2.6: Handle irregularities</p> <p>Alternative Scenario: Handle an interruption of the PDI connection (case 1) [LC SD 2.6.5]</p> <p>Precondition: The Subsystem - Level Crossing is in the state OPERATIONAL. 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 <u>not</u> in the state ISOLATED LC.</p> <p>Interaction 2.6.5.A:</p> <ol style="list-style-type: none"> The PDI connection has been terminated. <p>alt [The Subsystem - Level Crossing is in state DEACTIVATED AND UNPROTECTED or PRE-ACTIVATED]</p> <p>par</p> <ol style="list-style-type: none"> 2.a1.a1 The Subsystem - Level Crossing activates the Level Crossing protection facility. <p>also par</p> <ol style="list-style-type: none"> 2.a1.a2 If the Subsystem - Level Crossing is configured to deactivate the Level Crossing protection facility after a configured timer when the PDI connection is interrupted, the Subsystem - Level Crossing starts to monitor the time period Con_t_PDI_Loss_Deactivation. <p>also par</p> <ol style="list-style-type: none"> 2.a1.a3 If the Subsystem - Level Crossing is configured to use the Closure Timer the Subsystem - Level Crossing starts to monitor the time period Con_tmax_Closure. <p>end par</p> <p>else alt [The Subsystem - Level Crossing is in state ACTIVATED AND PROTECTED or ACTIVATED AND UNPROTECTED]</p> <ol style="list-style-type: none"> 2.b1 If the Subsystem - Level Crossing is configured to deactivate the Level Crossing protection facility after a configured timer when the PDI connection is interrupted, the Subsystem - Level Crossing starts to monitor the time period Con_t_PDI_Loss_Deactivation. <p>end alt</p> <p>Interaction 2.6.5.B:</p> <ol style="list-style-type: none"> The Subsystem - Level Crossing detects that Con_t_PDI_Loss_Deactivation has expired. The Subsystem - Level Crossing deactivates the Level Crossing protection facility. <p>Postcondition: The Subsystem - Level Crossing is in the state INITIALISING. The PDI connection is terminated. The Subsystem - Level Crossing is in state DEACTIVATED AND UNPROTECTED.</p>	<p>If the PDI connection is re-established before the expiration of Con_t_PDI_Loss_Deactivation_Timer, the timer is stopped.</p>	Basic LC
Eu.LC.1045	Info	<p>LC SD 2.6.6</p> <p>LC UC2.6: Handle irregularities</p> <p>Alternative Scenario: Reset occurs (case 1) [LC SD 2.6.6]</p> <p>Precondition: The Subsystem - Level Crossing is in the state OPERATIONAL.</p> <p>Interaction 2.6.6.A:</p> <ol style="list-style-type: none"> A reset has occurred. The Subsystem - Level Crossing activates the Level Crossing protection facility. If the Subsystem - Level Crossing is configured to use the Closure Timer the Subsystem - Level Crossing starts to monitor the time period Con_tmax_Closure. <p>Postcondition: The Subsystem - Level Crossing is in the state BOOTING. The Subsystem - Level Crossing is in the state ACTIVATED AND UNPROTECTED.</p>		Basic LC

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.1060	Info	<p>LC SD 2.6.7</p> <p>LC UC2.6: Handle irregularities</p> <p>Alternative Scenario: Reset occurs (case 2) [LC SD 2.6.7]</p> <p>Precondition: The Subsystem - Level Crossing is in the state INITIALISING or FALLBACK_MODE.</p> <p>Interaction 2.6.7.A:</p> <ol style="list-style-type: none"> - A reset has occurred. The Subsystem - Level Crossing activates the Level Crossing protection facility. If the Subsystem - Level Crossing is configured to use the Closure Timer the Subsystem - Level Crossing starts to monitor the time period Con_tmax_Closure. <p>Postcondition: The Subsystem - Level Crossing is in the state BOOTING. The Subsystem - Level Crossing is in the state ACTIVATED AND UNPROTECTED.</p>  <pre> sequenceDiagram actor Actor participant LCP as Level Crossing protection facility participant SLC as Subsystem - Level Crossing Actor->>LCP: Activate LCP->>SLC: Activate </pre>		Basic LC
Eu.LC.1073	Info	<p>LC SD 2.6.8</p> <p>LC UC2.6: Handle irregularities</p> <p>Alternative Scenario: Supply voltage of the Subsystem has gone outside the required range for operation [LC SD 2.6.8]</p> <p>Precondition: —</p> <p>Interaction 2.6.8.A:</p> <ol style="list-style-type: none"> - The Subsystem - Level Crossing enters the state NO_OPERATING_VOLTAGE. The Subsystem - Level Crossing activates the Level Crossing protection facility. If the Subsystem - Level Crossing is configured to use the Closure Timer the Subsystem - Level Crossing starts to monitor the time period Con_tmax_Closure. <p>Postcondition: The Subsystem - Level Crossing is in the state ACTIVATED AND UNPROTECTED.</p>  <pre> sequenceDiagram actor Actor participant LCP as Level Crossing protection facility participant SLC as Subsystem - Level Crossing Actor->>LCP: Activate LCP->>SLC: Activate </pre>		Basic LC
Eu.LC.1126	Info	<p>LC SD 2.6.9</p> <p>LC UC2.6: Handle irregularities</p> <p>Alternative Scenario: Report changed status of protection of an activated LC [LC SD 2.6.9]</p> <p>Precondition: The Subsystem - Level Crossing is in the state OPERATIONAL. The Subsystem - Level Crossing is in the state ACTIVATED AND PROTECTED.</p> <p>Interaction 2.6.9.A:</p> <ol style="list-style-type: none"> - The Level Crossing protection facility reports the new status unprotected to the Subsystem - Level Crossing. The Subsystem - Level Crossing reports to the Subsystem - Electronic Interlocking that the functional status has been changed to ACTIVATED AND UNPROTECTED. <p>Postcondition: The Subsystem - Level Crossing is in the state ACTIVATED AND UNPROTECTED.</p>  <pre> sequenceDiagram actor Actor1 actor Actor2 participant SEI as Subsystem - Electronic Interlocking participant LCP as Level Crossing protection facility participant SLC as Subsystem - Level Crossing LCP->>SLC: Status_Level_Crossing_Protection_Facility SLC->>SEI: Msg_LC_Functional_Status </pre>		Basic LC

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.1140	Info	<p>LC SD 2.6.10 LC UC2.6: Handle irregularities</p> <p>Alternative Scenario: Handle an interruption of the Safe communication protocol connection (case 4) [LC SD 2.6.10]</p> <p>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.</p> <p>Interaction 2.6.10.A: 1. - The PDI connection has been terminated. 2. The Subsystem - Level Crossing stays in the state ISOLATED LC.</p> <p>Postcondition: The Subsystem - Level Crossing is in the state INITIALISING. The PDI connection is terminated. The Subsystem - Level Crossing is in state ISOLATED LC.</p> 		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
Eu.LC.1165	Info	<p>LC 2.7.1 LC UC2.7: Handle isolate LC</p> <p>Alternative Scenario: Isolate LC [LC 2.7.1]</p> <p>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 UNPROTECTED.</p> <p>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.</p> <p>Postcondition: The Subsystem - Level Crossing is in the state ISOLATED LC.</p> 		Basic LC
Eu.LC.1174	Info	<p>LC 2.7.2 LC UC2.7: Handle isolate LC</p> <p>Alternative Scenario: Not isolate LC [LC 2.7.2]</p> <p>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.</p> <p>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.</p> <p>Postcondition: The Subsystem - Level Crossing is in the state DEACTIVATED AND UNPROTECTED.</p> 		Basic LC

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.1249	Info	<p>[Package] Subsystem Level Crossing - Functional Context [UseCase Definition Maintenance]</p> <p>uc [Package] Subsystem Level Crossing - Functional Context [Functional Viewpoint - Subsystem Definition - Maintenance]</p> 		Basic LC
Eu.LC.1250	Info	LC_UC3.1: Collect and provide event-driven diagnostic data	The Subsystem-UseCase LC_UC3.1: 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_UC3.2: Collect and provide preventive diagnostic data	The Subsystem-UseCase LC_UC3.2: 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_UC3.3: Update specific software	The Subsystem-UseCase LC_UC3.3: 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_UC3.4: Display status of Subsystem - Level Crossing locally	The LC_UC3.4: Display status of Subsystem - Level Crossing locally defines the local display of the EULYNX field element Subsystem.	Basic LC
Eu.LC.3222	Head	3.3.3 Subsystem Level Crossing - Functional Partitioning		

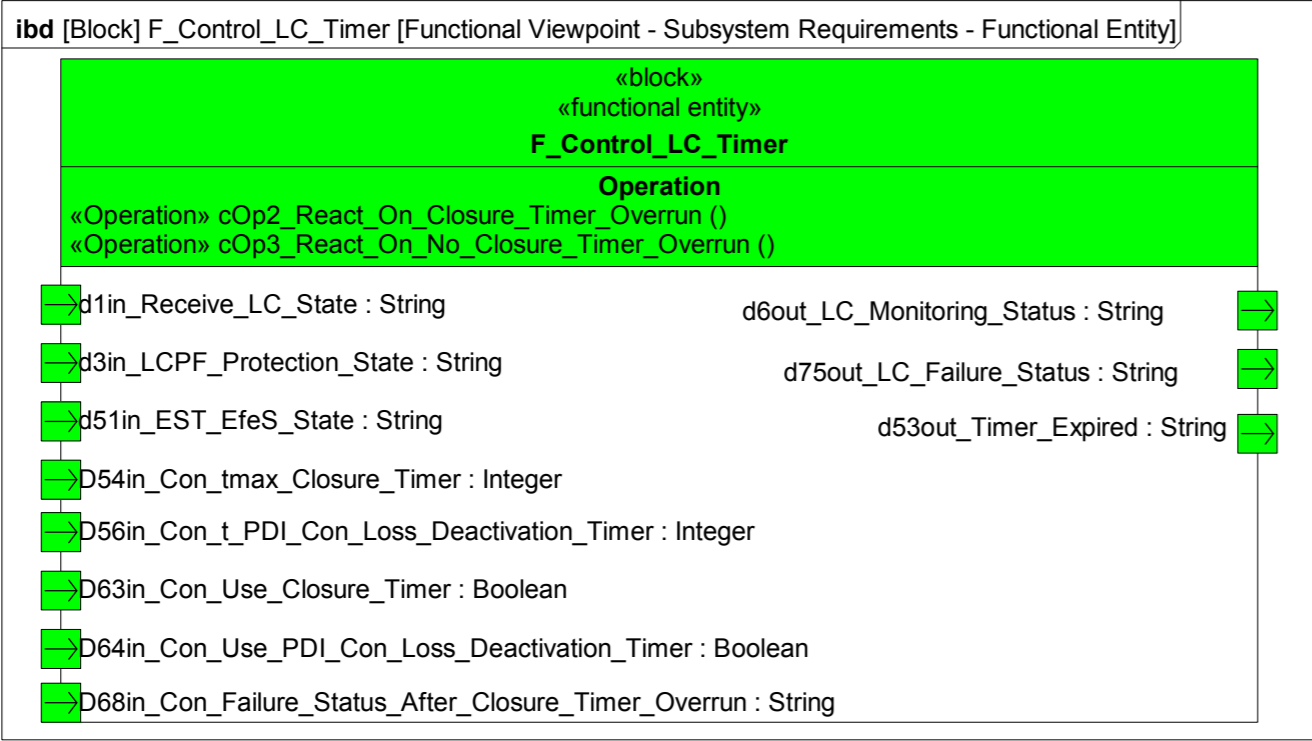
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.1985	Info	<p>[Package] Subsystem Level Crossing - Functional Entities [Functional Viewpoint - Subsystem Requirements - Functional Partitioning]</p> <p>bdd [Package] Subsystem Level Crossing - Functional Partitioning [Functional Viewpoint - Subsystem Requirements]</p>		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	Info	<p>[Block] Subsystem Level Crossing [Functional Viewpoint - Subsystem Requirements - Functional Architecture]</p> <p>ibd [Block] Subsystem Level Crossing [Functional Viewpoint - Subsystem Requirements - Functional Architecture]</p> <p>Legend:</p> <ul style="list-style-type: none"> LC1 : Basic_Data_Identifier LC2 : Maintainer SDI-LC : Subsystem_MDM_D SMI-LC : Subsystem_MDM_M SSI-LC : Subsystem_SSP 		Basic LC
Eu.LC.3086	Info	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	Info	LC1	The functional System Data interface to the Basic Data identifier. The InformationFlow through the interface is defined by "Basic_Data_Identifier".	Basic LC

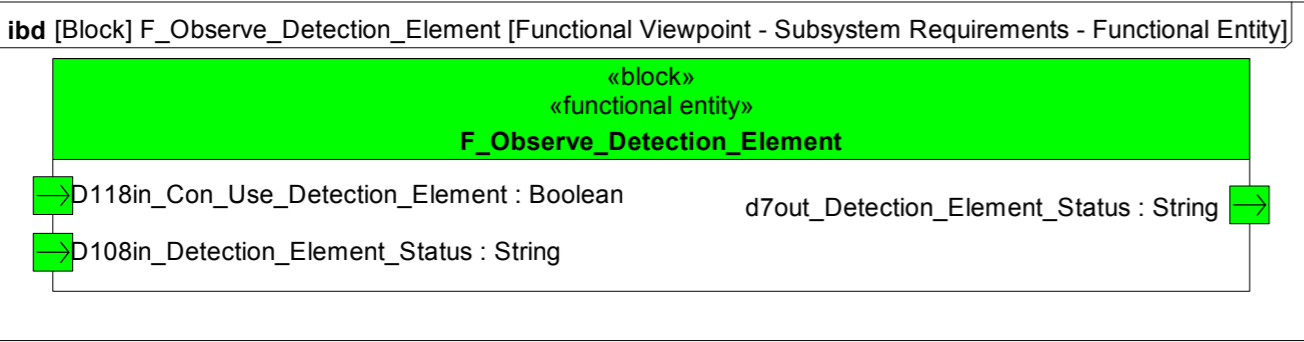
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.3211	Info	LC2	The functional Maintenance/Operation/Display interface to the Maintainer. The InformationFlow through the interface is defined by "Maintainer".	Basic LC
Eu.LC.76	Info	SDI-LC	The functional Diagnostic interface to the Subsystem - Maintenance and Data Management for the InformationFlow through the interface, which is defined by "Subsystem_MDM_D".	Basic LC
Eu.LC.75	Info	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	Info	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	Info	LC5	The functional Control interface to the Detection element. The InformationFlow through the interface is defined by "Detection_element".	Basic LC
Eu.LC.82	Info	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	Info	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	Info	[Block] F_Control_Local_Operation [Ports - LC IBD 0] <pre> ibid [Block] F_Control_Local_Operation [Functional Viewpoint - Subsystem Requirements - Functional Entity] classDiagram class F_Control_Local_Operation { <<block>> <<functional entity>> d8in_Handover_To_Local_Operator_State : String D81in_Local_Operation_Handover : String D82in_Request_by_Local_Operator : String D83in_Con_Use_Local_Operation : Boolean D80out_Handover_To_local_Operation_State : String d81out_Local_Operation_Handover : String d82out_Local_Operation_Handover : String } </pre>	Option LOH	
Eu.LC.2380	Info	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	Info	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	Info	d81out_Local_Operation_Handover		Option LOH
Eu.LC.2383	Info	D82in_Request_by_Local_Operator	The port D82in_Request_by_Local_Operator refines the Informationflow Activate and Deactivate.	Option LOH
Eu.LC.2384	Info	d82out_Local_Operation_Handover		Option LOH
Eu.LC.2385	Info	d8in_Handover_To_Local_Operator_State		Option LOH
Eu.LC.3799	Info	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	Info	<p>Functional Viewpoint - Subsystem Requirements - Functional Entity STD 0</p> <pre> stm [State Machine] F_Control_Local_Operation - Behaviour [Functional Viewpoint - Subsystem Requirements - Functional Entity STD 0] Initial0 CONTROL_LOCAL_OPERATION COMMAND_HANOVER_TO_LOCAL_OPERATION Initial1 WAITING NO_HANOVER Entry/D80out_Handover_To_local_Operation_State := "No handover to local operator"; HANOVER_ESTABLISHED Entry/D80out_Handover_To_local_Operation_State := "Handover to local operator established"; HANOVER_INITIATED Entry/D80out_Handover_To_local_Operation_State := "Handover to local operator initiated"; RECEIVE_INPUT_FOR_LOCAL_OPERATION_BY_LOCAL_OPERATOR Initial2 WAITING ALLOW Entry/d81out_Local_Operation_Handover := "Allow handover to local operator"; RETURN Entry/d81out_Local_Operation_Handover := "Return handover from local operator"; REQUEST_BY_LOCAL_OPERATOR Initial3 WAITING REQUEST_TO_ACTIVATE Entry/d82out_Local_Operation_Handover := "Request to activate the level Crossing"; REQUEST_TO_DEACTIVATE Entry/d82out_Local_Operation_Handover := "Request to deactivate the level Crossing"; </pre>		Option LOH
Eu.LC.2388	Info	CONTROL_LOCAL_OPERATION		Option LOH
Eu.LC.2389	Info	COMMAND_HANOVER_TO_LOCAL_OPERATION		Option LOH
Eu.LC.2390	Info	HANOVER_ESTABLISHED		Option LOH
Eu.LC.2391	Req	when(d8in_Handover_To_Local_Operator_State = "No Handover" And D83in_Con_Use_Local_Operation){HANOVER_ESTABLISHED - NO_HANOVER}		Option LOH
Eu.LC.2392	Req	when(d8in_Handover_To_Local_Operator_State = "Initiated" And D83in_Con_Use_Local_Operation){HANOVER_ESTABLISHED - HANOVER_INITIATED}		Option LOH
Eu.LC.3000	Req	entry/D80out_Handover_To_local_Operation_State := "Handover to local operator established";{State-internal in HANOVER_ESTABLISHED}		Option LOH
Eu.LC.2393	Info	HANOVER_INITIATED		Option LOH
Eu.LC.2394	Req	when(d8in_Handover_To_Local_Operator_State = "Established" And D83in_Con_Use_Local_Operation){HANOVER_INITIATED - HANOVER_ESTABLISHED}		Option LOH
Eu.LC.2395	Req	when(d8in_Handover_To_Local_Operator_State = "No Handover" And D83in_Con_Use_Local_Operation){HANOVER_INITIATED - NO_HANOVER}		Option LOH

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.3001	Req	entry/D80out_Handover_To_local_Operation_State := "Handover to local operator initiated";{State-internal in HANDOVER_INITIATED}		Option LOH
Eu.LC.2396	Info	Initial1		Option LOH
Eu.LC.2397	Req	/{Initial1 - WAITING}		Option LOH
Eu.LC.2398	Info	NO_HANDOVER		Option LOH
Eu.LC.2399	Req	when(d8in_Handover_To_Local_Operator_State = "Initiated" And D83in_Con_Use_Local_Operation){NO_HANDOVER - HANDOVER_INITIATED}		Option LOH
Eu.LC.2400	Req	when(d8in_Handover_To_Local_Operator_State = "Established" And D83in_Con_Use_Local_Operation){NO_HANDOVER - HANDOVER_ESTABLISHED}		Option LOH
Eu.LC.3002	Req	entry/D80out_Handover_To_local_Operation_State := "No handover to local operator";{State-internal in NO_HANDOVER}		Option LOH
Eu.LC.2401	Info	WAITING		Option LOH
Eu.LC.2402	Req	when(d8in_Handover_To_Local_Operator_State = "Established" And D83in_Con_Use_Local_Operation){WAITING - HANDOVER_ESTABLISHED}		Option LOH
Eu.LC.2403	Req	when(d8in_Handover_To_Local_Operator_State = "Initiated" And D83in_Con_Use_Local_Operation){WAITING - HANDOVER_INITIATED}		Option LOH
Eu.LC.2404	Req	when(d8in_Handover_To_Local_Operator_State = "No Handover" And D83in_Con_Use_Local_Operation){WAITING - NO_HANDOVER}		Option LOH
Eu.LC.2405	Info	RECEIVE_INPUT_FOR_LOCAL_OPERATION_BY_LOCAL_OPERATOR		Option LOH
Eu.LC.2406	Info	ALLOW		Option LOH
Eu.LC.2407	Req	when(D81in_Local_Operation_Handover = "Return handover from local operator" And D83in_Con_Use_Local_Operation){ALLOW - RETURN}		Option LOH
Eu.LC.3003	Req	entry/d81out_Local_Operation_Handover := "Allow handover to local operator";{State-internal in ALLOW}		Option LOH
Eu.LC.2408	Info	Initial2		Option LOH
Eu.LC.2409	Req	/{Initial2 - WAITING}		Option LOH
Eu.LC.2410	Info	RETURN		Option LOH
Eu.LC.2411	Req	when(D81in_Local_Operation_Handover = "Allow handover to local operator" And D83in_Con_Use_Local_Operation){RETURN - ALLOW}		Option LOH
Eu.LC.3004	Req	entry/d81out_Local_Operation_Handover := "Return handover from local operator";{State-internal in RETURN}		Option LOH
Eu.LC.2412	Info	WAITING		Option LOH
Eu.LC.2413	Req	when(D81in_Local_Operation_Handover = "Allow handover to local operator" And D83in_Con_Use_Local_Operation){WAITING - ALLOW}		Option LOH
Eu.LC.2414	Req	when(D81in_Local_Operation_Handover = "Return handover from local operator" And D83in_Con_Use_Local_Operation){WAITING - RETURN}		Option LOH
Eu.LC.2415	Info	REQUEST_BY_LOCAL_OPERATOR		Option LOH
Eu.LC.2416	Info	Initial3		Option LOH
Eu.LC.2417	Req	/{Initial3 - WAITING}		Option LOH
Eu.LC.2418	Info	REQUEST_TO_ACTIVATE		Option LOH
Eu.LC.2419	Req	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
Eu.LC.3005	Req	entry/d82out_Local_Operation_Handover := "Request to activate the level Crossing";{State-internal in REQUEST_TO_ACTIVATE}		Option LOH
Eu.LC.2420	Info	REQUEST_TO_DEACTIVATE		Option LOH
Eu.LC.2421	Req	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

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.3006	Req	entry/d82out_Local_Operation_Handover := "Request to deactivate the level Crossing";{State-internal in REQUEST_TO_DEACTIVATE}		Option LOH
Eu.LC.2422	Info	WAITING		Option LOH
Eu.LC.2423	Req	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	Req	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	Info	Initial0		Option LOH
Eu.LC.2426	Req	/{Initial0 - CONTROL_LOCAL_OPERATION}		Option LOH
Eu.LC.2427	Info	F_Control_LC_Timer		Basic LC
Eu.LC.2428	Info	[Block] F_Control_Timer [Ports - LC IBD 1] 		Basic LC
Eu.LC.2429	Info	cOp2_React_On_Closure_Timer_Overrun	if D68in_Con_Failure_Status_After_Closure_Timer_Overrun = "non critical failure report" then d75out_LC_Failure_Status := "A non critical failure is present"; elseif D68in_Con_Failure_Status_After_Closure_Timer_Overrun = "critical failure report" then d75out_LC_Failure_Status := "A critical failure is present"; end if	Basic LC
Eu.LC.2430	Info	cOp3_React_On_No_Closure_Timer_Overrun	if D68in_Con_Failure_Status_After_Closure_Timer_Overrun = "non critical failure report" then d75out_LC_Failure_Status := "No non critical failure present"; elseif D68in_Con_Failure_Status_After_Closure_Timer_Overrun = "critical failure report" then d75out_LC_Failure_Status := "No critical failure present"; end if	Basic LC
Eu.LC.2431	Info	d1in_Receive_LC_State		Basic LC
Eu.LC.2432	Info	d3in_LCPF_Protection_State		Basic LC
Eu.LC.2433	Info	d51in_EST_EfeS_State		Basic LC
Eu.LC.2434	Info	D54in_Con_tmax_Closure_Timer	The port D54in_Con_tmax_Closure_Timer refines the time value for Con_tmax_Closure. The following values are permitted: - 1 up to any number	Basic LC
Eu.LC.2435	Info	d53out_Timer_Expired		Basic LC
Eu.LC.2436	Info	D56in_Con_t_PDI_Con_Loss_Deactivation_Timer	The port D56in_Con_t_PDI_Con_Loss_Deactivation_Timer refines the time value for Con_t_PDI_Loss_Deactivation. The following values are permitted: - 1 up to any number	Basic LC
Eu.LC.2437	Info	D63in_Con_Use_Closure_Timer	The port D63in_Con_Use_Closure_Timer provides configuration values for the Con_tmax_Closure. true: Con_tmax_Closure is used false: Con_tmax_Closure is not used	Basic LC

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.2438	Info	D64in_Con_Use_PDI_Con_Loss_Deactivation_Timer	The port D64in_Con_Use_PDI_Con_Loss_Deactivation_Timer provides configuration values for the Con_t_PDI_Loss_Deactivation. true: Con_t_PDI_Loss_Deactivation is used false: Con_t_PDI_Loss_Deactivation is not used	Basic LC
Eu.LC.2439	Info	D68in_Con_Failure_Status_After_Closure_Timer_Overrun	The port D68in_Con_Failure_Status_After_Closure_Timer_Overrun provides the configuration value what failure status the Subsystem - Level Crossing is configured to report after a closure timer overrun occurred.	Basic LC
Eu.LC.2440	Info	d6out_LC_Monitoring_Status		Basic LC
Eu.LC.2441	Info	d75out_LC_Failure_Status		Basic LC
Eu.LC.2442	Info	F_Control_LC_Timer - Behaviour		Basic LC
Eu.LC.2443	Info	Functional Viewpoint - Subsystem Requirements - Functional Entity STD 1 <pre> strn [State Machine] F_Control_LC_Timer - Behaviour [Functional Viewpoint - Subsystem Requirements - Functional Entity STD 1] </pre>		Basic LC
Eu.LC.2444	Info	FALLBACKMODE		Basic LC
Eu.LC.2445	Info	Initial0		Basic LC
Eu.LC.2446	Req	/{Initial0 - OBSERVE_TIMER}		Basic LC
Eu.LC.2447	Info	OBSERVE_TIMER		Basic LC
Eu.LC.2448	Info	MONITOR_CLOSURE_TIMER		Basic LC
Eu.LC.2449	Info	CLOSURE_TIMER_EXPIRED		Basic LC
Eu.LC.2450	Req	when(d3in_LCPF_Protection_State = "Idle" OR d51in_EST_EfeS_State = "FALLBACK_MODE")/{CLOSURE_TIMER_EXPIRED - IDLE}		Basic LC
Eu.LC.3214	Req	entry/d6out_LC_Monitoring_Status := "Closure timer overrun occurred";{State-internal in CLOSURE_TIMER_EXPIRED}		Basic LC
Eu.LC.2451	Info	CLOSURE_TIMER_IS_RUNNING		Basic LC
Eu.LC.2452	Req	after(D54in_Con_tmax_Closure_Timer)/cOp2_React_On_Closure_Timer_Overrun();{CLOSURE_TIMER_IS_RUNNING - CLOSURE_TIMER_EXPIRED}		Basic LC
Eu.LC.2453	Req	when(d3in_LCPF_Protection_State = "Idle" OR d51in_EST_EfeS_State = "FALLBACK_MODE")/{CLOSURE_TIMER_IS_RUNNING - IDLE}		Basic LC
Eu.LC.2454	Info	IDLE		Basic LC
Eu.LC.2455	Req	when(d1in_Receive_LC_State = "Activated")[D63in_Con_Use_Closure_Timer = TRUE]/{IDLE - CLOSURE_TIMER_IS_RUNNING}		Basic LC
Eu.LC.3215	Req	entry/d6out_LC_Monitoring_Status := "No Closure timer overrun"; cOp3_React_On_No_Closure_Timer_Overrun();{State-internal in IDLE}		Basic LC

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.2456	Info	Initial1		Basic LC
Eu.LC.2457	Req	/{Initial1 - IDLE}		Basic LC
Eu.LC.2458	Req	PDI_CON_LOSS_TIMER		Basic LC
Eu.LC.2459	Info	IDLE		Basic LC
Eu.LC.3216	Req	when(d51in_EST_EfeS_State = "OPERATIONAL")/{IDLE - WAITING}		
Eu.LC.2461	Info	Initial2		Basic LC
Eu.LC.2462	Req	/{Initial2 - IDLE}		Basic LC
Eu.LC.2463	Info	TIMER_EXPIRED		Basic LC
Eu.LC.2464	Req	when(d51in_EST_EfeS_State = "OPERATIONAL")/d53out_Timer_Expired := "Timer not Expired";{TIMER_EXPIRED - WAITING}		Basic LC
Eu.LC.2465	Info	TIMER_RUNNING		Basic LC
Eu.LC.2466	Req	after(D56in_Con_t_PDI_Con_Loss_Deactivation_Timer)/ d53out_Timer_Expired := "Timer Expired";{TIMER_RUNNING - TIMER_EXPIRED}		Basic LC
Eu.LC.3217	Req	when(d51in_EST_EfeS_State = "OPERATIONAL" OR d51in_EST_EfeS_State = "BOOTING" OR d51in_EST_EfeS_State = "NO_OPERATING_VOLTAGE")/{TIMER_RUNNING - WAITING}		Basic LC
Eu.LC.3218	Info	WAITING		Basic LC
Eu.LC.2460	Req	when(d51in_EST_EfeS_State = "INITIALISING")[(NOT d1in_Receive_LC_State = "Isolated" AND D64in_Con_Use_PDI_Con_Loss_Deactivation_Timer = TRUE)]/{WAITING - TIMER_RUNNING}		Basic LC
Eu.LC.2467	Info	TECHNICAL_DISTURBANCE		Basic LC
Eu.LC.2468	Info	WAITING_FOR_FINISH_BOOTING		Basic LC
Eu.LC.2469	Info	F_Observe_Detection_Element		Basic LC
Eu.LC.2470	Info	[Block] F_Observe_Detection_Element [Ports - LC IBD 2] 		Basic LC
Eu.LC.2471	Info	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	Info	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	Info	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	Info	<p>Functional Viewpoint - Subsystem Requirements - Functional Entity STD 2</p> <pre> stm [State Machine] F_Observe_Detection_Element - Behaviour [Functional Viewpoint - Subsystem Requirements - Functional Entity STD 2] Initial0 state WAITING state VACANT state OCCUPIED state TECHNICAL_FAILURE when(D108in_Detection_Element_Status = "Detection element vacant" AND D118in_Con_Use_Detection_Element) / Initial0 --> WAITING when(D108in_Detection_Element_Status = "Detection element failed" AND D118in_Con_Use_Detection_Element) / Initial0 --> WAITING when(D108in_Detection_Element_Status = "Detection element occupied" AND D118in_Con_Use_Detection_Element) / WAITING --> OCCUPIED when(D108in_Detection_Element_Status = "Detection element occupied" AND D118in_Con_Use_Detection_Element) / WAITING --> TECHNICAL_FAILURE when(D108in_Detection_Element_Status = "Detection element failed" AND D118in_Con_Use_Detection_Element) / OCCUPIED --> TECHNICAL_FAILURE when(D108in_Detection_Element_Status = "Detection element failed" AND D118in_Con_Use_Detection_Element) / TECHNICAL_FAILURE --> VACANT when(D108in_Detection_Element_Status = "Detection element vacant" AND D118in_Con_Use_Detection_Element) / TECHNICAL_FAILURE --> VACANT when(D108in_Detection_Element_Status = "Detection element failed" AND D118in_Con_Use_Detection_Element) / VACANT --> TECHNICAL_FAILURE VACANT: Entry/d7out_Detection_Element_Status := "Vacant"; OCCUPIED: Entry/d7out_Detection_Element_Status := "Occupied"; TECHNICAL_FAILURE: Entry/d7out_Detection_Element_Status := "Failed"; </pre>		Basic LC
Eu.LC.2476	Info	Initial0		Basic LC
Eu.LC.2477	Req	/{Initial0 - WAITING}		Basic LC
Eu.LC.2478	Info	OCCUPIED		Basic LC
Eu.LC.2479	Req	when(D108in_Detection_Element_Status = "Detection element vacant" AND D118in_Con_Use_Detection_Element){OCCUPIED - VACANT}		Basic LC
Eu.LC.2480	Req	when(D108in_Detection_Element_Status = "Detection element failed" AND D118in_Con_Use_Detection_Element){OCCUPIED - TECHNICAL_FAILURE}		Basic LC
Eu.LC.3009	Req	entry/d7out_Detection_Element_Status := "Occupied";{State-internal in OCCUPIED}		Basic LC
Eu.LC.2481	Req	TECHNICAL_FAILURE		Basic LC
Eu.LC.2482	Req	when(D108in_Detection_Element_Status = "Detection element occupied" AND D118in_Con_Use_Detection_Element){TECHNICAL_FAILURE - OCCUPIED}		Basic LC
Eu.LC.2483	Req	when(D108in_Detection_Element_Status = "Detection element vacant" AND D118in_Con_Use_Detection_Element){TECHNICAL_FAILURE - VACANT}		Basic LC
Eu.LC.3010	Req	entry/d7out_Detection_Element_Status := "Failed";{State-internal in TECHNICAL_FAILURE}		Basic LC
Eu.LC.2484	Info	VACANT		Basic LC
Eu.LC.2485	Req	when(D108in_Detection_Element_Status = "Detection element occupied" AND D118in_Con_Use_Detection_Element){VACANT - OCCUPIED}		Basic LC
Eu.LC.2486	Req	when(D108in_Detection_Element_Status = "Detection element failed" AND D118in_Con_Use_Detection_Element){VACANT - TECHNICAL_FAILURE}		Basic LC
Eu.LC.3011	Req	entry/d7out_Detection_Element_Status := "Vacant";{State-internal in VACANT}		Basic LC
Eu.LC.2487	Info	WAITING		Basic LC
Eu.LC.2488	Req	when(D108in_Detection_Element_Status = "Detection element occupied" AND D118in_Con_Use_Detection_Element){WAITING - OCCUPIED}		Basic LC
Eu.LC.2489	Req	when(D108in_Detection_Element_Status = "Detection element vacant" AND D118in_Con_Use_Detection_Element){WAITING - VACANT}		Basic LC
Eu.LC.2490	Req	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	Info	<p>[Block] F_Observe_Failure_State [Ports - LC IBD 3]</p> <div style="border: 1px solid black; padding: 5px;"> <p>ibd [Block] F_Observe_Failure_State [Functional Viewpoint - Subsystem Requirements - Functional Entity]</p> <div style="background-color: #00FF00; padding: 2px; text-align: center;"> <p>«block» «functional entity» F_Observe_Failure_State</p> </div> <p> → d74in_LCPF_Failure_State : String d61out_LC_Failure_Status : String → → D60in_LC_Failure : Boolean d62out_LCPF_Failure_Status : String → → D67in_Con_LCPF_Failure_Status_After_Present_Failure : String → D69in_Con_LC_Failure_Status_After_Present_Failure : String </p> </div>		Basic LC
Eu.LC.2494	Info	d61out_LC_Failure_Status		Basic LC
Eu.LC.2495	Info	d62out_LCPF_Failure_Status		Basic LC
Eu.LC.2496	Info	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	Info	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	Info	d74in_LCPF_Failure_State		Basic LC
Eu.LC.3272	Info	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.2499	Info	F_Observe_Failure_State - Behaviour		Basic LC

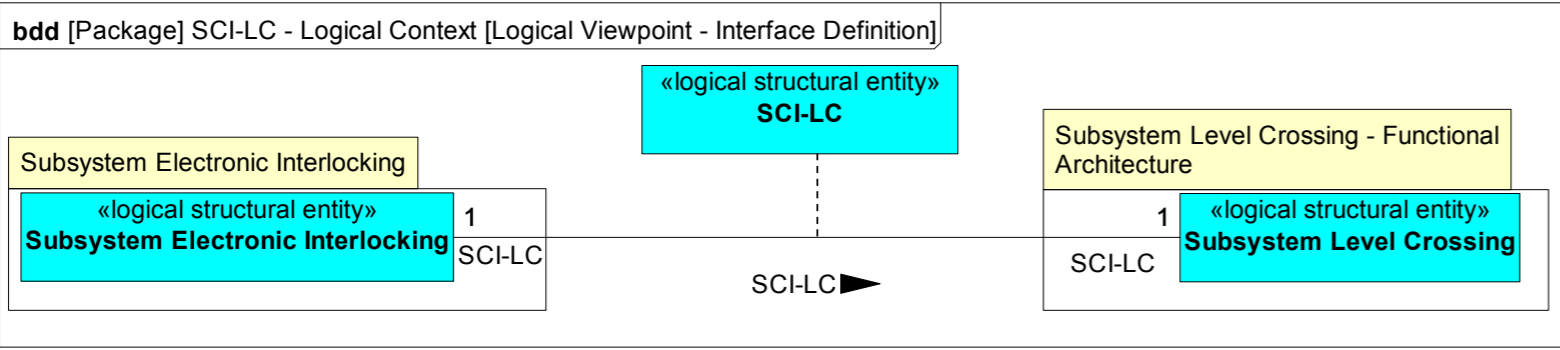
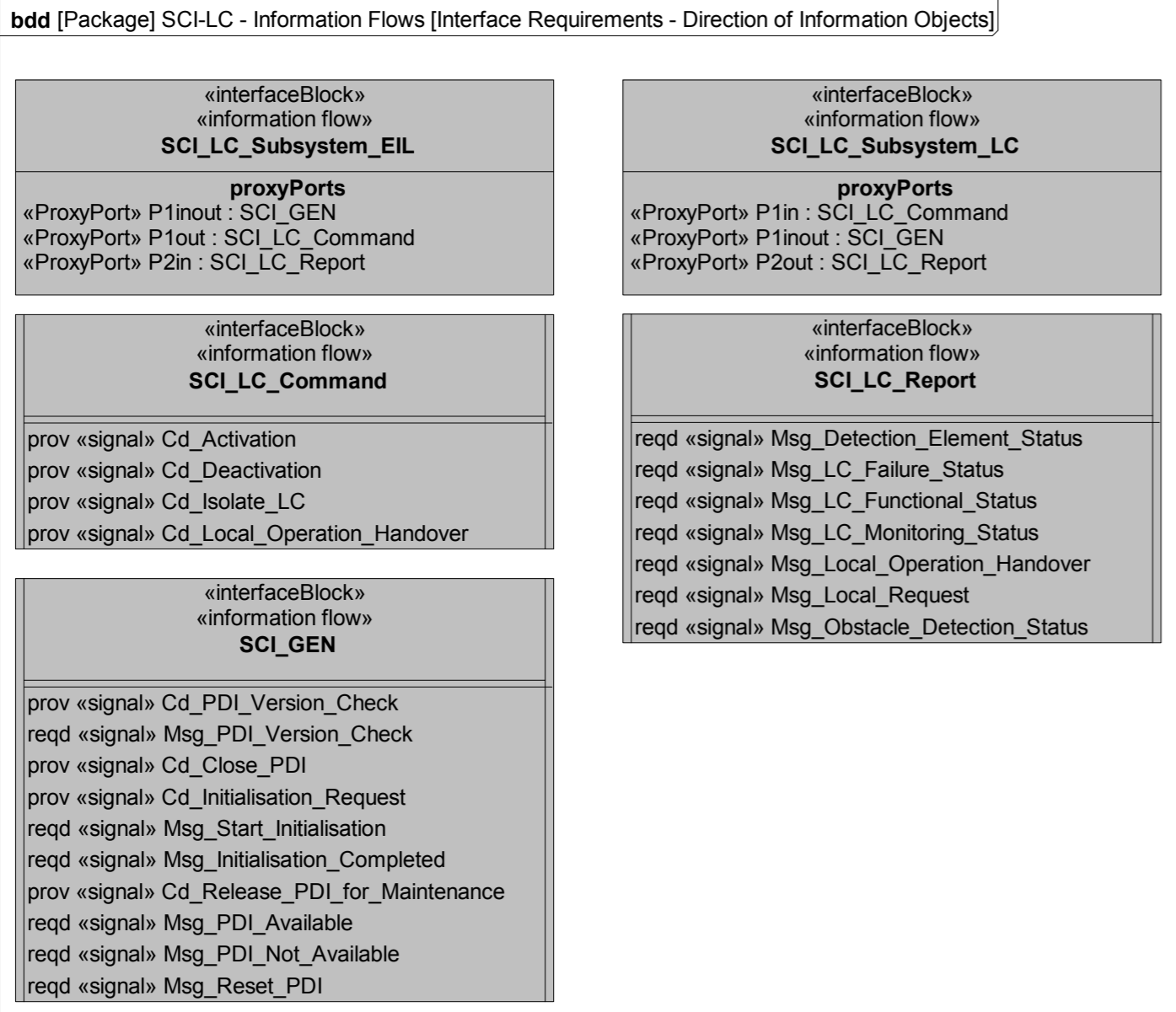
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.2500	Info	<p>Functional Viewpoint - Subsystem Requirements - Functional Entity STD 3</p> <pre> stm [State Machine] F_Observe_Failure_State - Behaviour [Functional Viewpoint - Subsystem Requirements - Functional Entity STD 3] ● Initial0 → WAITING_FOR_BOOTING when(d51out_EST_EfeS_State = "BOOTING") / → OBSERVE_FAILURE_STATES when(d51out_EST_EfeS_State = "FALLBACK_MODE") / → WAITING_FOR_BOOTING LC_FAILURE_STATE ● Initial1 [D60in_LC_Failure AND D69in_Con_LC_Failure_Status_After_Present_Failure = "non critical failure report"] / → Junction0 [else] / → Junction0 [D60in_LC_Failure AND D69in_Con_LC_Failure_Status_After_Present_Failure = "critical failure report"] / → Junction0 [D69in_Con_LC_Failure_Status_After_Present_Failure = "critical failure report" AND NOT D60in_LC_Failure] / → Junction0 Junction0 when(D60in_LC_Failure) [D69in_Con_LC_Failure_Status_After_Present_Failure = "critical failure report"] / → CRITICAL_FAILURE_DETECTED when(NOT D60in_LC_Failure) [D69in_Con_LC_Failure_Status_After_Present_Failure = "critical failure report"] / → NO_CRITICAL_FAILURE_DETECTED CRITICAL_FAILURE_DETECTED Entry/d61out_LC_Failure_Status := "A critical failure is present"; when(NOT D60in_LC_Failure) [D69in_Con_LC_Failure_Status_After_Present_Failure = "critical failure report"] / → NO_CRITICAL_FAILURE_DETECTED NO_CRITICAL_FAILURE_DETECTED Entry/d61out_LC_Failure_Status := "No critical failure present"; when(D60in_LC_Failure) [D69in_Con_LC_Failure_Status_After_Present_Failure = "non critical failure report"] / → NON_CRITICAL_FAILURE_DETECTED when(NOT D60in_LC_Failure) [D69in_Con_LC_Failure_Status_After_Present_Failure = "non critical failure report"] / → NO_NON_CRITICAL_FAILURE_DETECTED NON_CRITICAL_FAILURE_DETECTED Entry/d61out_LC_Failure_Status := "A non critical failure is present"; when(NOT D60in_LC_Failure) [D69in_Con_LC_Failure_Status_After_Present_Failure = "non critical failure report"] / → NO_NON_CRITICAL_FAILURE_DETECTED NO_NON_CRITICAL_FAILURE_DETECTED Entry/d61out_LC_Failure_Status := "No non critical failure present"; LCPF_FAILURE_STATE ● Initial2 [d74in_LCPF_Failure_State = "Failure present" AND D67in_Con_LCPF_Failure_Status_After_Present_Failure = "non critical failure report"] / → Junction1 [else] / → Junction1 [d74in_LCPF_Failure_State = "Failure present" AND D67in_Con_LCPF_Failure_Status_After_Present_Failure = "critical failure report"] / → Junction1 [d74in_LCPF_Failure_State = "No failure present" AND D67in_Con_LCPF_Failure_Status_After_Present_Failure = "critical failure report"] / → Junction1 Junction1 when(d74in_LCPF_Failure_State = "Failure present") [D67in_Con_LCPF_Failure_Status_After_Present_Failure = "critical failure report"] / → CRITICAL_FAILURE_DETECTED when(d74in_LCPF_Failure_State = "No failure present") [D67in_Con_LCPF_Failure_Status_After_Present_Failure = "critical failure report"] / → NO_CRITICAL_FAILURE_DETECTED CRITICAL_FAILURE_DETECTED Entry/d62out_LCPF_Failure_Status := "A critical failure is present"; when(d74in_LCPF_Failure_State = "No failure present") [D67in_Con_LCPF_Failure_Status_After_Present_Failure = "critical failure report"] / → NO_CRITICAL_FAILURE_DETECTED NO_CRITICAL_FAILURE_DETECTED Entry/d62out_LCPF_Failure_Status := "No critical failure present"; when(d74in_LCPF_Failure_State = "Failure present") [D67in_Con_LCPF_Failure_Status_After_Present_Failure = "non critical failure report"] / → NON_CRITICAL_FAILURE_DETECTED when(d74in_LCPF_Failure_State = "No failure present") [D67in_Con_LCPF_Failure_Status_After_Present_Failure = "non critical failure report"] / → NO_NON_CRITICAL_FAILURE_DETECTED NON_CRITICAL_FAILURE_DETECTED Entry/d62out_LCPF_Failure_Status := "A non critical failure is present"; when(d74in_LCPF_Failure_State = "No failure present") [D67in_Con_LCPF_Failure_Status_After_Present_Failure = "non critical failure report"] / → NO_NON_CRITICAL_FAILURE_DETECTED NO_NON_CRITICAL_FAILURE_DETECTED Entry/d62out_LCPF_Failure_Status := "No non critical failure present"; </pre>		Basic LC
Eu.LC.2501	Info	Initial0		Basic LC
Eu.LC.2502	Req	/{Initial0 - WAITING_FOR_BOOTING}		Basic LC
Eu.LC.2503	Info	OBSERVE_FAILURE_STATES		Basic LC
Eu.LC.2504	Info	LC_FAILURE_STATE		Basic LC
Eu.LC.2505	Info	CRITICAL_FAILURE_DETECTED		Basic LC
Eu.LC.2506	Req	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	Req	entry/d61out_LC_Failure_Status := "A critical failure is present";{State-internal in CRITICAL_FAILURE_DETECTED}		Basic LC
Eu.LC.2507	Info	Initial1		Basic LC

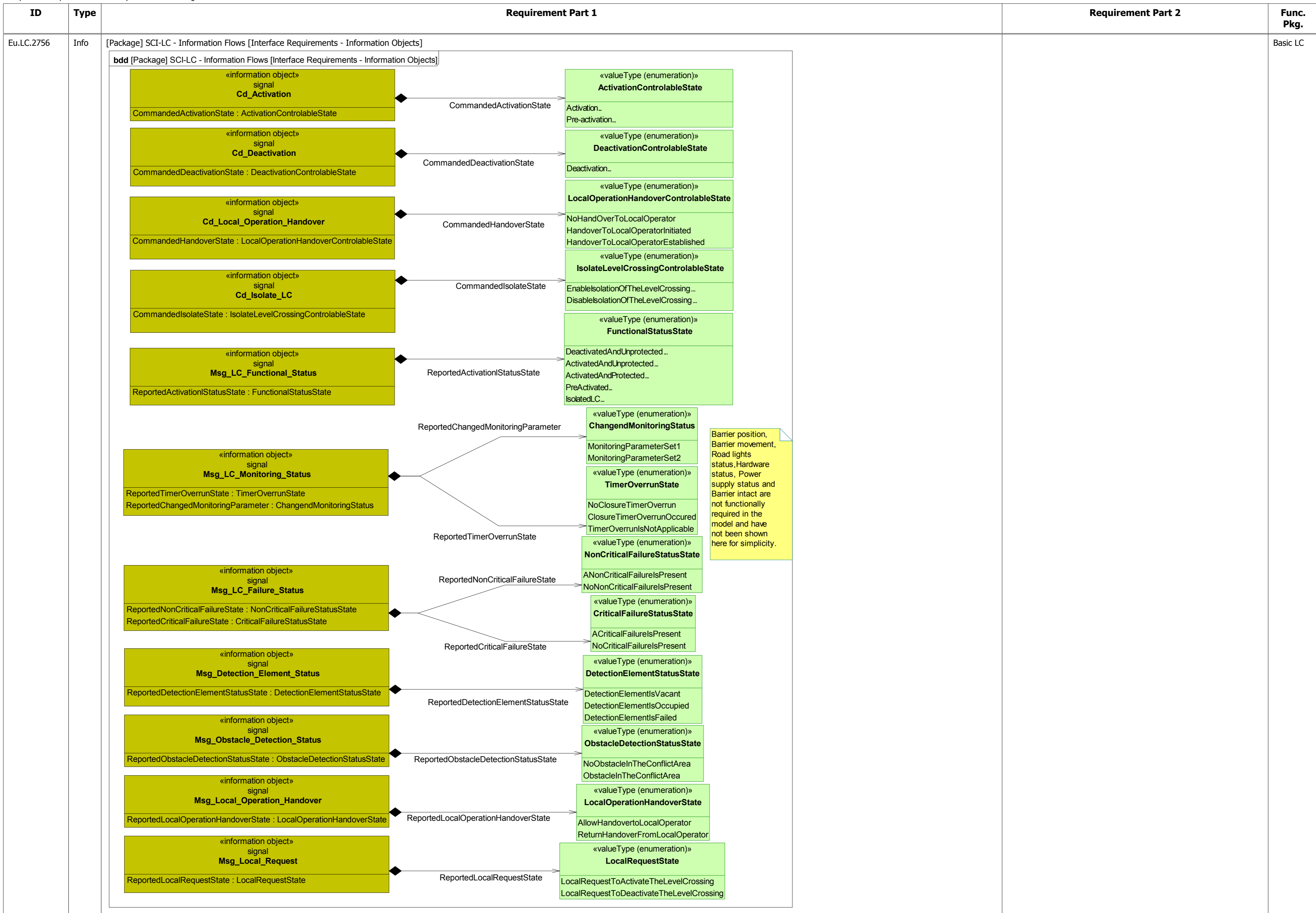
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.2508	Req	/{Initial1 - Junction0}		Basic LC
Eu.LC.2509	Info	NO_CRITICAL_FAILURE_DETECTED		Basic LC
Eu.LC.2510	Req	when(D60in_LC_Failure)[D69in_Con_LC_Failure_Status_After_Present_Failure = "critical failure report"]/{NO_CRITICAL_FAILURE_DETECTED - CRITICAL_FAILURE_DETECTED}		Basic LC
Eu.LC.3013	Req	entry/d61out_LC_Failure_Status := "No critical failure present";{State-internal in NO_CRITICAL_FAILURE_DETECTED}		Basic LC
Eu.LC.3800	Info	NO_NON_CRITICAL_FAILURE_DETECTED		Basic LC
Eu.LC.3801	Req	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		entry/d61out_LC_Failure_Status := "No non critical failure present";{State-internal in NO_NON_CRITICAL_FAILURE_DETECTED}		Basic LC
Eu.LC.3802	Req	NON_CRITICAL_FAILURE_DETECTED		Basic LC
Eu.LC.3803	Req	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		entry/d61out_LC_Failure_Status := "A non critical failure is present";{State-internal in NON_CRITICAL_FAILURE_DETECTED}		Basic LC
Eu.LC.3831		Junction0		Basic LC
Eu.LC.3832		[D60in_LC_Failure AND D69in_Con_LC_Failure_Status_After_Present_Failure = "critical failure report"]/{Junction0 - CRITICAL_FAILURE_DETECTED}		Basic LC
Eu.LC.2519	Req	[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	Req	[else]/{Junction0 - NO_NON_CRITICAL_FAILURE_DETECTED}		Basic LC
Eu.LC.3805	Req	[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	Info	LCPF_FAILURE_STATE		Basic LC
Eu.LC.2521	Info	CRITICAL_FAILURE_DETECTED		Basic LC
Eu.LC.2522	Req	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	Req	entry/d62out_LCPF_Failure_Status := "A critical failure is present";{State-internal in CRITICAL_FAILURE_DETECTED}		Basic LC
Eu.LC.2523	Info	Initial2		Basic LC
Eu.LC.3274	Req	/{Initial2 - Junction1}		Basic LC
Eu.LC.2525	Info	NO_CRITICAL_FAILURE_DETECTED		Basic LC
Eu.LC.2526	Req	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	Req	entry/d62out_LCPF_Failure_Status := "No critical failure present";{State-internal in NO_CRITICAL_FAILURE_DETECTED}		Basic LC
Eu.LC.3808	Info	NO_NON_CRITICAL_FAILURE_DETECTED		Basic LC
Eu.LC.3809	Req	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		entry/d62out_LCPF_Failure_Status := "No non critical failure present";{State-internal in NO_NON_CRITICAL_FAILURE_DETECTED}		Basic LC
Eu.LC.3810	Info	NON_CRITICAL_FAILURE_DETECTED		Basic LC
Eu.LC.3811	Req	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		entry/d62out_LCPF_Failure_Status := "A non critical failure is present";{State-internal in NON_CRITICAL_FAILURE_DETECTED}		Basic LC
Eu.LC.3835		Junction1		Basic LC
Eu.LC.3814	Req	[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	Req	[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	Req	[else]/{Junction1 - NO_NON_CRITICAL_FAILURE_DETECTED}		Basic LC
Eu.LC.3813	Req	[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		when(d51out_EST_EfeS_State = "FALLBACK_MODE")/{OBSERVE_FAILURE_STATES - WAITING_FOR_BOOTING}		Basic LC
Eu.LC.3839		WAITING_FOR_BOOTING		Basic LC
Eu.LC.3840		when(d51out_EST_EfeS_State = "BOOTING")/{WAITING_FOR_BOOTING - OBSERVE_FAILURE_STATES}		Basic LC
Eu.LC.3830		d51out_EST_EfeS_State		Basic LC
Eu.LC.2536	Info	F_Observe_LCPF		Basic LC

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.2537	Info	[Block] F_Observe_LCPF [Ports - LC IBD 4] 		Basic LC
Eu.LC.2538	Info	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	Info	D30in_LCPF_Protection_State	The port D30in_LCPF_Protection_State refines the Informationflow Status_Level_Crossing_Protection_Facility.	Basic LC
Eu.LC.2540	Info	d3out_LCPF_Protection_State		Basic LC
Eu.LC.2541	Info	d4out_Obstacle_Detection_Status		Basic LC
Eu.LC.2542	Info	d74out_LCPF_Failure_State		Basic LC
Eu.LC.2543	Info	d9out_Changed_Monitoring_State		Basic LC
Eu.LC.3816		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		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		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

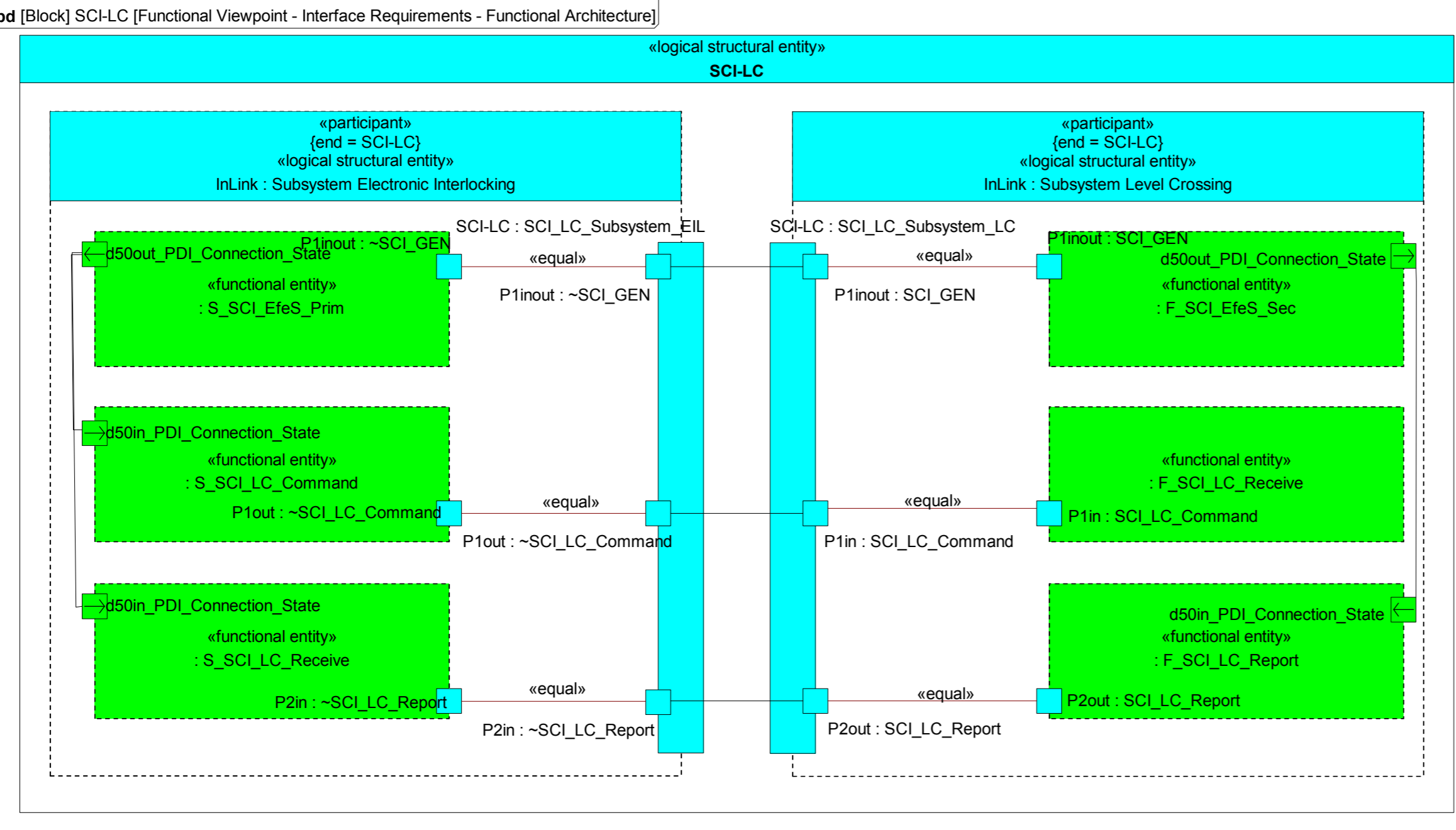
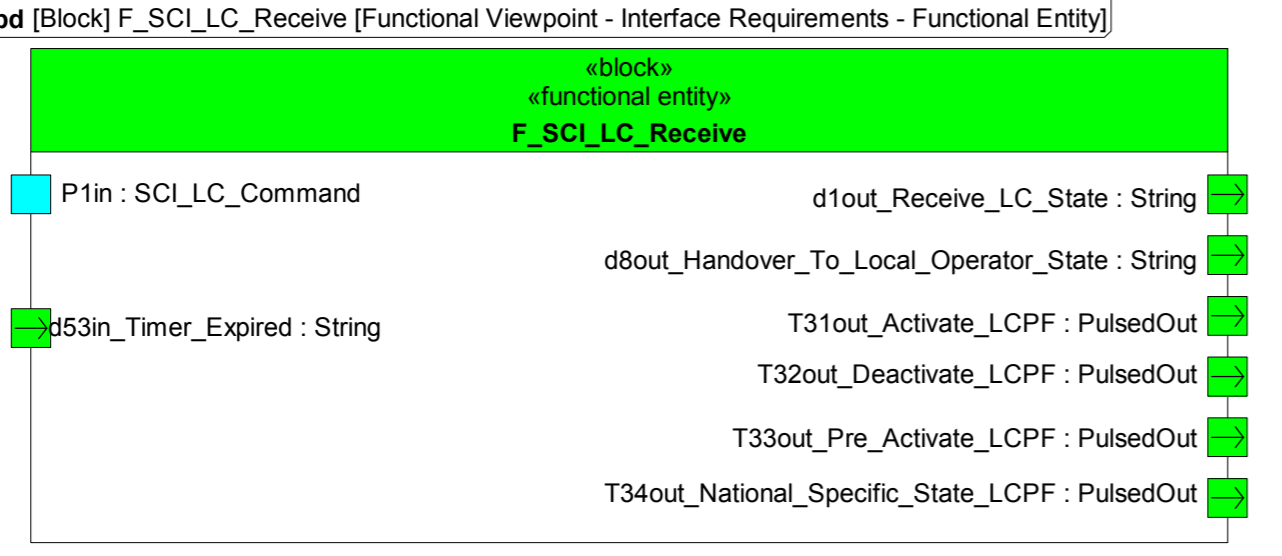
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.2545	Info	<p>Functional Viewpoint - Subsystem Requirements - Functional Entity STD 4</p> <p>stm [State Machine] F_Observe_LCPF - Behaviour [Functional Viewpoint - Subsystem Requirements - Functional Entity STD 4]</p> <p>OBSERVE_LCPF_PROTECTION_STATE</p> <ul style="list-style-type: none"> Initial1 UNPROTECTED: Entry/d3out_LCPF_Protection_State := "Unprotected"; PROTECTED: Entry/d3out_LCPF_Protection_State := "Protected"; IDLE: Entry/d3out_LCPF_Protection_State := "Idle"; <p>OBSERVE_LCPF_OBSTACLE_DETECTION_STATE</p> <ul style="list-style-type: none"> Initial2 WAITING OBSTACLE_IN_CONFLICT_AREA: Entry/d4out_Obstacle_Detection_Status := "Obstacle detected in the conflict area"; NO_OBSTACLE_IN_CONFLICT_AREA: Entry/d4out_Obstacle_Detection_Status := "No obstacle in the conflict area"; <p>OBSERVE_LCPF_FAILURE_STATE</p> <ul style="list-style-type: none"> Initial3 WAITING FAILURE_DETECTED: Entry/d74out_LCPF_Failure_State := "Failure present"; NO_FAILURE_DETECTED: Entry/d74out_LCPF_Failure_State := "No failure present"; <p>OBSERVE_MONITORING_STATUS</p> <ul style="list-style-type: none"> Initial4 MONITORING_PARAMETER_SET_1: Entry/d9out_Changed_Monitoring_State := "Monitoring Parameter Set 1"; MONITORING_PARAMETER_SET_2: Entry/d9out_Changed_Monitoring_State := "Monitoring Parameter Set 2"; 		Basic LC
Eu.LC.2546	Info	Initial0		Basic LC
Eu.LC.2547	Req	/{Initial0 - OBSERVE_LCPF}		Basic LC
Eu.LC.2548	Info	OBSERVE_LCPF		Basic LC
Eu.LC.2549	Info	OBSERVE_LCPF_FAILURE_STATE		Basic LC
Eu.LC.2550	Info	FAILURE_DETECTED		Basic LC
Eu.LC.2551	Req	when(NOT D32in_LCPF_Failure_State){FAILURE_DETECTED - NO_FAILURE_DETECTED}		Basic LC
Eu.LC.3020	Req	entry/d74out_LCPF_Failure_State := "Failure present";{State-internal in FAILURE_DETECTED}		Basic LC
Eu.LC.2552	Info	Initial3		Basic LC
Eu.LC.2553	Req	/{Initial3 - WAITING}		Basic LC
Eu.LC.2554	Info	NO_FAILURE_DETECTED		Basic LC

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

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.2998	Info	<p>[Package] SCI-LC - Logical Context [Logical Viewpoint - Interface Definition]</p> 	Process Data Interface SCI-LC	Basic LC
Eu.LC.2755	Head	3.4.1.2 SCI-LC - Information Flows		
Eu.LC.3269	Info	The generic commands and messages through the SCI_LC_Subsystem_EIL are specified in Eu.Doc.119		Basic LC
Eu.LC.2997	Info	<p>[Package] SCI-LC - Information Flows [Interface Requirements - Directions of Information Objects]</p> 		Basic LC



ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.2757	Info	Cd_Activation		Basic LC
Eu.LC.2758	Info	Cd_Deactivation		Basic LC
Eu.LC.2759	Info	Cd_Isolate_LC		Basic LC
Eu.LC.2760	Info	Cd_Local_Operation_Handover		Option LOH
Eu.LC.2761	Info	Msg_Detection_Element_Status		Basic LC
Eu.LC.2762	Info	Msg_LC_Failure_Status		Basic LC
Eu.LC.2763	Info	Msg_LC_Functional_Status		Basic LC
Eu.LC.2764	Info	Msg_LC_Monitoring_Status		Basic LC
Eu.LC.2765	Info	Msg_Local_Operation_Handover		Option LOH
Eu.LC.2766	Info	Msg_Local_Request		Option LOH
Eu.LC.2767	Info	Msg_Obstacle_Detection_Status		Basic LC
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.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	Info	<p>[Package] SCI-LC - Logical Viewpoint [Interface Requirements - Logical Partitioning]</p>	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	Info	<p>[Block] SCI-LC [Functional Viewpoint - Interface Requirements - Functional Architecture]</p> <p>ibd [Block] SCI-LC [Functional Viewpoint - Interface Requirements - Functional Architecture]</p> 		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	Info	<p>[Block] F_SCI_LC_Receive [Ports - SCI_LC IBD 2]</p> <p>ibd [Block] F_SCI_LC_Receive [Functional Viewpoint - Interface Requirements - Functional Entity]</p> 		Basic LC
Eu.LC.3282	Info	T31out_Activate_LCPF	The port T31_Activate_LCPF refines the Flow Property Activate.	Basic LC
Eu.LC.3283	Info	T32out_Deactivate_LCPF	The port T32_Deactivate_LCPF refines the Flow Property Deactivate.	Basic LC
Eu.LC.3284	Info	T33out_Pre_Activate_LCPF	The port T33_Pre_Activate_LCPF refines the Flow Property Pre-Activate.	Basic LC
Eu.LC.3285	Info	T34out_National_Specific_State_LCPF	The port T34_National_Specific_State_LCPF refines the Flow Property National_Specific_State.	Basic LC
Eu.LC.2633	Info	P1in	The port P1in exchanges information objects according to SCI_LC_Subsystem_LC.	Basic LC
Eu.LC.3276	Info	d51in_EST_EfeS_State		Basic LC
Eu.LC.2593	Info	d1out_Receive_LC_State		Basic LC
Eu.LC.2596	Info	d53in_Timer_Expired		Basic LC
Eu.LC.2597	Info	d8out_Handover_To_Local_Operator_State		Option LOH

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.2598	Info	F_SCI_LC_Receive - Behaviour		Basic LC
Eu.LC.2599	Info	<p>Functional Viewpoint - Interface Requirements - Functional Entity STD 2</p> <p>stm [State Machine] F_SCI_LC_Receive - Behaviour [Functional Viewpoint - Interface Requirements - Functional Entity STD 2]</p>		Basic LC Option LOH
Eu.LC.2600	Info	Initial0		Basic LC
Eu.LC.2601	Req	{/Initial0 - WAITING}		Basic LC
Eu.LC.2602	Info	RECEIVING_LEVEL_CROSSING_COMMANDS		Basic LC
Eu.LC.2603	Info	RECEIVE_LOCAL_OPERATION_HANDOVER		Option LOH
Eu.LC.2604	Info	HANOVER_TO_LOCAL_OPERATOR_ESTABLISHED		Option LOH
Eu.LC.2605	Req	Cd_Local_Operation_Handover[CommandedHandoverState = HandoverToLocalOperatorInitiated]/{HANOVER_TO_LOCAL_OPERATOR_ESTABLISHED - HANOVER_TO_LOCAL_OPERATOR_INITIATED}		Option LOH
Eu.LC.2606	Req	Cd_Local_Operation_Handover[CommandedHandoverState = NoHandOverToLocalOperator]/{HANOVER_TO_LOCAL_OPERATOR_ESTABLISHED - NO_HANOVER_TO_LOCAL_OPERATOR}		Option LOH
Eu.LC.3028	Req	entry/d8out_Handover_To_Local_Operator_State := "Established";{State-internal in HANOVER_TO_LOCAL_OPERATOR_ESTABLISHED}		Option LOH
Eu.LC.2607	Info	HANOVER_TO_LOCAL_OPERATOR_INITIATED		Option LOH

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.2608	Req	Cd_Local_Operation_Handover[CommandedHandoverState = HandoverToLocalOperatorEstablished]/{HANDOVER_TO_LOCAL_OPERATOR_INITIATED - HANDOVER_TO_LOCAL_OPERATOR_ESTABLISHED}		Option LOH
Eu.LC.2609	Req	Cd_Local_Operation_Handover[CommandedHandoverState = NoHandOverToLocalOperator]/{HANDOVER_TO_LOCAL_OPERATOR_INITIATED - NO_HANDOVER_TO_LOCAL_OPERATOR}		Option LOH
Eu.LC.3029	Req	entry/d8out_Handover_To_Local_Operator_State := "Initiated";{State-internal in HANDOVER_TO_LOCAL_OPERATOR_INITIATED}		Option LOH
Eu.LC.2610	Info	Initial2		Option LOH
Eu.LC.2611	Req	/{Initial2 - WAITING}		Option LOH
Eu.LC.2612	Info	NO_HANDOVER_TO_LOCAL_OPERATOR		Option LOH
Eu.LC.2613	Req	Cd_Local_Operation_Handover[CommandedHandoverState = HandoverToLocalOperatorEstablished]/{NO_HANDOVER_TO_LOCAL_OPERATOR - HANDOVER_TO_LOCAL_OPERATOR_ESTABLISHED}		Option LOH
Eu.LC.2614	Req	Cd_Local_Operation_Handover[CommandedHandoverState = HandoverToLocalOperatorInitiated]/{NO_HANDOVER_TO_LOCAL_OPERATOR - HANDOVER_TO_LOCAL_OPERATOR_INITIATED}		Option LOH
Eu.LC.3030	Req	entry/d8out_Handover_To_Local_Operator_State := "No Handover";{State-internal in NO_HANDOVER_TO_LOCAL_OPERATOR}		Option LOH
Eu.LC.2615	Info	WAITING		Option LOH
Eu.LC.2616	Req	Cd_Local_Operation_Handover[CommandedHandoverState = HandoverToLocalOperatorEstablished]/{WAITING - HANDOVER_TO_LOCAL_OPERATOR_ESTABLISHED}		Option LOH
Eu.LC.2617	Req	Cd_Local_Operation_Handover[CommandedHandoverState = HandoverToLocalOperatorInitiated]/{WAITING - HANDOVER_TO_LOCAL_OPERATOR_INITIATED}		Option LOH
Eu.LC.2618	Req	Cd_Local_Operation_Handover[CommandedHandoverState = NoHandOverToLocalOperator]/{WAITING - NO_HANDOVER_TO_LOCAL_OPERATOR}		Option LOH
Eu.LC.2619	Info	RECEIVING_ACTIVATION_STATE		Basic LC
Eu.LC.2620	Info	ACTIVATED		Basic LC
Eu.LC.2621	Req	Cd_Deactivation[CommandedDeactivationState = Deactivation]/T32out_Deactivate_LCPF := TRUE;{ACTIVATED - DEACTIVATED}		Basic LC
Eu.LC.2622	Req	when(d53in_Timer_Expired = "Timer Expired")/T32out_Deactivate_LCPF := TRUE;{ACTIVATED - DEACTIVATED}		Basic LC
Eu.LC.3031	Req	entry/d1out_Receive_LC_State := "Activated"; T31out_Activate_LCPF := TRUE;{State-internal in ACTIVATED}		Basic LC
Eu.LC.2623	Info	DEACTIVATED		Basic LC
Eu.LC.2624	Req	Cd_Activation[CommandedActivationState = Activation]/{DEACTIVATED - ACTIVATED}		Basic LC
Eu.LC.2625	Req	Cd_Activation[CommandedActivationState = Pre-activation]/{DEACTIVATED - PRE_ACTIVATED}		Basic LC
Eu.LC.2626	Req	Cd_Isolate_LC[CommandedIsolateState = EnableIsolationOfTheLevelCrossing]/{DEACTIVATED - ISOLATED}		Basic LC
Eu.LC.3032	Req	entry/d1out_Receive_LC_State := "Deactivated";{State-internal in DEACTIVATED}		Basic LC
Eu.LC.3296	Req	when(d51in_EST_EfeS_State = "INITIALISING")/{DEACTIVATED - ACTIVATED}		Basic LC
Eu.LC.2627	Info	Initial1		Basic LC
Eu.LC.2628	Req	/{Initial1 - ACTIVATED}		Basic LC
Eu.LC.2629	Info	ISOLATED		Basic LC
Eu.LC.2630	Req	Cd_Isolate_LC[CommandedIsolateState = DisableIsolationOfTheLevelCrossing]/{ISOLATED - DEACTIVATED}		Basic LC
Eu.LC.3033	Req	entry/d1out_Receive_LC_State := "Isolated";{State-internal in ISOLATED}		Basic LC
Eu.LC.2631	Info	PRE_ACTIVATED		Basic LC
Eu.LC.2632	Req	Cd_Activation[CommandedActivationState = Activation]/{PRE_ACTIVATED - ACTIVATED}		Basic LC
Eu.LC.3034	Req	entry/d1out_Receive_LC_State := "Pre Activated"; T33out_Pre_Activate_LCPF := TRUE;{State-internal in PRE_ACTIVATED}		Basic LC
Eu.LC.3224	Req	Cd_Deactivation[CommandedDeactivationState = Deactivation]/T32out_Deactivate_LCPF := TRUE;{PRE_ACTIVATED - DEACTIVATED}		Basic LC
Eu.LC.3297	Req	when(d51in_EST_EfeS_State = "INITIALISING")/{PRE_ACTIVATED - ACTIVATED}		Basic LC
Eu.LC.3280	Req	when(d51in_EST_EfeS_State = "BOOTING")/{RECEIVING_LEVEL_CROSSING_COMMANDS - RECEIVING_LEVEL_CROSSING_COMMANDS}		Basic LC
Eu.LC.3281	Req	when(d51in_EST_EfeS_State = "FALLBACK_MODE")/{RECEIVING_LEVEL_CROSSING_COMMANDS - FALLBACK_MODE}		Basic LC
Eu.LC.3300	Req	when(d51in_EST_EfeS_State = "NO_OPERATING_VOLTAGE")/{RECEIVING_LEVEL_CROSSING_COMMANDS - RECEIVING_LEVEL_CROSSING_COMMANDS}		Basic LC
Eu.LC.3277	Info	FALLBACK_MODE		Basic LC
Eu.LC.3278	Req	entry/T34out_National_Specific_State_LCPF := TRUE;{State-internal in FALLBACK_MODE}		Basic LC
Eu.LC.3279	Req	when(d51in_EST_EfeS_State = "OPERATIONAL" OR d51in_EST_EfeS_State = "BOOTING" OR d51in_EST_EfeS_State = "NO_OPERATING_VOLTAGE" OR d51in_EST_EfeS_State = "INITIALISING")/{FALLBACK_MODE - RECEIVING_LEVEL_CROSSING_COMMANDS}		Basic LC

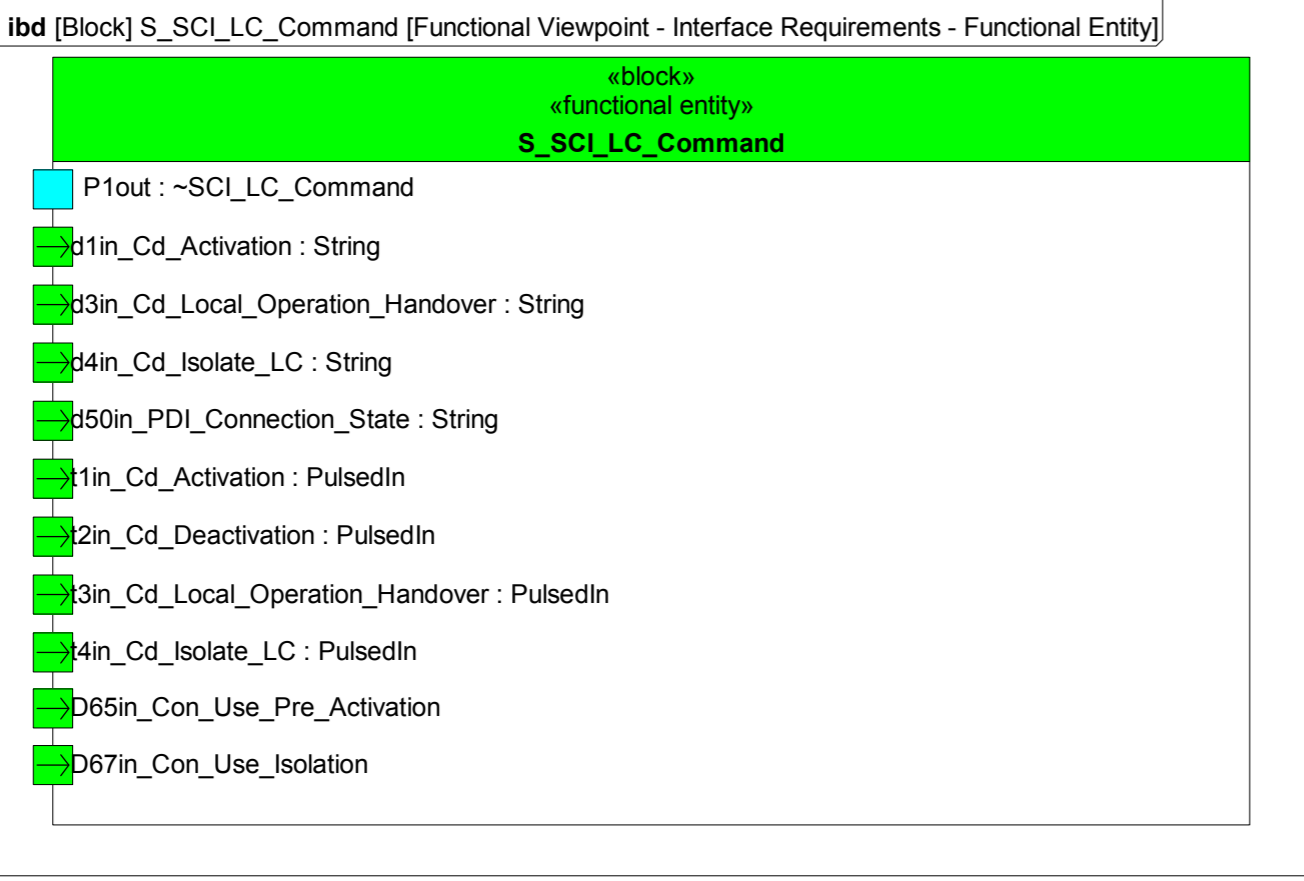
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.3298	Info	WAITING		Basic LC
Eu.LC.3299	Req	when(d51in_EST_EfeS_State = "BOOTING"){WAITING - RECEIVING_LEVEL_CROSSING_COMMANDS}		Basic LC
Eu.LC.2634	Info	F_SCI_LC_Report		Basic LC
Eu.LC.3090	Info	<p>[Block] F_SCI_LC_Report [Ports - SCI_LC IBD 3]</p> <pre> ibd [Block] F_SCI_LC_Report [Functional Viewpoint - Interface Requirements - Functional Entity] «block» «functional entity» F_SCI_LC_Report Operation «Operation» cOp1_Critical_Failure () : CriticalFailureStatusState «Operation» cOp2_Non_Critical_Failure () : NonCriticalFailureStatusState «Operation» cOp3_Status_Report () P2out : SCI_LC_Report p3inout : F_SCI_Specific d1in_Receive_LC_State : String d3in_LCPF_Protection_State : String d4in_Obstacle_Detection_Status : String d50in_PDI_Connection_State : String d61in_LC_Failure_Status : String d62in_LCPF_Failure_Status : String d6in_LC_Monitoring_Status : String d75in_LC_Failure_Status : String d9in_Changed_Monitoring_State : String d82in_Local_Operation_Handover : String d81in_Local_Operation_Handover : String d7in_Detection_Element_Status : String </pre>		Basic LC
Eu.LC.3225	Info	cOp1_Critical_Failure	<pre> 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 </pre>	Basic LC
Eu.LC.3227	Info	cOp2_Non_Critical_Failure	<pre> 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 </pre>	Basic LC
Eu.LC.3229	Info	cOp3_Status_Report	<pre> 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 send Msg_LC_Functional_Status (FunctionalStatusState.ActivatedAndProtected) to P2out; elseif (d1in_Receive_LC_State = "Deactivated") then send Msg_LC_Functional_Status (FunctionalStatusState.DeactivatedAndUnprotected)to P2out; elseif (d1in_Receive_LC_State = "Pre Activated") then send Msg_LC_Functional_Status (FunctionalStatusState.PreActivated)to P2out; else send Msg_LC_Functional_Status (FunctionalStatusState.IsolatedLC)to P2out; end if if (d4in_Obstacle_Detection_Status = "Obstacle detected in the conflict area") then </pre>	Basic LC

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
			<pre> 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.ACriticalFailureIsPre sent)to P2out; else send Msg_LC_Failure_Status (cOp2 _Non_Critical_Failure(),CriticalFailureStatusState.NoCriticalFailureIsPr esent)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 timer overrun" AND d9in_Changed_Monitoring_State = "Monitoring Parameter Set 1") then send Msg_LC_Monitoring_Status (TimerOverrunState.NoClosureTimerOverrun,ChangendMonitoringSta tus.MonitoringParameterSet1)to P2out; elseif (d6in_LC_Monitoring_Status = "Closure timer overrun occurred" AND d9in_Changed_Monitoring_State = "Monitoring Parameter Set 1") then send Msg_LC_Monitoring_Status (TimerOverrunState.ClosureTimerOverrunOccured,ChangendMonitori ngStatus.MonitoringParameterSet1)to P2out; elseif (d6in_LC_Monitoring_Status = "No Closure timer overrun" AND d9in_Changed_Monitoring_State <> "Monitoring Parameter Set 1") then send Msg_LC_Monitoring_Status (TimerOverrunState.NoClosureTimerOverrun,ChangendMonitoringSta tus.MonitoringParameterSet2)to P2out; elseif (d6in_LC_Monitoring_Status = "Closure timer overrun occurred" AND d9in_Changed_Monitoring_State <> "Monitoring Parameter Set 1") then send Msg_LC_Monitoring_Status (TimerOverrunState.ClosureTimerOverrunOccured,ChangendMonitori ngStatus.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 (TimerOverrunState.NoClosureTimerOverrun,ChangendMonitoringSta tus.MonitoringParameterSet1)to P2out; else send Msg_LC_Monitoring_Status (TimerOverrunState.ClosureTimerOverrunOccured,ChangendMonitori ngStatus.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; send Status_Report_Completed to p3inout; end if </pre>	
Eu.LC.2635	Info	d1in_Receive_LC_State		Basic LC
Eu.LC.2636	Info	d3in_LCPF_Protection_State		Basic LC
Eu.LC.2637	Info	d4in_Obstacle_Detection_Status		Basic LC
Eu.LC.2638	Info	d50in_PDI_Connection_State		Basic LC


ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.2639	Info	d61in_LC_Failure_Status		Basic LC
Eu.LC.2640	Info	d62in_LCPF_Failure_Status		Basic LC
Eu.LC.2641	Info	d6in_LC_Monitoring_Status		Basic LC
Eu.LC.2642	Info	d75in_LC_Failure_Status		Basic LC
Eu.LC.2643	Info	d7in_Detection_Element_Status		Basic LC
Eu.LC.2644	Info	d81in_Local_Operation_Handover		Basic LC
Eu.LC.2645	Info	d82in_Local_Operation_Handover		Basic LC
Eu.LC.2646	Info	d9in_Changed_Monitoring_State		Basic LC
Eu.LC.3251	Info	p3inout		Basic LC
Eu.LC.2716	Info	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	Info	<p>Functional Viewpoint - Interface Requirements - Functional Entity STD 3</p> <pre> stm [State Machine] F_SCI_LC_Report - Behaviour [Functional Viewpoint - Interface Requirements - Functional Entity STD 3] when(d50in_PDI_Connection_State = "NOT_READY_FOR_PDI_NO_SCP" OR d50in_PDI_Connection_State = "READY_FOR_PDI_NO_SCP" OR d50in_PDI_Connection_State = "NOT_READY_FOR_PDI" OR d50in_PDI_Connection_State = "READY_FOR_PDI" OR d50in_PDI_Connection_State = "SUSPENDED")/ when(d50in_PDI_Connection_State = "ESTABLISHED")/ Start_Status_Report/cOp3_Status_Report(); SENDING_LEVEL_CROSSING_REPORTS REPORT_FUNCTIONAL_STATUS FUNCTIONAL_STATUS when(d1in_Receive_LC_State = "Activated" AND (d3in_LCPF_Protection_State = "Unprotected" OR d3in_LCPF_Protection_State = "Idle"))/send Msg_LC_Functional_Status (ActivatedAndUnprotected) to P2out; when(d1in_Receive_LC_State = "Activated" AND d3in_LCPF_Protection_State = "Protected")/send Msg_LC_Functional_Status (ActivatedAndProtected) to P2out; when(d1in_Receive_LC_State = "Deactivated")/send Msg_LC_Functional_Status (DeactivatedAndUnprotected) to P2out; when(d1in_Receive_LC_State = "Pre Activated")/send Msg_LC_Functional_Status (PreActivated) to P2out; when(d1in_Receive_LC_State = "Isolated")/send Msg_LC_Functional_Status (IsolatedLC) to P2out; REPORT_OBSACLE_DETECTION_STATUS OBSTACLE_IN_CONFLICT_AREA when(d4in_Obstacle_Detection_Status = "Obstacle detected in the conflict area")/send Msg_Obstacle_Detection_Status (ObstacleInTheConflictArea) to P2out; when(d4in_Obstacle_Detection_Status = "No obstacle in the conflict area")/send Msg_Obstacle_Detection_Status (NoObstacleInTheConflictArea) to P2out; REPORT_LEVEL_CROSSING_CRITICAL_FAILURE CRITICAL_FAILURE 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; 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; REPORT_LEVEL_CROSSING_NON_CRITICAL_FAILURE NON_CRITICAL_FAILURE 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; 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; REPORT_DETECTION_ELEMENT_STATUS DETECTION_ELEMENT_STATUS when(d7in_Detection_Element_Status = "Vacant")/send Msg_Detection_Element_Status (DetectionElementsVacant) to P2out; when(d7in_Detection_Element_Status = "Occupied")/send Msg_Detection_Element_Status (DetectionElementsOccupied) to P2out; when(d7in_Detection_Element_Status = "Failed")/send Msg_Detection_Element_Status (DetectionElementsFailed) to P2out; REPORT_LOCAL_HANDOVER ALLOW_OR_RETURN_HANDOVER when(d81in_Local_Operation_Handover = "Allow handover to local operator")/send Msg_Local_Operation_Handover (AllowHandoverToLocalOperator) to P2out; when(d81in_Local_Operation_Handover = "Return handover from local operator")/send Msg_Local_Operation_Handover (ReturnHandoverFromLocalOperator) to P2out; REPORT_MONITORING_PARAMETER REPORT_CLOSURE_TIMER_OVERRUN when(d6in_LC_Monitoring_Status = "No Closure timer overrun")/d9in_Changed_Monitoring_State = "Monitoring Parameter Set 1"/ send Msg_LC_Monitoring_Status (NoClosureTimerOverrun, MonitoringParameterSet1) to P2out; when(d6in_LC_Monitoring_Status = "Closure timer overrun occurred")/d9in_Changed_Monitoring_State = "Monitoring Parameter Set 1"/ send Msg_LC_Monitoring_Status (ClosureTimerOverrunOccured, MonitoringParameterSet1) to P2out; when(d6in_LC_Monitoring_Status = "No Closure timer overrun")/d9in_Changed_Monitoring_State = "Monitoring Parameter Set 2"/ send Msg_LC_Monitoring_Status (NoClosureTimerOverrun, MonitoringParameterSet2) to P2out; when(d6in_LC_Monitoring_Status = "Closure timer overrun occurred")/d9in_Changed_Monitoring_State = "Monitoring Parameter Set 2"/ send Msg_LC_Monitoring_Status (ClosureTimerOverrunOccured, MonitoringParameterSet2) to P2out; when(d9in_Changed_Monitoring_State = "Monitoring Parameter Set 1")/d6in_LC_Monitoring_Status = "No Closure timer overrun"/ send Msg_LC_Monitoring_Status (NoClosureTimerOverrun, MonitoringParameterSet1) to P2out; when(d9in_Changed_Monitoring_State = "Monitoring Parameter Set 1")/d6in_LC_Monitoring_Status = "Closure timer overrun occurred"/ send Msg_LC_Monitoring_Status (ClosureTimerOverrunOccured, MonitoringParameterSet1) to P2out; when(d9in_Changed_Monitoring_State = "Monitoring Parameter Set 2")/d6in_LC_Monitoring_Status = "No Closure timer overrun"/ send Msg_LC_Monitoring_Status (NoClosureTimerOverrun, MonitoringParameterSet2) to P2out; when(d9in_Changed_Monitoring_State = "Monitoring Parameter Set 2")/d6in_LC_Monitoring_Status = "Closure timer overrun occurred"/ send Msg_LC_Monitoring_Status (ClosureTimerOverrunOccured, MonitoringParameterSet2) to P2out; REPORT_REQUEST_BY_LOCAL_OPERATOR REQUEST_TO_ACTIVATE_OR_DEACTIVATE when(d82in_Local_Operation_Handover = "Request to activate the level Crossing")/send Msg_Local_Request (LocalRequestToActivateTheLevelCrossing) to P2out; when(d82in_Local_Operation_Handover = "Request to deactivate the level Crossing")/send Msg_Local_Request (LocalRequestToDeactivateTheLevelCrossing) to P2out; </pre>		Basic LC Option LOH
Eu.LC.2649	Info	Initial0		Basic LC
Eu.LC.2650	Info	/{Initial0 - WAITING}		Basic LC
Eu.LC.2651	Info	SENDING_LEVEL_CROSSING_REPORTS		Basic LC
Eu.LC.2656	Info	REPORT_DETECTION_ELEMENT_STATUS		Basic LC

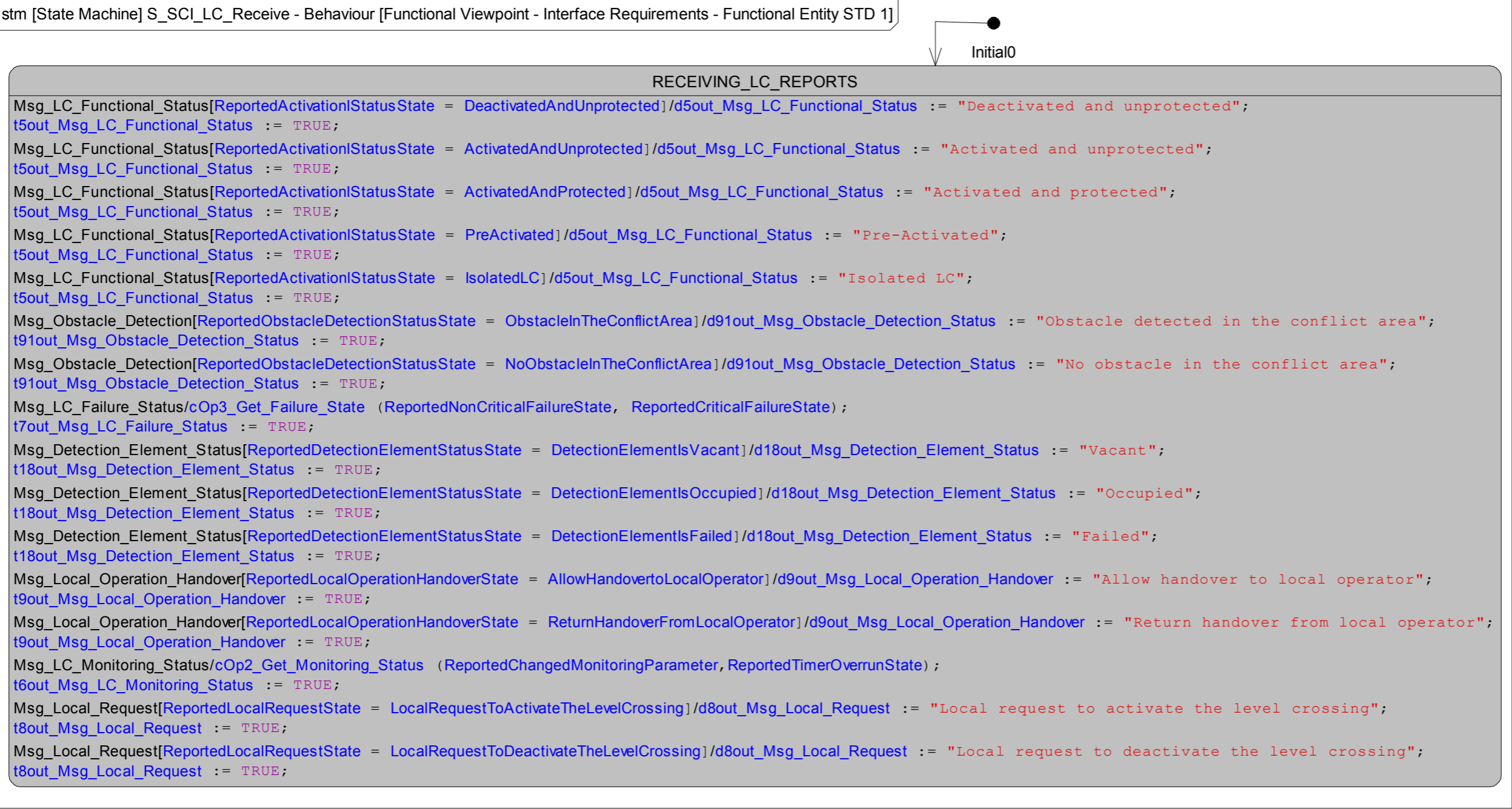
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.2657	Info	DETECTION_ELEMENT_STATUS		Basic LC
Eu.LC.3036	Req	when(d7in_Detection_Element_Status = "Failed")/send Msg_Detection_Element_Status (DetectionElementIsFailed)to P2out;{State-internal in DETECTION_ELEMENT_STATUS}		Basic LC
Eu.LC.3037	Req	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	Req	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	Info	Initial5		Basic LC
Eu.LC.2659	Req	/{Initial5 - DETECTION_ELEMENT_STATUS}		Basic LC
Eu.LC.2660	Info	REPORT_FUNCTIONAL_STATUS		Basic LC
Eu.LC.2664	Info	FUNCTIONAL_STATUS		Basic LC
Eu.LC.3040	Req	when(d1in_Receive_LC_State = "Activated" AND (d3in_LCPF_Protection_State = "Unprotected" OR d3in_LCPF_Protection_State = "Idle"))/send Msg_LC_Functional_Status (ActivatedAndUnprotected)to P2out;{State-internal in FUNCTIONAL_STATUS}		Basic LC
Eu.LC.3234	Req	when(d1in_Receive_LC_State = "Activated" AND d3in_LCPF_Protection_State = "Protected")/send Msg_LC_Functional_Status (ActivatedAndProtected) to P2out;{State-internal in FUNCTIONAL_STATUS}		Basic LC
Eu.LC.3235	Req	when(d1in_Receive_LC_State = "Deactivated")/send Msg_LC_Functional_Status (DeactivatedAndUnprotected)to P2out;{State-internal in FUNCTIONAL_STATUS}		Basic LC
Eu.LC.3236	Req	when(d1in_Receive_LC_State = "Isolated")/send Msg_LC_Functional_Status (IsolatedLC)to P2out;{State-internal in FUNCTIONAL_STATUS}		Basic LC
Eu.LC.3237	Req	when(d1in_Receive_LC_State = "Pre Activated")/send Msg_LC_Functional_Status (PreActivated)to P2out;{State-internal in FUNCTIONAL_STATUS}		Basic LC
Eu.LC.2667	Info	Atomic State		Basic LC
Eu.LC.2672	Info	Initial1		Basic LC
Eu.LC.2673	Req	/{Initial1 - FUNCTIONAL_STATUS}		Basic LC
Eu.LC.2679	Info	REPORT_LEVEL_CROSSING_CRITICAL_FAILURE		Basic LC
Eu.LC.2682	Info	Initial3		Basic LC
Eu.LC.2683	Req	/{Initial3 - CRITICAL_FAILURE}		Basic LC
Eu.LC.2686	Info	CRITICAL_FAILURE		Basic LC
Eu.LC.3238	Req	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	Req	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	Info	REPORT_LEVEL_CROSSING_NON_CRITICAL_FAILURE		Basic LC
Eu.LC.2690	Info	Initial4		Basic LC
Eu.LC.2691	Req	/{Initial4 - NON_CRITICAL_FAILURE}		Basic LC
Eu.LC.2696	Info	NON_CRITICAL_FAILURE		Basic LC
Eu.LC.3240	Req	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	Req	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	Info	REPORT_LOCAL_HANDOVER		Option LOH
Eu.LC.2700	Info	ALLOW_OR_RETURN_HANDOVER		Option LOH
Eu.LC.3048	Req	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	Req	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	Info	Initial6		Option LOH
Eu.LC.2702	Req	/{Initial6 - ALLOW_OR_RETURN_HANDOVER}		Option LOH
Eu.LC.2703	Info	REPORT_OBSTACLE_DETECTION_STATUS		Basic LC
Eu.LC.2704	Info	Initial2		Basic LC
Eu.LC.2705	Req	/{Initial2 - OBSTACLE_IN_CONFLICT_AREA}		Basic LC
Eu.LC.2706	Info	OBSTACLE_IN_CONFLICT_AREA		Basic LC
Eu.LC.3050	Req	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	Req	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	Info	REPORT_REQUEST_BY_LOCAL_OPERATOR		Option LOH

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.2708	Info	Initial8		Option LOH
Eu.LC.2709	Req	{/Initial8 - REQUEST_TO_ACTIVATE_OR_DEACTIVATE}		Option LOH
Eu.LC.2710	Info	REQUEST_TO_ACTIVATE_OR_DEACTIVATE		Option LOH
Eu.LC.3052	Req	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	Req	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	Info	REPORT_MONITORING_PARAMETER		Basic LC
Eu.LC.2712	Info	Initial7		Basic LC
Eu.LC.2713	Req	{/Initial7 - REPORT_CLOSURE_TIMER_OVERUN}		Basic LC
Eu.LC.2714	Info	REPORT_CLOSURE_TIMER_OVERUN		Basic LC
Eu.LC.3054	Req	when(d6in_LC_Monitoring_Status = "Closure timer overrun occurred")/d9in_Changed_Monitoring_State = "Monitoring Parameter Set 1"/ send Msg_LC_Monitoring_Status (ClosureTimerOverrunOccurred,MonitoringParameterSet1)to P2out;{State-internal in REPORT_CLOSURE_TIMER_OVERUN}		Basic LC
Eu.LC.3055	Req	when(d6in_LC_Monitoring_Status = "No Closure timer overrun")/d9in_Changed_Monitoring_State = "Monitoring Parameter Set 1"/ send Msg_LC_Monitoring_Status (NoClosureTimerOverrun,MonitoringParameterSet1)to P2out;{State-internal in REPORT_CLOSURE_TIMER_OVERUN}		Basic LC
Eu.LC.3242	Req	when(d6in_LC_Monitoring_Status = "Closure timer overrun occurred")/d9in_Changed_Monitoring_State = "Monitoring Parameter Set 2"/ send Msg_LC_Monitoring_Status (ClosureTimerOverrunOccurred,MonitoringParameterSet2)to P2out;{State-internal in REPORT_CLOSURE_TIMER_OVERUN}		Basic LC
Eu.LC.3243	Req	when(d6in_LC_Monitoring_Status = "No Closure timer overrun")/d9in_Changed_Monitoring_State = "Monitoring Parameter Set 2"/ send Msg_LC_Monitoring_Status (NoClosureTimerOverrun,MonitoringParameterSet2)to P2out;{State-internal in REPORT_CLOSURE_TIMER_OVERUN}		Basic LC
Eu.LC.3244	Req	when(d9in_Changed_Monitoring_State = "Monitoring Parameter Set 1")/d6in_LC_Monitoring_Status = "Closure timer overrun occurred"/ send Msg_LC_Monitoring_Status (ClosureTimerOverrunOccurred,MonitoringParameterSet1)to P2out;{State-internal in REPORT_CLOSURE_TIMER_OVERUN}		Basic LC
Eu.LC.3245	Req	when(d9in_Changed_Monitoring_State = "Monitoring Parameter Set 1")/d6in_LC_Monitoring_Status = "No Closure timer overrun"/ send Msg_LC_Monitoring_Status (NoClosureTimerOverrun,MonitoringParameterSet1)to P2out;{State-internal in REPORT_CLOSURE_TIMER_OVERUN}		Basic LC
Eu.LC.3246	Req	when(d9in_Changed_Monitoring_State = "Monitoring Parameter Set 2")/d6in_LC_Monitoring_Status = "Closure timer overrun occurred"/ send Msg_LC_Monitoring_Status (ClosureTimerOverrunOccurred,MonitoringParameterSet2)to P2out;{State-internal in REPORT_CLOSURE_TIMER_OVERUN}		Basic LC
Eu.LC.3247	Req	when(d9in_Changed_Monitoring_State = "Monitoring Parameter Set 2")/d6in_LC_Monitoring_Status = "No Closure timer overrun"/ send Msg_LC_Monitoring_Status (NoClosureTimerOverrun,MonitoringParameterSet2)to P2out;{State-internal in REPORT_CLOSURE_TIMER_OVERUN}		Basic LC
Eu.LC.2715	Req	when(d50in_PDI_Connection_State = "NOT_READY_FOR_PDI_NO_SCP" OR d50in_PDI_Connection_State = "READY_FOR_PDI_NO_SCP" OR d50in_PDI_Connection_State = "NOT_READY_FOR_PDI" OR d50in_PDI_Connection_State = "READY_FOR_PDI" OR d50in_PDI_Connection_State = "SUSPENDED")/{SENDING_LEVEL_CROSSING_REPORTS - WAITING}		Basic LC
Eu.LC.3286	Info	WAITING		Basic LC
Eu.LC.3287	Req	Start_Status_Report/cOp3_Status_Report();{State-internal in WAITING}		Basic LC
Eu.LC.3288	Req	when(d50in_PDI_Connection_State = "ESTABLISHED")/{WAITING - SENDING_LEVEL_CROSSING_REPORTS}		Basic LC
Eu.LC.2717	Info	S_SCI_LC_Command		Basic LC
Eu.LC.3091	Info	[Block] S_SCI_LC_Command [Ports - SCI_LC IBD 0] 		Basic LC
Eu.LC.2718	Info	d1in_Cd_Activation	The FlowPort d1in_Cd_Activation belongs to T1_Cd_Activation.	Basic LC

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.2719	Info	d3in_Cd_Local_Operation_Handover	The FlowPort d3in_Cd_Local_Operation_Handover belongs to T3_Cd_Local_Operation_Handover.	Basic LC
Eu.LC.2720	Info	d4in_Cd_Isolate_LC	The FlowPort d4in_Cd_Isolate_LC belongs to T4_Cd_Isolate_LC.	Basic LC
Eu.LC.2721	Info	d50in_PDI_Connection_State		Basic LC
Eu.LC.2728	Info	t1in_Cd_Activation	The FlowPort T1_Cd_Activation refines the Flow Property Cd_Activation.	Basic LC
Eu.LC.2729	Info	t2in_Cd_Deactivation	The FlowPort T2_Cd_Deactivation refines the Flow Property Cd_Deactivation.	Basic LC
Eu.LC.2730	Info	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	Info	t4in_Cd_Isolate_LC	The FlowPort T4_Cd_Isolate_LC refines the Flow Property Cd_Isolate_LC.	Basic LC
Eu.LC.3252	Info	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	Info	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	Info	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	Info	<p>Functional Viewpoint - Interface Requirements - Functional Entity STD 0</p> <pre> stm [State Machine] S_SCI_LC_Command - Behaviour [Functional Viewpoint - Interface Requirements - Functional Entity STD 0] stateDiagram-v2 [*] --> Initial0 state "SENDING_COMMANDS" as SENDING_COMMANDS Initial0 --> SENDING_COMMANDS SENDING_COMMANDS --> SENDING_COMMANDS : when(t1in_Cd_Activation) [d1in_Cd_Activation = "Activation" AND d50in_PDI_Connection_State = "ESTABLISHED"] / send Cd_Activation (Activation) to P1out; SENDING_COMMANDS --> SENDING_COMMANDS : 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; SENDING_COMMANDS --> SENDING_COMMANDS : 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; SENDING_COMMANDS --> SENDING_COMMANDS : when(t2in_Cd_Deactivation) [d50in_PDI_Connection_State = "ESTABLISHED"] / send Cd_Deactivation (Deactivation) to P1out; SENDING_COMMANDS --> SENDING_COMMANDS : 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; SENDING_COMMANDS --> SENDING_COMMANDS : 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; SENDING_COMMANDS --> SENDING_COMMANDS : 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; SENDING_COMMANDS --> SENDING_COMMANDS : 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; </pre>	Basic LC Option LOH	
Eu.LC.2725	Info	Initial0		Basic LC
Eu.LC.2726	Req	{/Initial0 - SENDING_COMMANDS}		Basic LC
Eu.LC.2727	Info	SENDING_COMMANDS		Basic LC
Eu.LC.3056	Req	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	Req	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	Req	when(t2in_Cd_Deactivation)[d50in_PDI_Connection_State = "ESTABLISHED"] / send Cd_Deactivation (Deactivation) to P1out; {State-internal in SENDING_COMMANDS}		Basic LC
Eu.LC.3059	Req	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	Req	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	Req	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	Req	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

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.3254	Req	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.3819		D84in_Con_Use_Local_Operation	The port D84in_Con_Use_Local_Operation provides configuration values for Local_operator. true: Local_operator is used. false: Local_operator is not used.	
Eu.LC.2732	Info	S_SCI_LC_Receive		Basic LC
Eu.LC.3092	Info	<p>[Block] S_SCI_LC_Receive [Ports - SCI_LC IBD 1]</p> <p>ibd [Block] S_SCI_LC_Receive [Functional Viewpoint - Interface Requirements - Functional Entity]</p> 		Basic LC
Eu.LC.3255	Info	cOp2_Get_Monitoring_Status	<pre> if ParameterChangedMonitoringParameters = ChangendMonitoringStatus.MonitoringParameterSet1 then d62out_Msg_LC_Monitoring_Status_Changed_Monitoring_Pa rameter := "Monitoring Parameter Set 1"; elseif ParameterChangedMonitoringParameters = ChangendMonitoringStatus.MonitoringParameterSet2 then d62out_Msg_LC_Monitoring_Status_Changed_Monitoring_Pa rameter := "Monitoring Parameter Set 2"; end if if ParameterTimerOverrun = TimerOverrunState.ClosureTimerOverrunOccured then d61out_Msg_LC_Monitoring_Status_Closure_timer : = "Closure timer overrun occurred"; elseif ParameterTimerOverrun = TimerOverrunState.NoClosureTimerOverrun then d61out_Msg_LC_Monitoring_Status_Closure_timer : = "No Closure timer overrun"; end if </pre>	Basic LC
Eu.LC.3257	Info	cOp3_Get_Failure_State	<pre> 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"; </pre>	Basic LC

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
			elseif ParameterNonCriticalFailure = NonCriticalFailureStatusState.NoNonCriticalFailureIsPresent then d71out_Msg_LC_Failure_Status_Non_Critical := "No non critical failure is present"; end if	
Eu.LC.2733	Info	d18out_Msg_Detection_Element_Status	The FlowPort d18out_Msg_Detection_Element_Status belongs to T18_Msg_Detection_Element_Status.	Basic LC
Eu.LC.2734	Info	d50in_PDI_Connection_State		Basic LC
Eu.LC.2735	Info	d5out_Msg_LC_Functional_Status	The FlowPort d5out_Msg_LC_Functional_Status belongs to T5_Msg_LC_Functional_Status.	Basic LC
Eu.LC.2736	Info	d61out_Msg_LC_Monitoring_Status_Closure_timer	The FlowPort d61out_Msg_LC_Monitoring_Status_Closure_timer belongs to T6_Msg_LC_Monitoring_Status.	Basic LC
Eu.LC.2737	Info	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	Info	d8out_Msg_Local_Request	The FlowPort d8out_Msg_Local_Request belongs to T8_Msg_Local_Request.	Option LOH
Eu.LC.2739	Info	d91out_Msg_Obstacle_Detection_Status	The FlowPort d91out_Msg_Obstacle_Detection_Status belongs to T91_Msg_Obstacle_Detection_Status.	Basic LC
Eu.LC.2740	Info	d9out_Msg_Local_Operation_Handover	The FlowPort d9out_Msg_Local_Operation_Handover belongs to T9_Msg_Local_Operation_Handover.	Option LOH
Eu.LC.3259	Info	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	Info	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	Info	t18out_Msg_Detection_Element_Status	The FlowPort t18out_Msg_Detection_Element_Status refines the Flow Property Vacated_Detection_Element, Occupied_Detection_Element or Failed_Detection_Element.	Basic LC
Eu.LC.2748	Info	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	Info	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	Info	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	Info	t8out_Msg_Local_Request	The FlowPort T8_Msg_Local_Request refines the Flow Property Activate and Deactivate.	Option LOH
Eu.LC.2752	Info	t91out_Msg_Obstacle_Detection_Status	The FlowPort t91out_Msg_Obstacle_Detection_Status refines the Flow Property Status_Level_Crossing_Protection_Facility.	Basic LC
Eu.LC.2753	Info	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	Info	P2in	The port P2in exchanges information objects according to SCI_LC_Subsystem_EIL.	Basic LC
Eu.LC.2742	Req	S_SCI_LC_Receive - Behaviour		Basic LC

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.2743	Info	<p>Functional Viewpoint - Interface Requirements - Functional Entity STD 1</p> <p>stm [State Machine] S_SCI_LC_Receive - Behaviour [Functional Viewpoint - Interface Requirements - Functional Entity STD 1]</p>  <pre> stateDiagram-v2 [*] --> Initial0 state "RECEIVING_LC_REPORTS" as RECEIVING_LC_REPORTS RECEIVING_LC_REPORTS --> RECEIVING_LC_REPORTS : Msg_LC_Functional_Status[ReportedActivationStatusState = DeactivatedAndUnprotected]/d5out_Msg_LC_Functional_Status := "Deactivated and unprotected"; t5out_Msg_LC_Functional_Status := TRUE; RECEIVING_LC_REPORTS --> RECEIVING_LC_REPORTS : Msg_LC_Functional_Status[ReportedActivationStatusState = ActivatedAndUnprotected]/d5out_Msg_LC_Functional_Status := "Activated and unprotected"; t5out_Msg_LC_Functional_Status := TRUE; RECEIVING_LC_REPORTS --> RECEIVING_LC_REPORTS : Msg_LC_Functional_Status[ReportedActivationStatusState = ActivatedAndProtected]/d5out_Msg_LC_Functional_Status := "Activated and protected"; t5out_Msg_LC_Functional_Status := TRUE; RECEIVING_LC_REPORTS --> RECEIVING_LC_REPORTS : Msg_LC_Functional_Status[ReportedActivationStatusState = PreActivated]/d5out_Msg_LC_Functional_Status := "Pre-Activated"; t5out_Msg_LC_Functional_Status := TRUE; RECEIVING_LC_REPORTS --> RECEIVING_LC_REPORTS : Msg_LC_Functional_Status[ReportedActivationStatusState = IsolatedLC]/d5out_Msg_LC_Functional_Status := "Isolated LC"; t5out_Msg_LC_Functional_Status := TRUE; RECEIVING_LC_REPORTS --> RECEIVING_LC_REPORTS : Msg_Obstacle_Detection[ReportedObstacleDetectionStatusState = ObstacleInTheConflictArea]/d91out_Msg_Obstacle_Detection_Status := "Obstacle detected in the conflict area"; t91out_Msg_Obstacle_Detection_Status := TRUE; RECEIVING_LC_REPORTS --> RECEIVING_LC_REPORTS : Msg_Obstacle_Detection[ReportedObstacleDetectionStatusState = NoObstacleInTheConflictArea]/d91out_Msg_Obstacle_Detection_Status := "No obstacle in the conflict area"; t91out_Msg_Obstacle_Detection_Status := TRUE; RECEIVING_LC_REPORTS --> RECEIVING_LC_REPORTS : Msg_LC_Failure_Status/cOp3_Get_Failure_State (ReportedNonCriticalFailureState, ReportedCriticalFailureState); t7out_Msg_LC_Failure_Status := TRUE; RECEIVING_LC_REPORTS --> RECEIVING_LC_REPORTS : Msg_Detection_Element_Status[ReportedDetectionElementStatusState = DetectionElementIsVacant]/d18out_Msg_Detection_Element_Status := "Vacant"; t18out_Msg_Detection_Element_Status := TRUE; RECEIVING_LC_REPORTS --> RECEIVING_LC_REPORTS : Msg_Detection_Element_Status[ReportedDetectionElementStatusState = DetectionElementIsOccupied]/d18out_Msg_Detection_Element_Status := "Occupied"; t18out_Msg_Detection_Element_Status := TRUE; RECEIVING_LC_REPORTS --> RECEIVING_LC_REPORTS : Msg_Detection_Element_Status[ReportedDetectionElementStatusState = DetectionElementIsFailed]/d18out_Msg_Detection_Element_Status := "Failed"; t18out_Msg_Detection_Element_Status := TRUE; RECEIVING_LC_REPORTS --> RECEIVING_LC_REPORTS : Msg_Local_Operation_Handover[ReportedLocalOperationHandoverState = AllowHandoverToLocalOperator]/d9out_Msg_Local_Operation_Handover := "Allow handover to local operator"; t9out_Msg_Local_Operation_Handover := TRUE; RECEIVING_LC_REPORTS --> RECEIVING_LC_REPORTS : Msg_Local_Operation_Handover[ReportedLocalOperationHandoverState = ReturnHandoverFromLocalOperator]/d9out_Msg_Local_Operation_Handover := "Return handover from local operator"; t9out_Msg_Local_Operation_Handover := TRUE; RECEIVING_LC_REPORTS --> RECEIVING_LC_REPORTS : Msg_LC_Monitoring_Status/cOp2_Get_Monitoring_Status (ReportedChangedMonitoringParameter, ReportedTimerOverrunState); t6out_Msg_LC_Monitoring_Status := TRUE; RECEIVING_LC_REPORTS --> RECEIVING_LC_REPORTS : Msg_Local_Request[ReportedLocalRequestState = LocalRequestToActivateTheLevelCrossing]/d8out_Msg_Local_Request := "Local request to activate the level crossing"; t8out_Msg_Local_Request := TRUE; RECEIVING_LC_REPORTS --> RECEIVING_LC_REPORTS : Msg_Local_Request[ReportedLocalRequestState = LocalRequestToDeactivateTheLevelCrossing]/d8out_Msg_Local_Request := "Local request to deactivate the level crossing"; t8out_Msg_Local_Request := TRUE; </pre>		Basic LC Option LOH
Eu.LC.2744	Info	Initial0		Basic LC
Eu.LC.2745	Req	/{Initial0 - RECEIVING_LC_REPORTS}		Basic LC
Eu.LC.2746	Info	RECEIVING_LC_REPORTS		Basic LC
Eu.LC.3063	Req	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	Req	Msg_LC_Functional_Status[ReportedActivationStatusState = DeactivatedAndUnprotected]/d5out_Msg_LC_Functional_Status := "Deactivated and unprotected"; t5out_Msg_LC_Functional_Status := TRUE;{State-internal in RECEIVING_LC_REPORTS}		Basic LC
Eu.LC.3065	Req	Msg_LC_Functional_Status[ReportedActivationStatusState = 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	Req	Msg_LC_Functional_Status[ReportedActivationStatusState = 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	Req	Msg_LC_Monitoring_Status/cOp2_Get_Monitoring_Status (ReportedChangedMonitoringParameter,ReportedTimerOverrunState); t6out_Msg_LC_Monitoring_Status := TRUE;{State-internal in RECEIVING_LC_REPORTS}		Basic LC
Eu.LC.3070	Req	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	Req	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
Eu.LC.3072	Req	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	Req	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	Req	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	Req	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	Req	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	Req	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

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.3078	Req	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	Req	Msg_LC_Functional_Status[ReportedActivation]StatusState = 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	Req	Msg_LC_Functional_Status[ReportedActivation]StatusState = IsolatedLC]/d5out_Msg_LC_Functional_Status := "Isolated LC"; 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.		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	Req	Occupied_Detection_Element	The Subsystem - Level Crossing detects that the Detection element is occupied.	Basic LC
Eu.LC.151	Req	Vacated_Detection_Element	The Subsystem - Level Crossing detects that the Detection element is vacant.	Basic LC
Eu.LC.152	Req	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	Req	Activate	The Subsystem - Level Crossing detects the local activation of the Level Crossing protection facility from the Local operator.	Option LOH
Eu.LC.156	Req	Deactivate	The Subsystem - Level Crossing detects the local deactivation of the Level Crossing protection facility from the Local operator.	Option LOH
Eu.LC.159	Req	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	Req	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
Eu.LC.161	Req	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	Req	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	Req	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	Req	Activate	The Subsystem - Level Crossing request the Level Crossing protection facility to activate the Level Crossing protection facility.	Basic LC
Eu.LC.167	Req	Deactivate	The Subsystem - Level Crossing request the Level Crossing protection facility to deactivate the Level Crossing protection facility.	Basic LC
Eu.LC.168	Req	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	Req	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	Req	Status_Level_Crossing_Protection_Facility	The Level Crossing protection facility reports its new status to the Subsystem - Level Crossing.	Basic LC

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
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].		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: <ul style="list-style-type: none"> • LCPF_Monitoring_Status_Barrier_Position • LCPF_Monitoring_Status_Barrier_Movement • LCPF_Monitoring_Status_Barrier_Intact • LCPF_Monitoring_Status_Road_Lights • LCPF_Monitoring_Status_Hardware • LCPF_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
Eu.LC.1317	Info	General assumptions:		Basic LC
Eu.LC.1315	Info	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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	Info	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.3925	Info	The response time to send the corresponding message telegram to the Subsystem – Electronic Interlocking after detecting a change of state at the Level Crossing Protection Facility shall be defined by national specifications. Note: In future phases of the System Pillar, national specifications will be replaced by harmonised specifications.		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

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.LC.1275	Req	• The applicable timers defined in chapter Definition of time values (Eu.LC.172).		Basic LC
Eu.LC.1321	Req	• The usage of the Closure Timer.		Basic LC
Eu.LC.1323	Req	• The usage of the PDI Loss Deactivation Timer.		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
Eu.LC.1292	Req	• The engineering data is safety-relevant. This data shall be used to calculate the CSS.		Basic LC