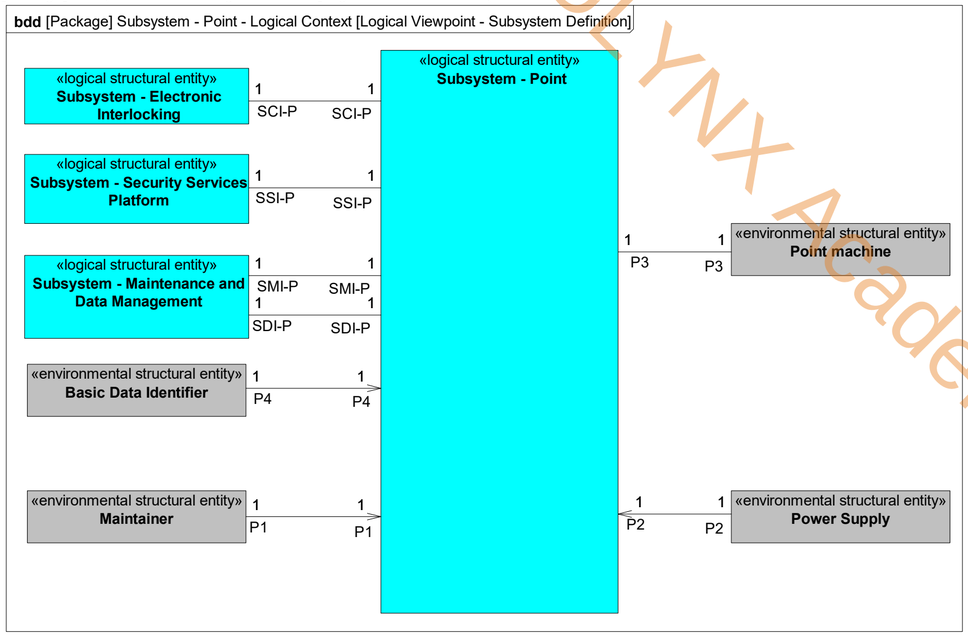
EULYNX Initiative

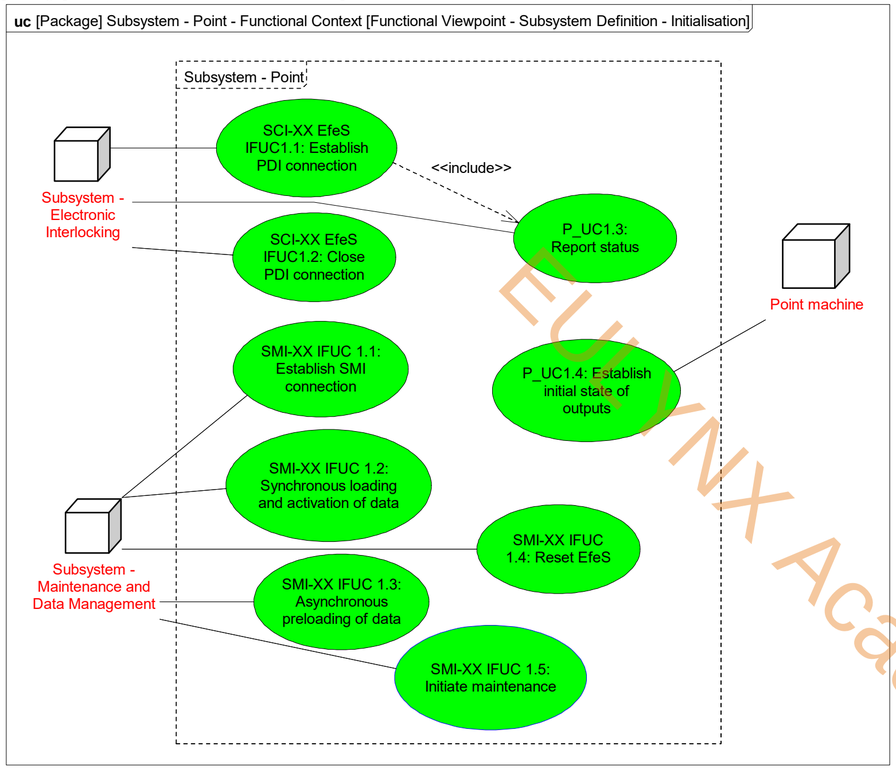
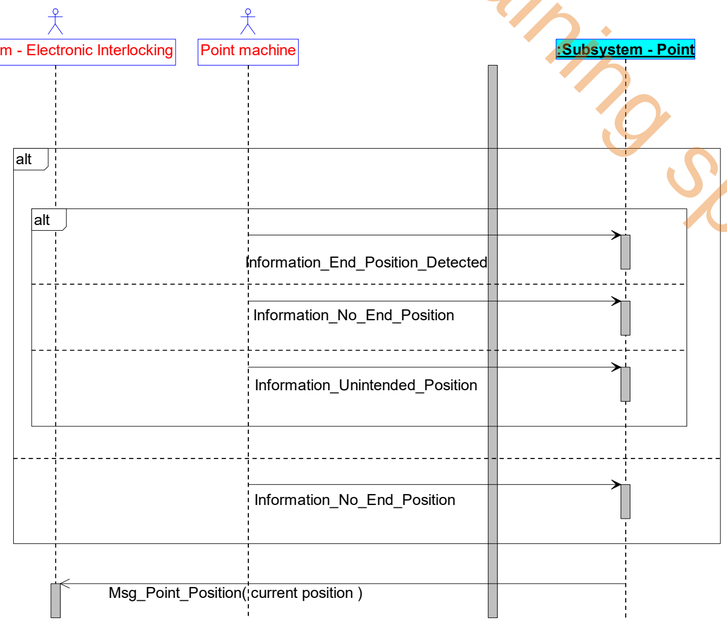
Academy - Requirements specification for subsystem Point - without Redrive

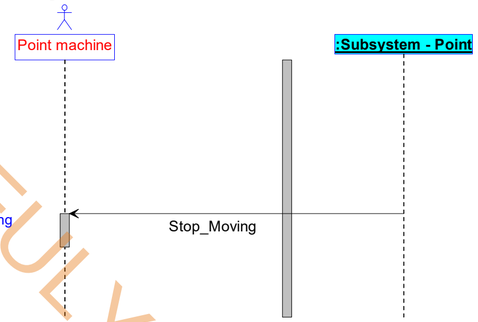
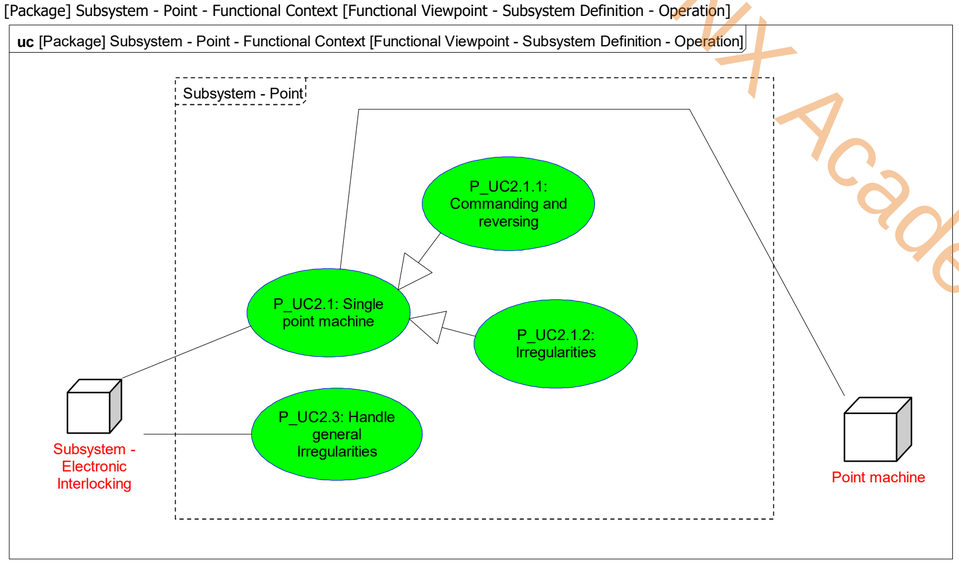
Document number: Eu.Doc.36
no Baseline, Current Version!

Contents

ID	Type	Requirement Part 1	Requirement Part 2
Eu.P.1	Head	**1 Introduction**	
Eu.P.2	Head	**1.1 Release information**	
Eu.P.3	Info	[Eu.Doc.36] Requirements specification for subsystem Point CENELEC Phase: 4 Version: 1.0 (0.A) Approval date: 24.05.2024	
Eu.P.3032	Info	**Version history**	
Eu.P.7369	Info	version number: 1.0 (0.A) date: 28.05.2024 author: Philipp Wolber, Nico Huurman changes: EULYNX Academy base version	
Eu.P.7	Head	**1.2 Impressum**	
Eu.P.8	Info	Publisher: **EULYNX Academy**	
Eu.P.3038	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.P.10	Head	**1.3 Purpose**	
Eu.P.11	Info	The purpose of the document is the specification of requirements for the Subsystem - Point.	
Eu.P.12	Info	This document describes functional, non-functional and technical requirements for the Subsystem - Point and functional requirements for interface SCI-P.	
Eu.P.13	Info	This document is meant solely for training purposes of the EULYNX Academy.	
Eu.P.15	Head	**1.4 Applicable standards and regulations**	
Eu.P.314	Info	The applicable standards and regulations used in EULYNX are listed in the EULYNX Reference Document List [Eu.Doc.12].	
Eu.P.35	Head	**1.5 Applicable documents**	
Eu.P.36	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.P.51	Head	**1.6 Terms and abbreviations**	
Eu.P.52	Info	The terms and abbreviations are listed in the EULYNX Glossary [Eu.Doc.9].	
Eu.P.1350	Head	**1.7 Variability management**	
Eu.P.1351	Info	This document describes harmonised requirements. Variability management is not applicable.	
Eu.P.3024	Head	**1.8 Definition of object types**	
Eu.P.3025	Info	The following definition for object types is applied in this document:	
Eu.P.3026	Info	• "Req" - This denotes a mandatory requirement.	
Eu.P.7246	Info	• "Def" - This denotes referenceable model elements that are used in the model-based creation of requirements	
Eu.P.3027	Info	• "Info" - This denotes additional information to help understand the specification. These objects do not specify any additional requirements.	
Eu.P.3028	Info	• "Head" - This denotes chapter headings.	
Eu.P.53	Head	**1.9 Modelling**	
Eu.P.54	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 - Point operational in stimulus-response form. Furthermore the information objects (stimuli and responses) exchanged over the interfaces of the Subsystem - Point are defined.	
Eu.P.55	Info	The diagrams presented in this document are modelled in SysML [SysML].	
Eu.P.3050	Info	The rules for the interpretation of the model based parts of specification are defined in [Eu.Doc.29].	
Eu.P.3051	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.P.3052	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.P.3053	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.P.3054	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.P.7247	Info	State machines or several state machines linked together in a Functional Architecture define the totality of all functional requirements of an SUS or an SIUS in a coherent and consistent manner. State diagrams of a corresponding state machine are marked with the object type “Req”. For the later design and implementation, it is not the description language SysML that is binding, but the domain-specific meaning expressed by it. The specified behavior can be converted into a vendor specific language but must retain the domain specific meaning describing the functional requirements. The specific model elements are additionally specified and defined by object type “Def” to allow for traceability to supplier designs or test cases. The compliance of products to the specifications must be demonstrated by testing against EULYNX test cases, which are derived from the functionality specified by the models.	
Eu.P.57	Head	**2 Conditions of use**	
Eu.P.4952	Req	All references to [Eu.Doc.20] refer to version 4.0 (4.A) of that document.	
Eu.P.6374	Req	All references to [Eu.Doc.119] refer to version 1.0 (4.A) of that document.	

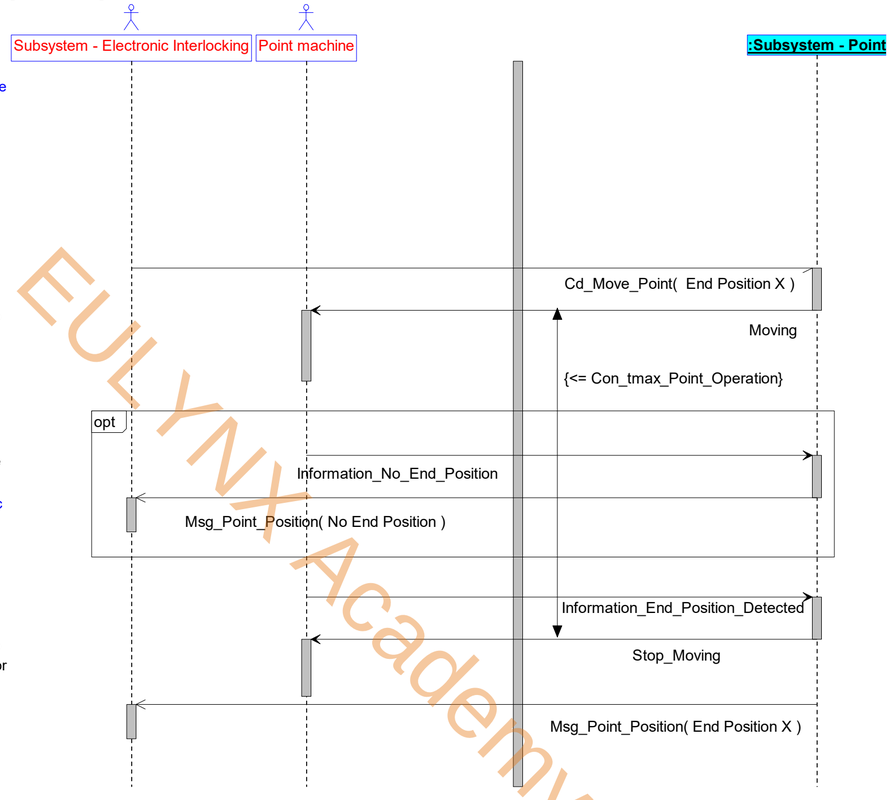
ID	Type	Requirement Part 1	Requirement Part 2
Eu.P.6375	Info	References to [Eu.Doc.120] do not refer to a concrete version of that document. The applicable version shall be defined by national specifications. Note: In future phases of the System Pillar, national specifications will be replaced by harmonised specifications.	
Eu.P.58	Info	The specifications defined in this document shall follow the requirements of the EULYNX System Architecture Specification [Eu.Doc.16].	
Eu.P.884	Head	**3 Functional requirements specification**	
Eu.P.7889	Head	**3.1 Subsystem - Point - General Infos and Assumptions**	
Eu.P.8054	Info	The defined model elements represent the Subsystem - Point in • The functional architectures shown in the internal block diagrams • The defined number of Point machines in the state diagrams • The timing behaviour related to individual Point Machines.	
Eu.P.8032	Head	**3.2 Subsystem - Point - Logical Viewpoint**	
Eu.P.8033	Head	**3.2.1 Subsystem - Point - Logical Context**	
Eu.P.8034	Def	[Package] Subsystem - Point - Logical Context [Logical Viewpoint - Subsystem Definition]  	
Eu.P.8035	Req	The Subsystem - Point shall provide a logical interface SCI-P to exactly one Subsystem - Electronic Interlocking.	
Eu.P.8036	Req	The Subsystem - Point shall provide a logical interface SMI-P to exactly one Subsystem - Maintenance and Data Management.	
Eu.P.8037	Req	The Subsystem - Point shall provide a logical interface SDI-P to exactly one Subsystem - Maintenance and Data Management.	
Eu.P.8038	Req	The Subsystem - Point shall provide a logical interface SSI-P to exactly one Subsystem - Security Services Platform.	
Eu.P.8039	Req	The Subsystem - Point shall provide a logical interface P4 to exactly one Basic Data identifier.	
Eu.P.8040	Req	The Subsystem - Point shall provide a logical interface P1 to exactly one Maintainer.	
Eu.P.8041	Req	The Subsystem - Point shall provide a logical interface P2 to exactly one Power Supply.	
Eu.P.8042	Req	The Subsystem - Point shall provide a logical interface P3 to exactly one Point machine.	
Eu.P.7749	Head	**3.3 Subsystem - Point - Functional Viewpoint**	
Eu.P.7750	Head	**3.3.1 Definition of time values**	
Eu.P.7751	Def	Con_tmax_Point_Operation	Con_tmax_Point_Operation defines the max. time period the Point has to arrive to an End position, starting with the command Moving to the Point machine. After that time period the command Moving to the Point machine is set to Stop moving.
Eu.P.7752	Info	The generic time values for SCI are specified in [Eu.Doc.119].	
Eu.P.7763	Head	**3.3.2 Subsystem - Point - Functional Context**	

ID	Type	Requirement Part 1	Requirement Part 2
Eu.P.7765	Info	[Package] Subsystem - Point - Functional Context [Functional Viewpoint - Subsystem Definition - Initialisation] 	
Eu.P.7767	Info	P_UC1.3: Report status	The Subsystem-UseCase P_UC1.3: Report status defines a scenario about the transmission of status data of Subsystem - Point to Subsystem - Electronic Interlocking, while Process Data Interface protocol connection is establishing.
Eu.P.7768	Info	Main Success Scenario: Report point position status with single point machine [P SD 1.3.1] **P_UC1.3: Report status** Main Success Scenario: Report point position status with single point machine [P SD 1.3.1] **Interaction 1.3.1.A** **alt** [The Subsystem - Point is configured with a non-4-wire interface to the point machine OR the Subsystem - Point is configured with a 4-wire interface to the point machine AND the last commanded position is End position "Y"] **alt** [The Point is in an End position "Y"] **1.a1.a1** - The Subsystem - Point receives from the Point machine the Information that the Point is in an End position "Y". **else alt** [The Point is in No end position] **1.a1.b1** - The Subsystem - Point receives from the Point machine the Information that the Point is in No end position. **else alt** [The Point is in a Unintended position] **1.a1.c1** - The Subsystem - Point receives from the Point machine the Information that the Point is in a Unintended position. **end alt** **else alt** [The Subsystem - Point is configured with a 4-wire interface to the Point machine AND the last commanded position is not available] **1.b1** - The Subsystem - Point receives from the Point machine the Information that the Point is in No end position **end alt** **Interaction 1.3.1.1.1.B** **2.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking the Point position. 	

ID	Type	Requirement Part 1	Requirement Part 2
Eu.P.7769	Info	P_UC1.4: Establish initial state of outputs	The Subsystem-UseCase P_UC1.4: Establish initial state of outputs defines the main success scenario for establishing the initial state of outputs of the Subsystem - Point.
Eu.P.7771	Info	Main Success Scenario: Set Initial State of Outputs with single point machine [P SD 1.4.1] **P_UC1.4: Establish initial state of outputs** Main Success Scenario: Set Initial State of Outputs with single point machine [P SD 1.4.1] **Precondition:** The Subsystem - Point is in the state BOOTING. **Interaction 1.4.1.A:** **1.** - The Subsystem - Point enters the state INITIALISING. **2.** The Subsystem - Point sends the Command to the Point machine to Stop moving *the* Point machine. **Postconditions:** The Subsystem - Point is in the state INITIALISING. The Initial State Of Outputs of the Subsystem - Point has been set. 	Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE.
Eu.P.7766	Info	[Package] Subsystem - Point - Functional Context [Functional Viewpoint - Subsystem Definition - Operation] 	
Eu.P.7790	Info	P_UC2.1: Single point machine	The Subsystem-UseCase "P_UC2.1: Single point machine" defines the behaviour of the Subsystem - Point which works with a single point machine via non-4-wire interface. The behaviour will be defined in the following UseCases: P_UC2.1.1: Commanding and reversing P_UC2.1.2: Irregularities
Eu.P.7772	Info	P_UC2.1.1: Commanding and reversing	The Subsystem-UseCase "P_UC2.1.1: Commanding and reversing" defines the behaviour of commanding and reversing a single point machine via non-4-wire interface.

ID	Type	Requirement Part 1	Requirement Part 2
Eu.P.7778	Info	Main Success Scenario: Moving of the Point with a single point machine [P SD 2.1.1.1]	Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE.

P_UC2.1.1: Commanding and reversing



Main Success Scenario: Moving of the Point with a single point machine [P SD 2.1.1.1]

Precondition:

The Subsystem - Point is in the state OPERATIONAL.
The Subsystem - Point is in:
- an End position "Y", or
- No end position, or
- an Unintended position.

Interaction 2.1.1.1.A:

1. - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position "X".

2. The Subsystem - Point sends the Command to the Point machine to move the Point machine to an End position "X". At this moment the Subsystem - Point starts to monitor the time period Con_tmax_Point_Operation.

Interaction 2.1.1.1.B:

opt [The Subsystem - Point was previously in an End position or a Unintended position]

3.a1 - The Subsystem - Point receives from the Point machine the Information that the Point machine is in No end position.

3.b1 The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in No end position.

end opt

Interaction 2.1.1.1.C:

4. - The Subsystem - Point receives from the Point machine the Information that the Point machine is in an End position "X".

5. The Subsystem - Point sends the Command to the Point machine to stop moving the Point machine. The Subsystem - Point stops to monitor the time period Con_tmax_Point_Operation.

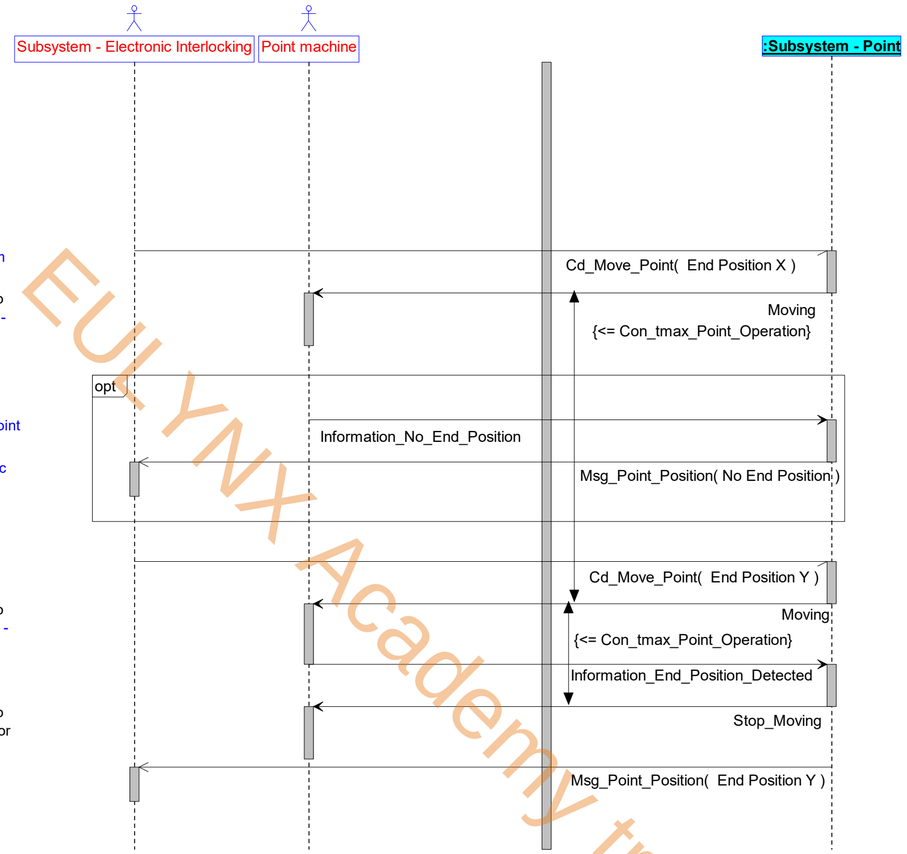
6. The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in an End position "X".

Postcondition:

The Subsystem - Point is in an End position "X".

ID	Type	Requirement Part 1	Requirement Part 2
Eu.P.7775	Info	Alternative Scenario: Reversing Point [P SD 2.1.1.2]	Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE.

P_UC2.1.1: Commanding and reversing



Alternative Scenario: Reversing Point [P SD 2.1.1.2]

Precondition:

The Subsystem - Point is in the state OPERATIONAL.
The Subsystem - Point is in:
- an End position "Y", or
- No end position, or
- an Unintended position.

Interaction 2.1.1.2.A:

1. - The Subsystem - Point receives the Command from the Subsystem - Electronic Interlocking to move the Point to an End position "X".

2. The Subsystem - Point sends the Command to the Point machine to move the Point to an End position "X". At this moment the Subsystem - Point starts to monitor the time period Con_tmax_Point_Operation.

Interaction 2.1.1.2.B:

opt [The Subsystem - Point was previously in an End position or an Unintended position]

3.a1 - The Subsystem - Point receives the Information from the Point machine that the Point is in No end position.

3.a2 The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in No end position.

end opt

Interaction 2.1.1.2.C:

4. - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position "Y".

5. The Subsystem - Point sends the Command to the Point machine to move the Point to an End position "Y". At this moment the Subsystem - Point restarts to monitor the time period Con_tmax_Point_Operation.

6. The Subsystem - Point receives from the Point machine the Information that the Point machine is in an End position "Y".

7. The Subsystem - Point sends the Command to the Point machine to stop moving the Point machine. The Subsystem - Point stops to monitor the time period Con_tmax_Point_Operation.

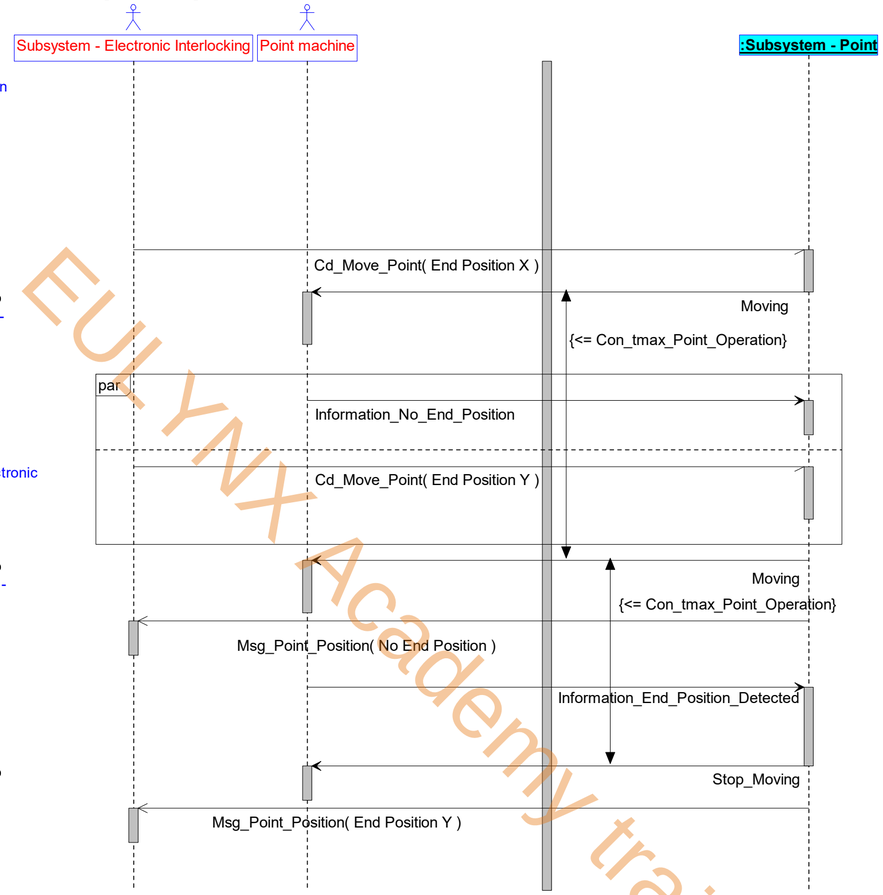
8. The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in an End position "Y".

Postcondition:

The Subsystem - Point is in an End position "Y".

ID	Type	Requirement Part 1	Requirement Part 2
Eu.P.7776	Info	Alternative Scenario: Reversing Point directly after the position has been commanded [P SD 2.1.1.3]	Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE.

P_UC2.1.1: Commanding and reversing



Alternative Scenario: Reversing Point directly after the position has been commanded [P SD 2.1.1.3]

Precondition:

The Subsystem - Point is in the state OPERATIONAL.
The Subsystem - Point is in:
- an End position "Y", or
- an Unintended position.

Interaction 2.1.1.3.A:

1. - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position "X".

2. The Subsystem - Point sends the Command to the Point machine to move the Point to an End position "X". At this moment the Subsystem - Point starts to monitor the time period Con_tmax_Point_Operation.

Interaction 2.1.1.3.B:

par

3.a1 The Subsystem - Point receives from the Point machine the Information that the Point machine is in No end position.

also par

3.b1 - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to the opposite End position "Y".

end par

4. The Subsystem - Point sends the Command to the Point machine to move the Point to an End position "Y". At this moment the Subsystem - Point restarts to monitor the time period Con_tmax_Point_Operation.

5. The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in No end position.

Interaction 2.1.1.3.C:

6. - The Subsystem - Point receives from the Point machine the Information that the Point machine is in an End position "Y". The Subsystem - Point stops to monitor the time period Con_tmax_Point_Operation.

7. The Subsystem - Point sends the Command to the Point machine to stop moving the Point machine.

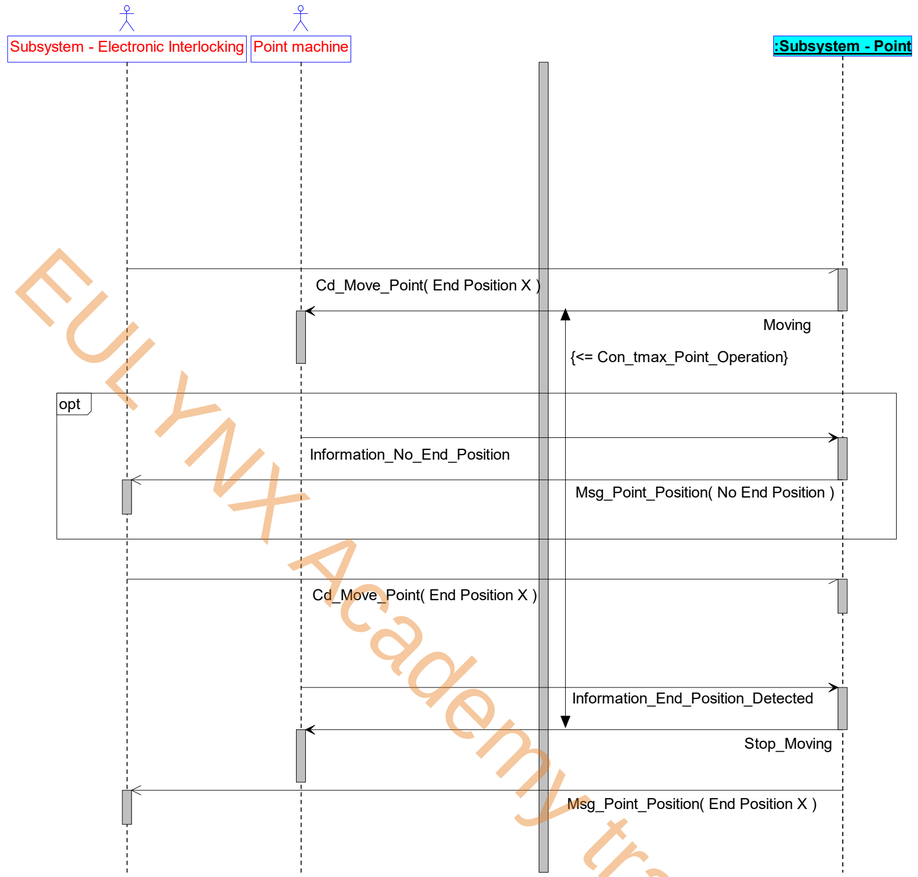
8. The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in an End position "Y".

Postcondition:

The Subsystem - Point is in an End position "Y".

ID	Type	Requirement Part 1	Requirement Part 2
Eu.P.7773	Info	Alternative Scenario: Moving of the Point with repeated command of moving #1 [P SD 2.1.1.4]	Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE.

P_UC2.1.1: Commanding and reversing



Alternative Scenario: Moving of the Point with repeated command of moving #1 [P SD 2.1.1.4]

Precondition:

The Subsystem - Point is in the state OPERATIONAL.
The Subsystem - Point is in:
- an End position "Y", or
- No end position, or
- an Unintended position.

Interaction 2.1.1.4.A:

1. - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position "X".

2. The Subsystem - Point sends the Command to the Point machine to move the Point to an End position "X". At this moment the Subsystem - Point starts to monitor the time period Con_tmax_Point_Operation.

Interaction 2.1.1.4.B:

opt [The Subsystem - Point is in an End position or a Unintended position]

3.a1 - The Subsystem - Point receives from the Point machine the Information that the Point is in No end position.

3.a2 The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in No end position.

end opt

Interaction 2.1.1.4.C:

4. - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position "X".

5. The Subsystem - Point ignores the command from the Subsystem - Electronic Interlocking.

Interaction 2.1.1.4.D:

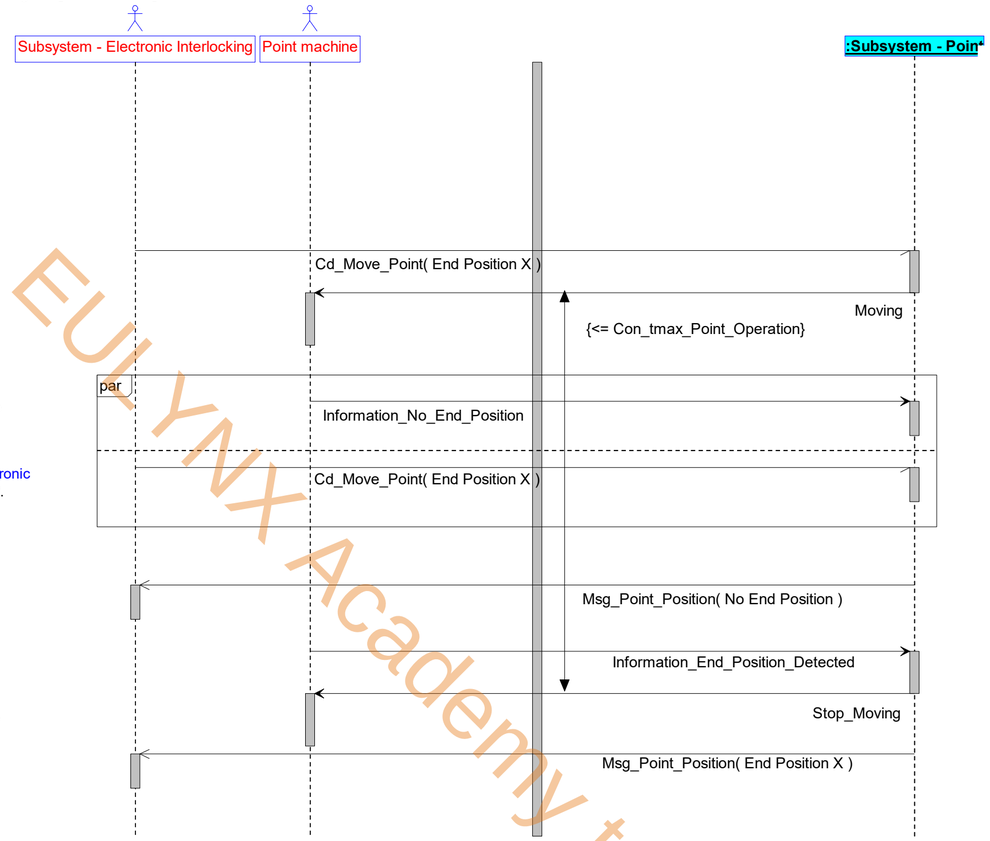
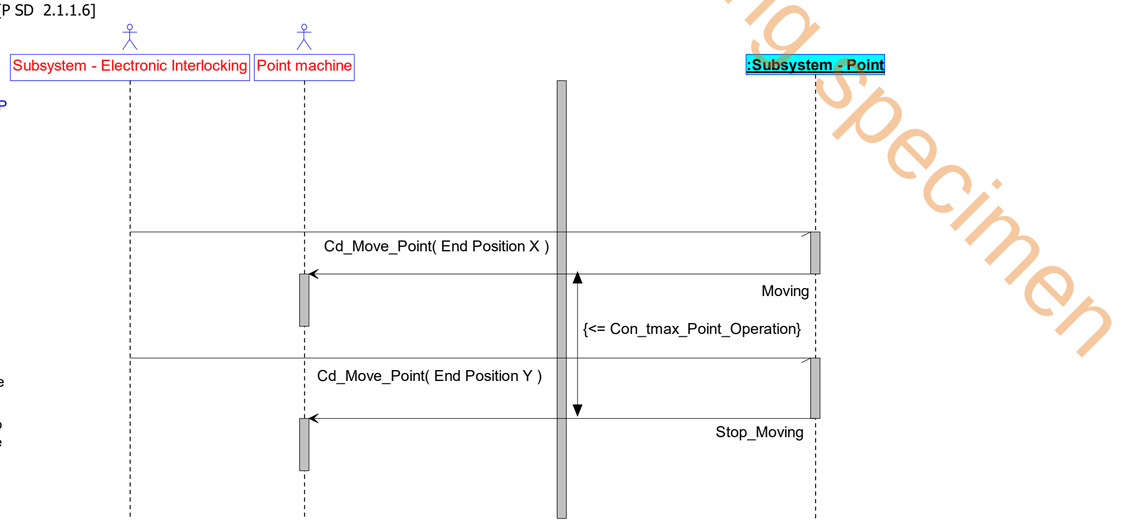
6. - The Subsystem - Point receives from the Point machine the Information that the Point is in an End position "X".

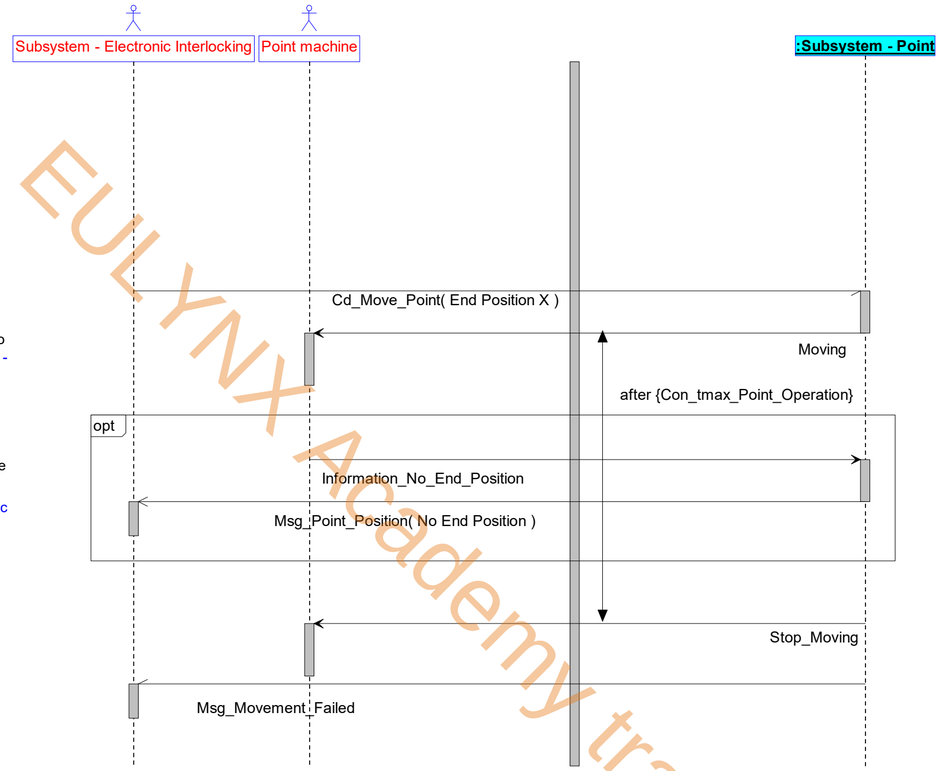
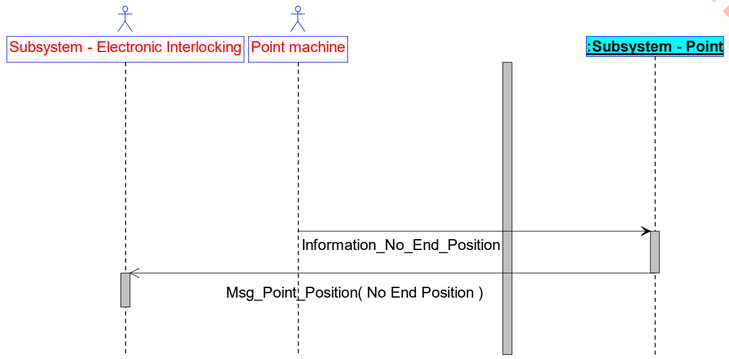
7. The Subsystem - Point sends the Command to the Point machine to stop moving the Point. The Subsystem - Point stops to monitor the time period Con_tmax_Point_Operation.

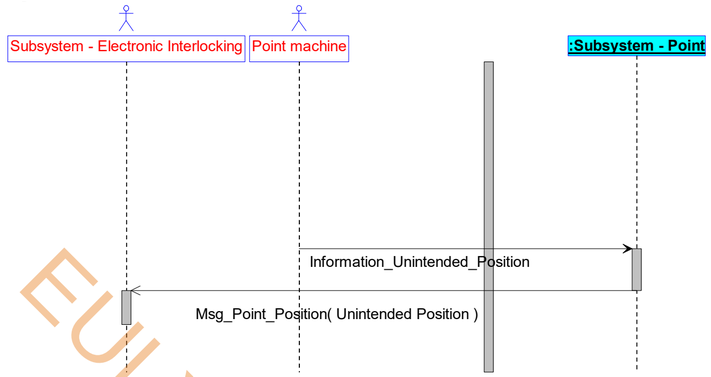
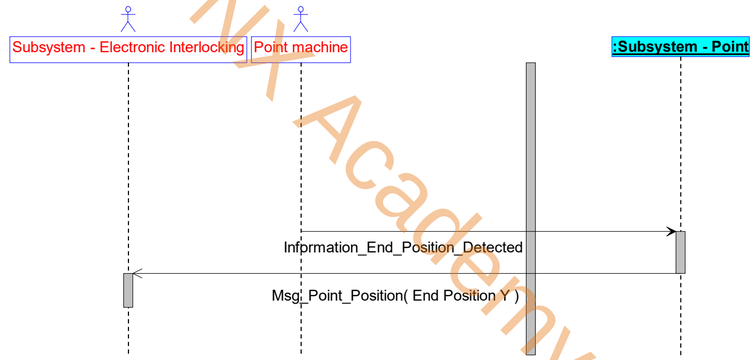
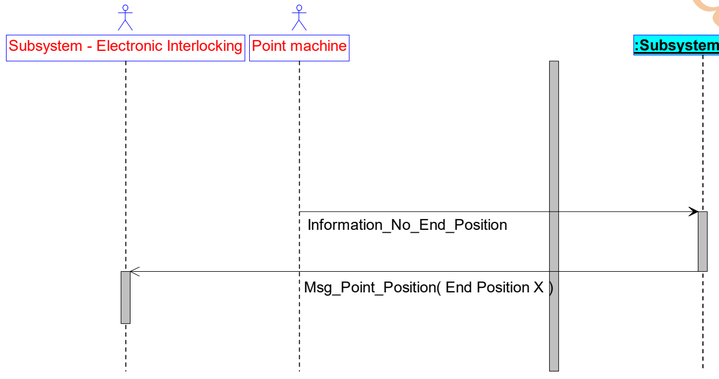
8. The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in an End position "X".

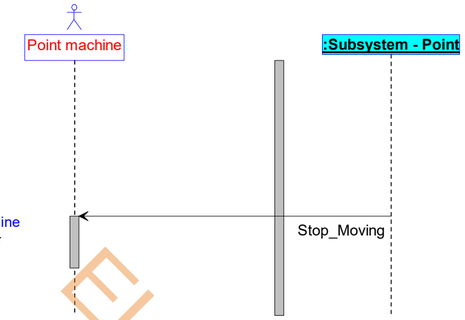
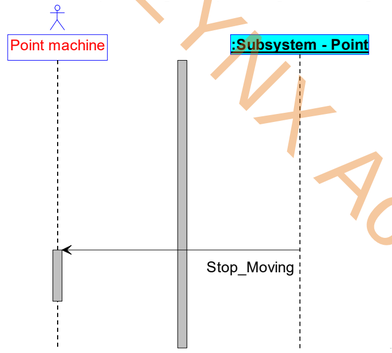
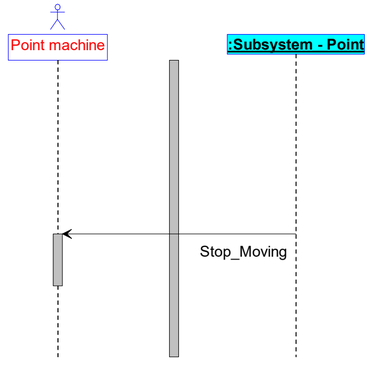
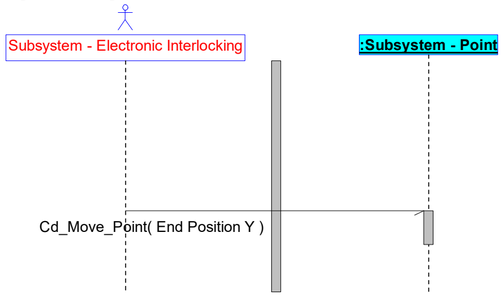
Postcondition:

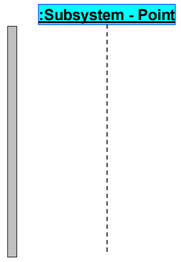
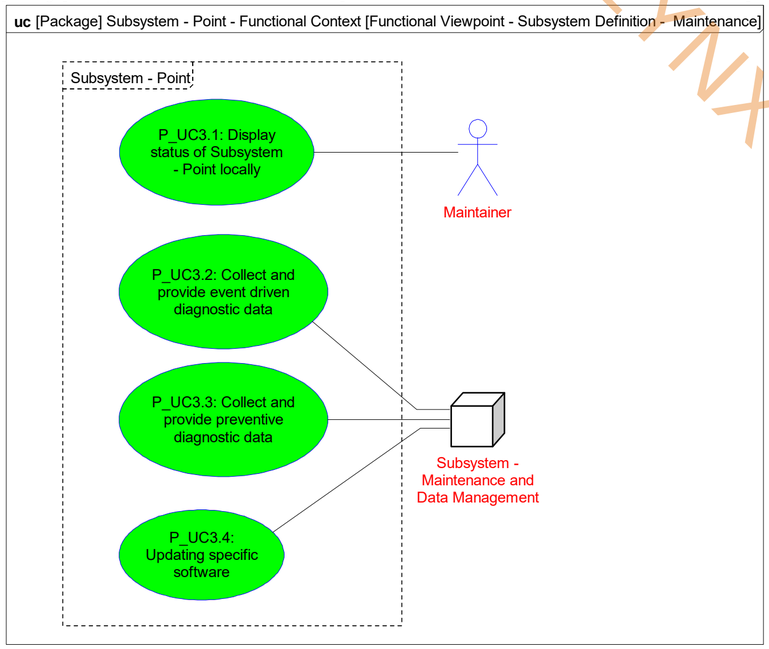
The Subsystem - Point is in an End position "X".

ID	Type	Requirement Part 1	Requirement Part 2
Eu.P.7774	Info	Alternative Scenario: Moving of the Point with repeated command of moving #2 [P SD 2.1.1.5] **P_UC2.1.1: Commanding and reversing** Alternative Scenario: Moving of the Point with repeated command of moving #2 [P SD 2.1.1.5] **Precondition:** The Subsystem - Point is in the state OPERATIONAL. The Subsystem - Point is in: - an End position "Y" or - an Unintended position. **Interaction 2.1.1.5.A:** **1.** - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position "X". **2.** The Subsystem - Point sends the Command to the Point machine to move the Point to an End position "X". At this moment the Subsystem - Point starts to monitor the time period Con_tmax_Point_Operation. **Interaction 2.1.1.5.B:** **par** **3.a1** - The Subsystem - Point receives from the Point machine the Information that the Point is in No end position. **also par** **3.b1** - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position "X". **end par** **4.** The Subsystem - Point ignores the command from the Subsystem - Electronic Interlocking. **5.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in No end position. **Interaction 2.1.1.5.C:** **6.** - The Subsystem - Point receives from the Point machine the Information that the Point is in an End position "X". **7.** The Subsystem - Point sends the Command to the Point machine to stop moving the Point. The Subsystem - Point stops to monitor the time period Con_tmax_Point_Operation. **8.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in an End position "X". **Postcondition:** The Subsystem - Point is in an End position "X". 	Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE.
Eu.P.7777	Info	Alternative Scenario: Reversing Point directly without position change [P SD 2.1.1.6] **P_UC2.1.1: Commanding and reversing** Alternative Scenario: Reversing Point directly without position change [P SD 2.1.1.6] **Precondition:** The Subsystem - Point is in the state OPERATIONAL. The Subsystem - Point is in an End position "Y". **Interaction 2.1.1.6.A:** **1.** - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position "X". **2.** The Subsystem - Point sends the Command to the Point machine to move the Point to an End position "X". At this moment the Subsystem - Point starts to monitor the time period Con_tmax_Point_Operation. **Interaction 2.1.1.6.B:** **3.** - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position "Y". The Subsystem - Point is still in an End position "Y". **4.** The Subsystem - Point sends the Command to the Point machine to stop moving the Point. The Subsystem - Point stops to monitor the time period Con_tmax_Point_Operation. **Postcondition:** The Subsystem - Point is in an End position "Y". 	Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE.

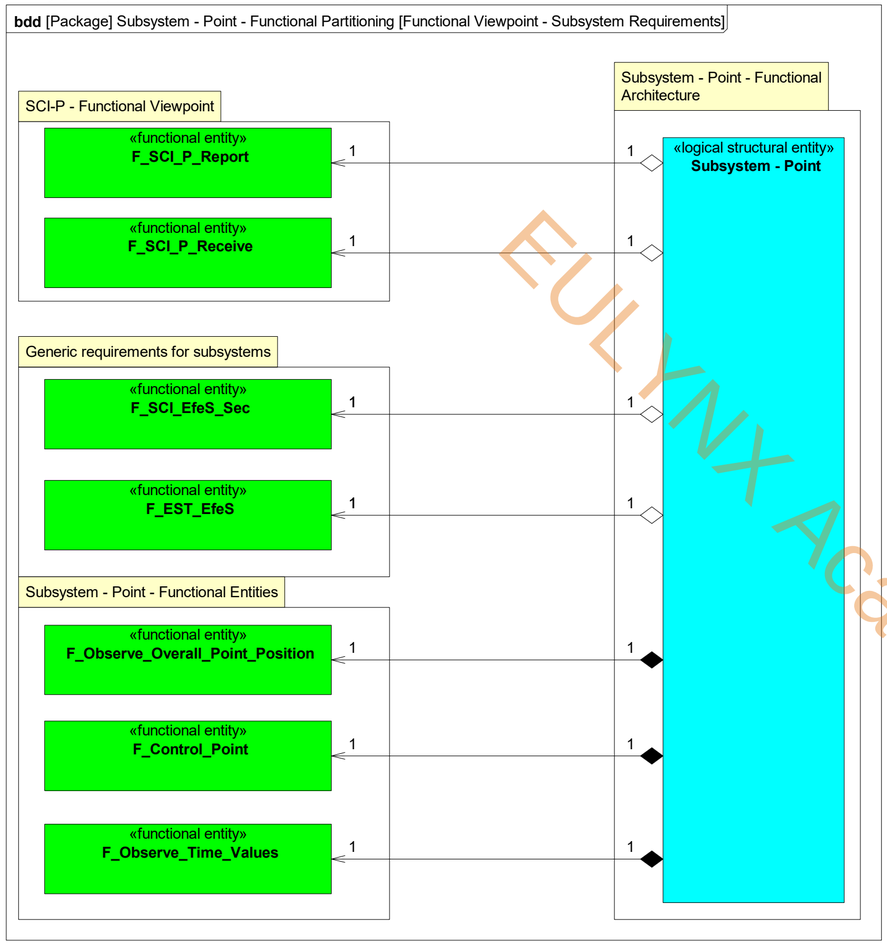
ID	Type	Requirement Part 1	Requirement Part 2
Eu.P.7779	Info	P_UC2.1.2: Irregularities	The Subsystem-UseCase "P_UC2.1.2: Irregularities" defines the behaviour of the Subsystem - Point which works with a single point machine via non-4-wire interface, when an irregularity occurs.
Eu.P.7784	Info	Alternative Scenario: Handle and report Point operation timeout with position change [P SD 2.1.2.1] **P_UC2.1.2: Irregularities** Alternative Scenario: Handle and report Point operation timeout with position change [P SD 2.1.2.1] **Precondition:** The Subsystem - Point is in the state OPERATIONAL. The Subsystem - Point is in: - an End position "Y", or - No end position, or - a Unintended position. **Interaction 2.1.2.1.A:** **1.** - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position "X". **2.** The Subsystem - Point sends the Command to the Point machine to move the Point to an End position "X". At this moment the Subsystem - Point starts to monitor the time period Con_tmax_Point_Operation. **Interaction 2.1.2.1.B:** **opt** [The Subsystem - Point was previously in an End position or a Unintended position] **3.a1** - The Subsystem - Point receives from the Point machine the Information that the Point is in No end position. **3.a2** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in No end position. **end opt** **Interaction 2.1.2.1.C:** **4.** - The Subsystem - Point detects that the time period Con_tmax_Point_Operation has exceeded and then sends the Command to the Point machine to stop moving the Point machine. **5.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that a Failed Movement has occurred. **Postcondition:** The Subsystem - Point is in No end position.  	Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE.
Eu.P.7783	Info	Alternative Scenario: Handle and report No end position [P SD 2.1.2.2] **P_UC2.1.2: Irregularities** Alternative Scenario: Handle and report No end position [P SD 2.1.2.2] **Precondition:** The Subsystem - Point is in the state OPERATIONAL. The Subsystem - Point is in: - an End position "Y", or - a Unintended position **Interaction 2.1.2.2.A:** **1.** - The Subsystem - Point receives from the Point machine the Information that the Point machine is in No end position. **2.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in No end position. **Postcondition:** The Subsystem - Point is in No end position.  	

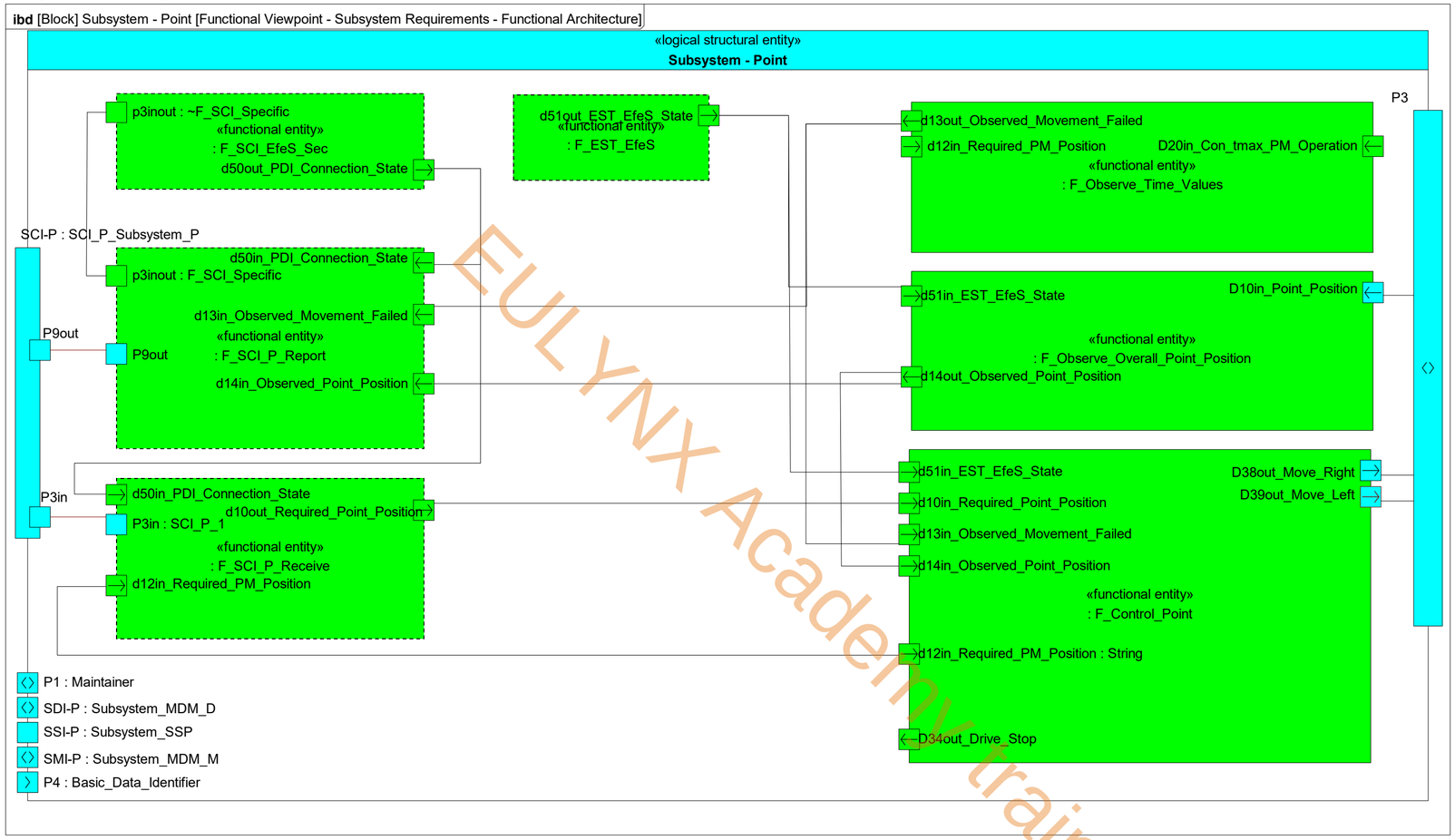
ID	Type	Requirement Part 1	Requirement Part 2
Eu.P.7785	Info	Alternative Scenario: Handle and report Unintended position [P SD 2.1.2.3] **P_UC2.1.2: Irregularities** Alternative Scenario: Handle and report Unintended position [P SD 2.1.2.3] **Precondition:** The Subsystem - Point is in the state OPERATIONAL. The Subsystem - Point is in: - an End position "Y", or - No end position. **Interaction 2.1.2.3.A:** **1.** - The Subsystem - Point receives from the Point machine the Information that the Point machine is in a Unintended position. **2.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in a Unintended position. **Postcondition:** The Subsystem - Point is in a Unintended position. 	
Eu.P.7782	Info	Alternative Scenario: Handle and report End Position[P SD 2.1.2.4] **P_UC2.1.2: Irregularities** Alternative Scenario: Handle and report End Position[P SD 2.1.2.4] **Precondition:** The Subsystem - Point is in the state OPERATIONAL. The Subsystem - Point is in: - No end position, or - a Unintended position. **Interaction 2.1.2.4.A:** **1.** - The Subsystem - Point receives from the Point machine the Information that the Point machine is in an End position. **2.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in an End position "Y". **Postcondition:** The Subsystem - Point is in an End position "Y". 	
Eu.P.7781	Info	Alternative Scenario: Handle and report End Position out of the other End Position [P SD 2.1.2.5] **P_UC2.1.2: Irregularities** Alternative Scenario: Handle and report End Position out of the other End Position [P SD 2.1.2.5] **Precondition:** The Subsystem - Point is in the state OPERATIONAL. The Subsystem - Point is in an End position "Y". **Interaction 2.1.2.5.A:** **1.** - The Subsystem - Point receives from the Point machine the Information that the Point machine is in an End position "X". **2.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in an End position "X". **Postcondition:** The Subsystem - Point is in End position "X". 	

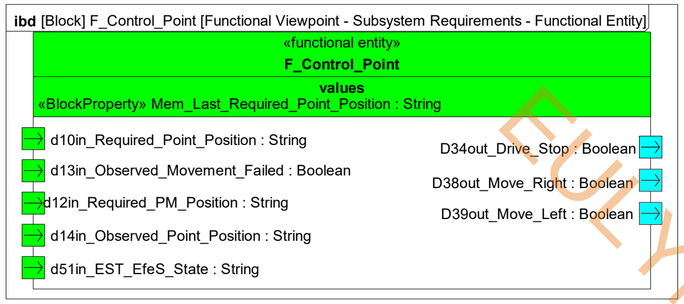
ID	Type	Requirement Part 1	Requirement Part 2
Eu.P.7787	Info	Alternative Scenario: Perform fallback operation [P SD 2.1.2.6] **P_UC2.1.2: Irregularities** Alternative Scenario: Perform fallback operation [P SD 2.1.2.6] **Precondition:** The Subsystem - Point monitors the Timevalue "Con_tmax_Point_Operation". **Interaction 2.1.2.6.A:** **1.** - The Subsystem - Point enters the state FALLBACK_MODE. **2.** The Subsystem - Point sends the Command to the Point machine to stop moving the Point. The Subsystem - Point stops to monitor the time period Con_tmax_Point_Operation. **Postcondition:** The Subsystem - Point is in the state FALLBACK_MODE.  	Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE.
Eu.P.7789	Info	Alternative Scenario: Supply voltage of the Subsystem has gone outside of the required range for operation [P SD 2.1.2.7] **P_UC2.1.2: Irregularities** Alternative Scenario: Supply voltage of the Subsystem has gone outside of the required range for operation [P SD 2.1.2.7] **Precondition:** The Subsystem - Point monitors the Timevalue "Con_tmax_Point_Operation". **Interaction 2.1.2.7.A:** **1.** - The Subsystem - Point enters the state NO_OPERATING_VOLTAGE. **2.** The Subsystem - Point sends the Command to the Point machine to stop moving the Point. The Subsystem - Point stops to monitor the time period Con_tmax_Point_Operation. **Postcondition:** The Subsystem - Point is in the state NO_OPERATING_VOLTAGE.  	Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE.
Eu.P.7788	Info	Alternative Scenario: Reset occurs [P SD 2.1.2.8] **P_UC2.1.2: Irregularities** Alternative Scenario: Reset occurs [P SD 2.1.2.8] **Precondition:** The Subsystem - Point is in the state INITIALISING or OPERATIONAL. The Subsystem - Point monitors the Timevalue "Con_tmax_Point_Operation". **Interaction 2.1.2.8.A:** **1.** - A reset has occurred. **2.** The Subsystem - Point sends the Command to the Point machine to stop moving the Point. The Subsystem - Point stops to monitor the time period Con_tmax_Point_Operation. **3.** The Subsystem - Point enters the state BOOTING. **Postcondition:** The Subsystem - Point is in the state BOOTING.  	Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE.
Eu.P.7780	Info	Alternative Scenario: Commanding of the Point to the current End position [P SD 2.1.2.9] **P_UC2.1.2: Irregularities** Alternative Scenario: Commanding of the Point to the current End position [P SD 2.1.2.9] **Precondition:** The Subsystem - Point is in the state OPERATIONAL. The Subsystem - Point is in an End position "Y". **Interaction 2.1.2.9.A:** **1.** - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position "Y". **Postcondition:** ---  	

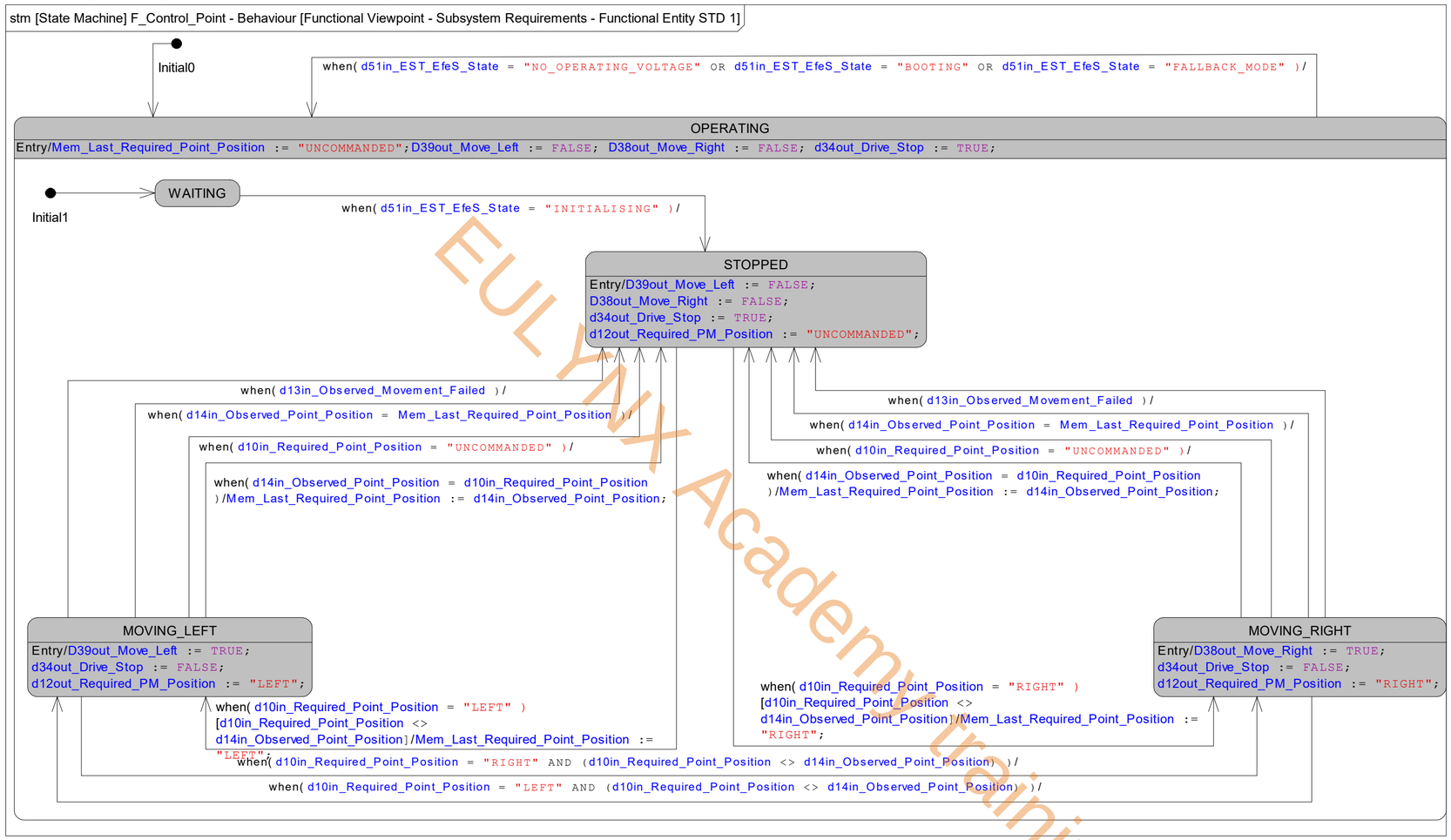
ID	Type	Requirement Part 1	Requirement Part 2
Eu.P.7786	Info	Alternative Scenario: Handling of interrupted PDI connection [P SD 2.1.2.10] **P_UC2.1.2: Irregularities**  Alternative Scenario: Handling of interrupted PDI connection [P SD 2.1.2.10] **Precondition:** The Subsystem - Point is in the state OPERATIONAL. **Interaction 2.1.2.10.A:** **1.** - The PDI connection has been terminated. **Postcondition:** The Subsystem - Point is in the state INITIALISING. The Process Data Interface protocol connection is terminated.	Note: If the PDI Connection is terminated during ongoing point movement, the movement is continued as normal.
Eu.P.7764	Info	[Package] Subsystem - Point - Functional Context [Functional Viewpoint - Subsystem Definition - Maintenance]  	
Eu.P.7791	Info	P_UC3.1: Display status of Subsystem - Point locally	Information: The Subsystem-UseCase P_UC3.1: Display status of Subsystem - Point locally defines the local display of the EULYNX field element Subsystem. See ID Eu.P.890.
Eu.P.7792	Info	P_UC3.2: Collect and provide event driven diagnostic data	Information: The Subsystem-UseCase P_UC3.2: Collect and provide event driven diagnostic data defines the event driven collection and provision of diagnostic data in case of irregularities. See ID Eu.P.925.
Eu.P.7793	Info	P_UC3.3: Collect and provide preventive diagnostic data	Information: The Subsystem-UseCase P_UC3.3: Collect and provide preventive diagnostic data defines the continuous collection and provision of diagnostic data for preventive maintenance. See ID Eu.P.925.
Eu.P.7794	Info	P_UC3.4: Updating specific software	Information: The Subsystem-UseCase P_UC3.4: Updating specific software defines the process of updating the specific software between Subsystem - Maintenance and Data Management and the Subsystem.
Eu.P.7795	Info	The generic UseCases SCI-XX EfeS IFUC1.1: Establish PDI connection and SCI-XX EfeS IFUC1.2: Close PDI connection are specified in [Eu.Doc.119]. The generic UseCases SMI-XX IFUC 1.1: Establish SMI connection, SMI-XX IFUC 1.2: Synchronous loading and activation of data, SMI-XX IFUC 1.3: Asynchronous preloading of data, SMI-XX IFUC 1.4: Reset EfeS and SMI-XX IFUC 1.5: Initiate maintenance are specified in [Eu.Doc.120].	

ID	Type	Requirement Part 1	Requirement Part 2
Eu.P.7887	Head	**3.3.3 Subsystem - Point - Functional Partitioning**	
Eu.P.7888	Def	[Package] Subsystem - Point - Functional Partitioning [Functional Viewpoint - Subsystem Requirements]	
Eu.P.7753	Head	**3.3.4 Subsystem - Point - Functional Architecture**	
Eu.P.7754	Info	Subsystem - Point	The Subsystem - Point integrates the moveable elements, that may be moved to a different position by a request from the Subsystem - Electronic Interlocking.



ID	Type	Requirement Part 1	Requirement Part 2
Eu.P.7755	Def	[Block] Subsystem - Point [Functional Viewpoint - Subsystem Requirements - Functional Architecture] 	The Subsystem - Point shall provide the functional interfaces shown in "[Block] Subsystem - Point [Functional Viewpoint - Subsystem Requirements - Functional Architecture]", typed by FlowSpecifications. Each FlowSpecification is defined by a set of FlowProperties that specify the possible exchange of information through the particular interface.
Eu.P.7756	Def	P1	The functional Maintenance/Operation/Display interface to the Maintainer. The InformationFlow through the interface is defined by "Maintainer".
Eu.P.7757	Def	P3	The functional control interface to Point machines for the information flow through the interface, which is defined by the FlowSpecification "Point_machine".
Eu.P.7758	Def	P4	The functional System Data interface to the Basic Data identifier. The InformationFlow through the interface is defined by "Basic_Data_Identifier".
Eu.P.7759	Def	SCI-P	The functional Process Data interface to the Subsystem - Electronic Interlocking (SCI: Standard Communication Interface). The InformationFlow through the interface is further defined in SCI_P_Subsystem_EIL.
Eu.P.7760	Def	SDI-P	The functional Diagnostic interface to the Subsystem - Maintenance and Data Management for the InformationFlow through the interface, which is defined by "Subsystem_MDM_D".
Eu.P.7761	Def	SMI-P	The functional Maintenance Interface to the Subsystem - Maintenance and Data Management for the InformationFlow through the interface, which is defined by "Subsystem_MDM_M".

ID	Type	Requirement Part 1	Requirement Part 2
Eu.P.7762	Def	SSI-P	The Security Service Interface to the Subsystem - Security Services Platform. The InformationFlow through the interface is further defined in SSI-P (Subsystem - Security Services Platform).
Eu.P.7796	Head	**3.3.5 Subsystem - Point - Functional Entities**	
Eu.P.7797	Info	F_Control_Point	
Eu.P.7798	Req	[Block] F_Control_Point [Functional Viewpoint - Subsystem Requirements - Functional Entity] 	
Eu.P.7799	Def	d10in_Required_Point_Position	
Eu.P.7800	Def	d12in_Required_PM_Position	
Eu.P.7801	Def	d13in_Observed_Movement_Failed	
Eu.P.7802	Def	d14in_Observed_Point_Position	The port d14in_Observed_Point_Position provides the Point machine position to the Subsystem - Point. The port d14in_Observed_Point_Position refines the InformationFlow Information_No_End_Position, Information_End_Position_Reached, Information_End_Position_Detected and Information_Unintended_Position.
Eu.P.7803	Def	D34out_Drive_Stop	
Eu.P.7804	Def	D38out_Move_Right	The port D38out_Move_Right provides the command Moving right to the point machine and refines the InformationFlow Moving and Stop moving. The following values are valid: "True" (driving right), "False" (not driving right, Stop moving).
Eu.P.7805	Def	D39out_Move_Left	The port D39out_Move_Left refines the Informationflow Moving and contains the information if driving left was started. The following values are valid: "True" (driving left), "False" (not driving left)
Eu.P.7806	Def	d51in_EST_EfeS_State	
Eu.P.7807	Info	F_Control_Point - Behaviour	

ID	Type	Requirement Part 1	Requirement Part 2
Eu.P.7808	Req	Functional Viewpoint - Subsystem Requirements - Functional Entity STD 1 	This state machine diagram describes the requirements for the following functionalities: - observes and forward the required PM position
Eu.P.7809	Def	Initial0	
Eu.P.7810	Def	/{Initial0 - OPERATING}	
Eu.P.7811	Def	OPERATING	
Eu.P.7812	Def	Initial1	
Eu.P.7813	Def	/{Initial1 - WAITING}	
Eu.P.7814	Def	MOVING_LEFT	

Eu.P.7815	Def	entry/D39out_Move_Left := TRUE;
		d34out_Drive_Stop := FALSE;
		d12out_Required_PM_Position := "LEFT";{State-internal in MOVING_LEFT}	

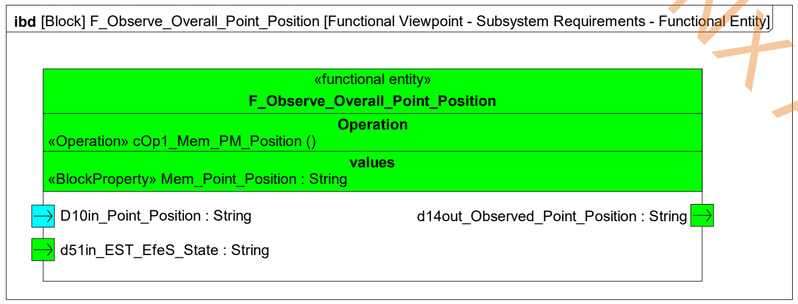
Eu.P.7816	Def	when(d10in_Required_Point_Position = "RIGHT" AND (d10in_Required_Point_Position <> d14in_Observed_Point_Position))/{MOVING_LEFT - MOVING_RIGHT}	
Eu.P.7817	Def	when(d10in_Required_Point_Position = "UNCOMMANDED")/{MOVING_LEFT - STOPPED}	
Eu.P.7818	Def	when(d13in_Observed_Movement_Failed)/{MOVING_LEFT - STOPPED}	
Eu.P.7819	Def	when(d14in_Observed_Point_Position = d10in_Required_Point_Position)/Mem_Last_Required_Point_Position := d14in_Observed_Point_Position;{MOVING_LEFT - STOPPED}	
Eu.P.7820	Def	when(d14in_Observed_Point_Position = Mem_Last_Required_Point_Position)/{MOVING_LEFT - STOPPED}	
Eu.P.7821	Def	MOVING_RIGHT	

Eu.P.7822	Def	entry/D38out_Move_Right := TRUE;
		d34out_Drive_Stop := FALSE;
		d12out_Required_PM_Position := "RIGHT";{State-internal in MOVING_RIGHT}	

Eu.P.7823	Def	when(d10in_Required_Point_Position = "LEFT" AND (d10in_Required_Point_Position <> d14in_Observed_Point_Position))/{MOVING_RIGHT - MOVING_LEFT}	
Eu.P.7824	Def	when(d10in_Required_Point_Position = "UNCOMMANDED")/{MOVING_RIGHT - STOPPED}	

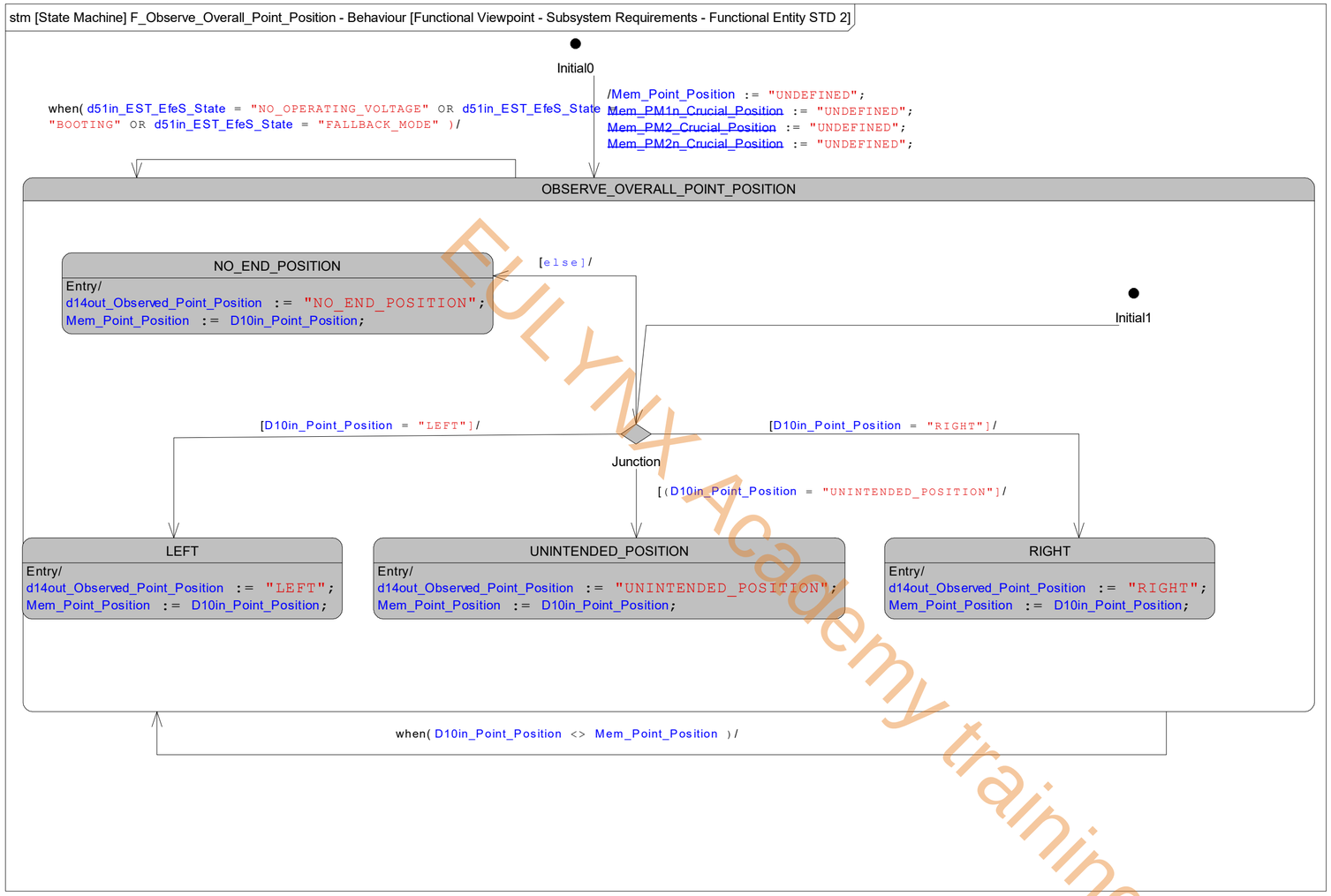
ID	Type	Requirement Part 1	Requirement Part 2
Eu.P.7825	Def	when(d13in_Observed_Movement_Failed)/{MOVING_RIGHT - STOPPED}	
Eu.P.7826	Def	when(d14in_Observed_Point_Position = d10in_Required_Point_Position)/Mem_Last_Required_Point_Position := d14in_Observed_Point_Position;{MOVING_RIGHT - STOPPED}	
Eu.P.7827	Def	when(d14in_Observed_Point_Position = Mem_Last_Required_Point_Position)/{MOVING_RIGHT - STOPPED}	
Eu.P.7828	Def	STOPPED	

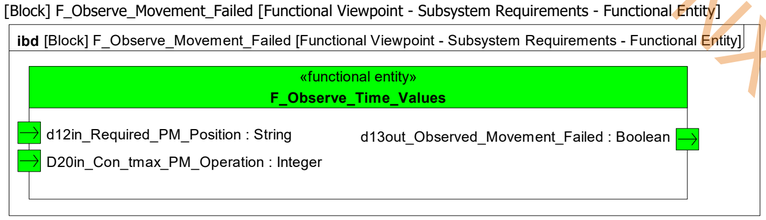
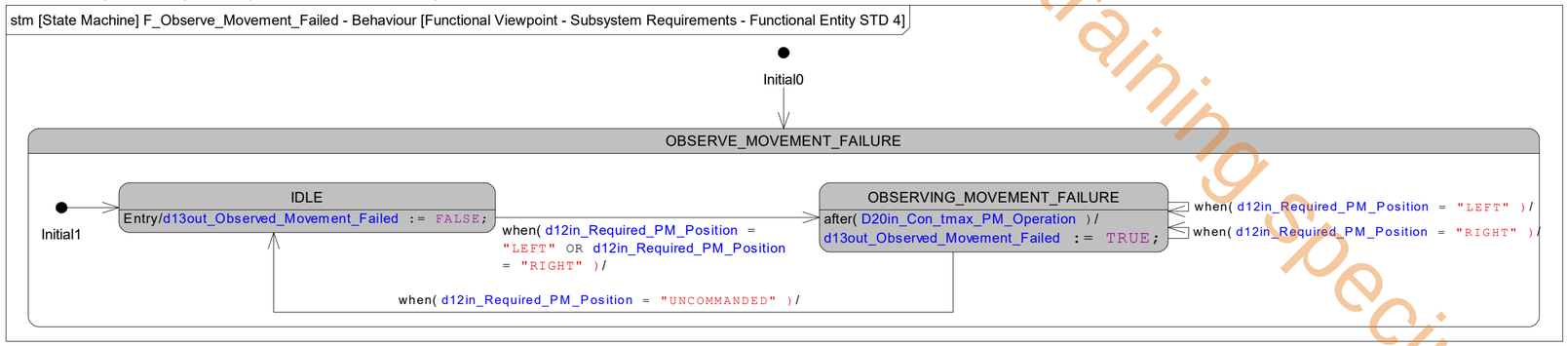
Eu.P.7829	Def	entry/D39out_Move_Left := FALSE;
		D38out_Move_Right := FALSE;
		d34out_Drive_Stop := TRUE;
		d12out_Required_PM_Position := "UNCOMMANDED";{State-internal in STOPPED}	

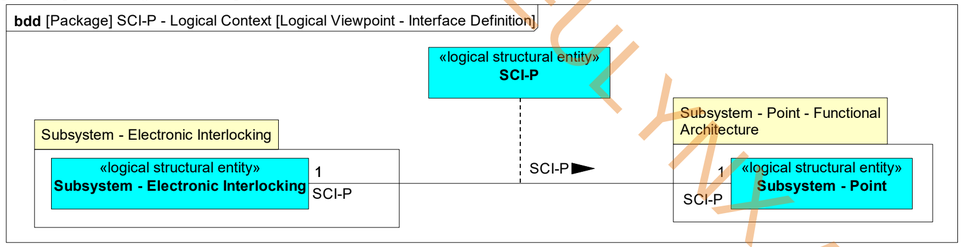
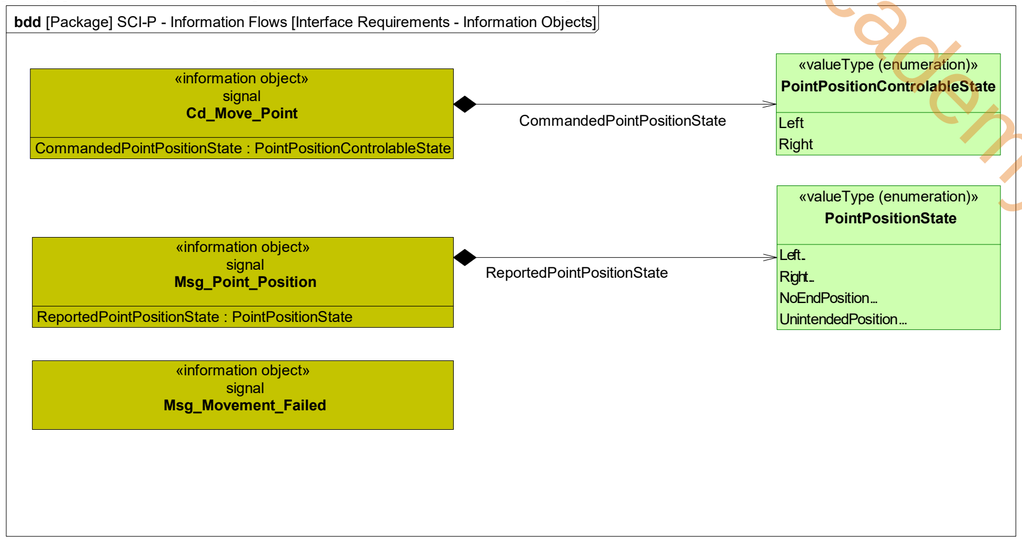
Eu.P.7830	Def	when(d10in_Required_Point_Position = "LEFT")[d10in_Required_Point_Position <> d14in_Observed_Point_Position]/Mem_Last_Required_Point_Position := "LEFT";{STOPPED - MOVING_LEFT}	
Eu.P.7831	Def	when(d10in_Required_Point_Position = "RIGHT")[d10in_Required_Point_Position <> d14in_Observed_Point_Position]/Mem_Last_Required_Point_Position := "RIGHT";{STOPPED - MOVING_RIGHT}	
Eu.P.7832	Def	entry/Mem_Last_Required_Point_Position := "UNCOMMANDED";D39out_Move_Left := FALSE; D38out_Move_Right := FALSE; d34out_Drive_Stop := TRUE;{State-internal in OPERATING}	
Eu.P.7833	Def	WAITING	
Eu.P.7834	Def	when(d51in_EST_EfeS_State = "INITIALISING")/{WAITING - STOPPED}	
Eu.P.7835	Def	when(d51in_EST_EfeS_State = "NO_OPERATING_VOLTAGE" OR d51in_EST_EfeS_State = "BOOTING" OR d51in_EST_EfeS_State = "FALLBACK_MODE")/{OPERATING - OPERATING}	
Eu.P.7836	Info	F_Observe_Overall_Point_Position	
Eu.P.7837	Req	[Block] F_Observe_Overall_Point_Position [Functional Viewpoint - Subsystem Requirements - Functional Entity] 	

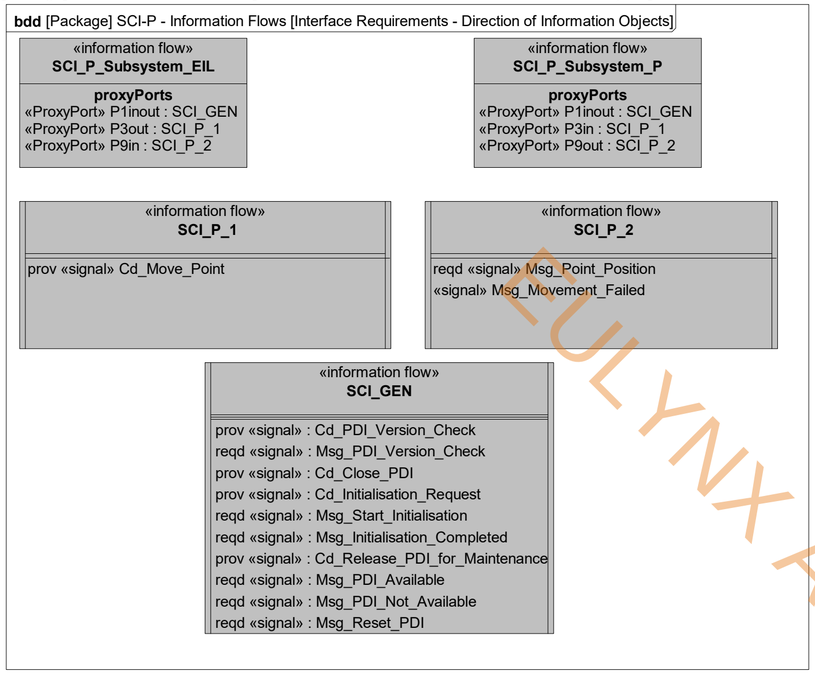
Eu.P.7838	Def	/* cOp1_Mem_PM_Position */
		
		if D26in_Con_PM1n_Crucial_Activation then
		Mem_PM1n_Crucial_Position := D23in_PM1n_Crucial_Position;
		end if
		if D27in_Con_PM2_Non_Crucial_Activation then
		Mem_PM2_Non_Crucial_Position := D24in_PM2_Non_Crucial_Position;
		end if
		if D28in_Con_PM2n_Non_Crucial_Activation then
		Mem_PM2n_Non_Crucial_Position := D25in_PM2n_Non_Crucial_Position;
		end if
			cOp1_Mem_PM_Position

Eu.P.7839	Def	D10in_Point_Position	The port D10in_Point_Position represents the Point position that was observed at the interface of the first crucial Point machine. The Point position is derived from the InformationFlow Information_No_End_Position, Information_End_Position_Reached, Information_End_Position_Detected and Information_Unintended_Position.
Eu.P.7840	Def	d14out_Observed_Point_Position	
Eu.P.7841	Def	d51in_EST_EfeS_State	
Eu.P.7842	Info	F_Observe_Overall_Point_Position - Behaviour	

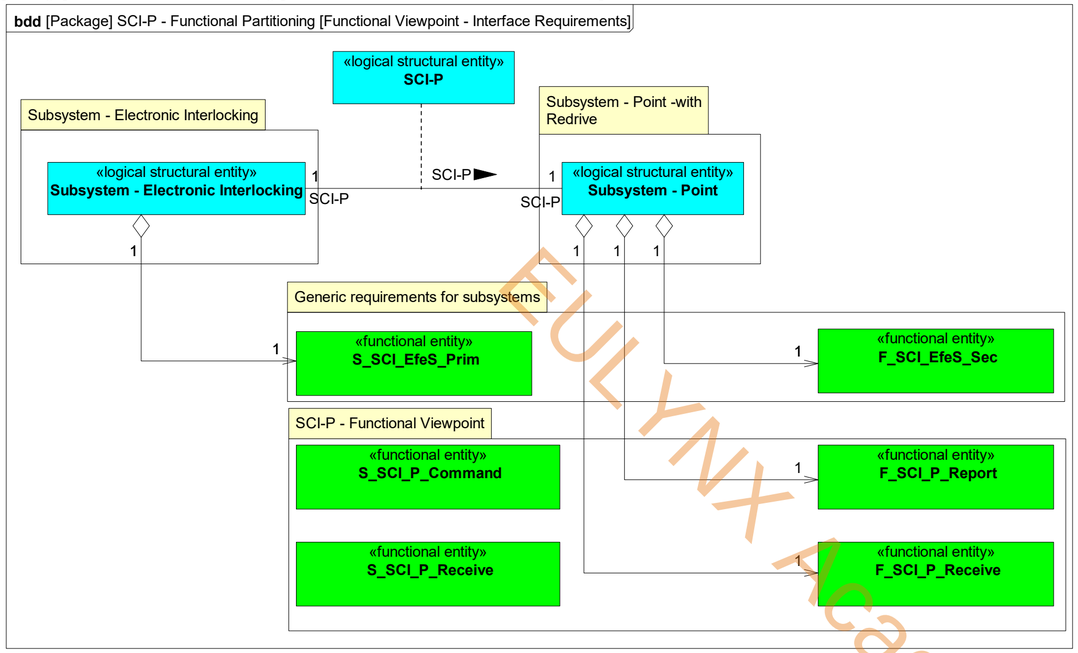
ID	Type	Requirement Part 1	Requirement Part 2
Eu.P.7843	Req	Functional Viewpoint - Subsystem Requirements - Functional Entity STD 2 	This state machine diagram describes the requirements for the following functionalities: - observes and forward the overall point position state
Eu.P.7844	Def	Initial0	
Eu.P.7845	Def	/Mem_Point_Position := "UNDEFINED"; Mem_PM1n_Crucial_Position := "UNDEFINED"; Mem_PM2_Crucial_Position := "UNDEFINED"; Mem_PM2n_Crucial_Position := "UNDEFINED";{Initial0 - OBSERVE_OVERALL_POINT_POSITION}	
Eu.P.7846	Def	LEFT	
Eu.P.7847	Def	entry/ d14out_Observed_Point_Position := "LEFT"; Mem_Point_Position := D10in_Point_Position;{State-internal in LEFT}	
Eu.P.7848	Def	OBSERVE_OVERALL_POINT_POSITION	
Eu.P.7849	Def	Initial1	
Eu.P.7850	Def	/{Initial1 - Junction}	
Eu.P.7851	Def	Junction	
Eu.P.7852	Def	[D10in_Point_Position = "LEFT"]/{Junction - LEFT}	
Eu.P.7853	Def	[else]/{Junction - NO_END_POSITION}	
Eu.P.7854	Def	[D10in_Point_Position = "RIGHT"]/{Junction - RIGHT}	
Eu.P.7855	Def	[(D10in_Point_Position = "UNINTENDED_POSITION"]/{Junction - UNINTENDED_POSITION}	
Eu.P.7856	Def	NO_END_POSITION	

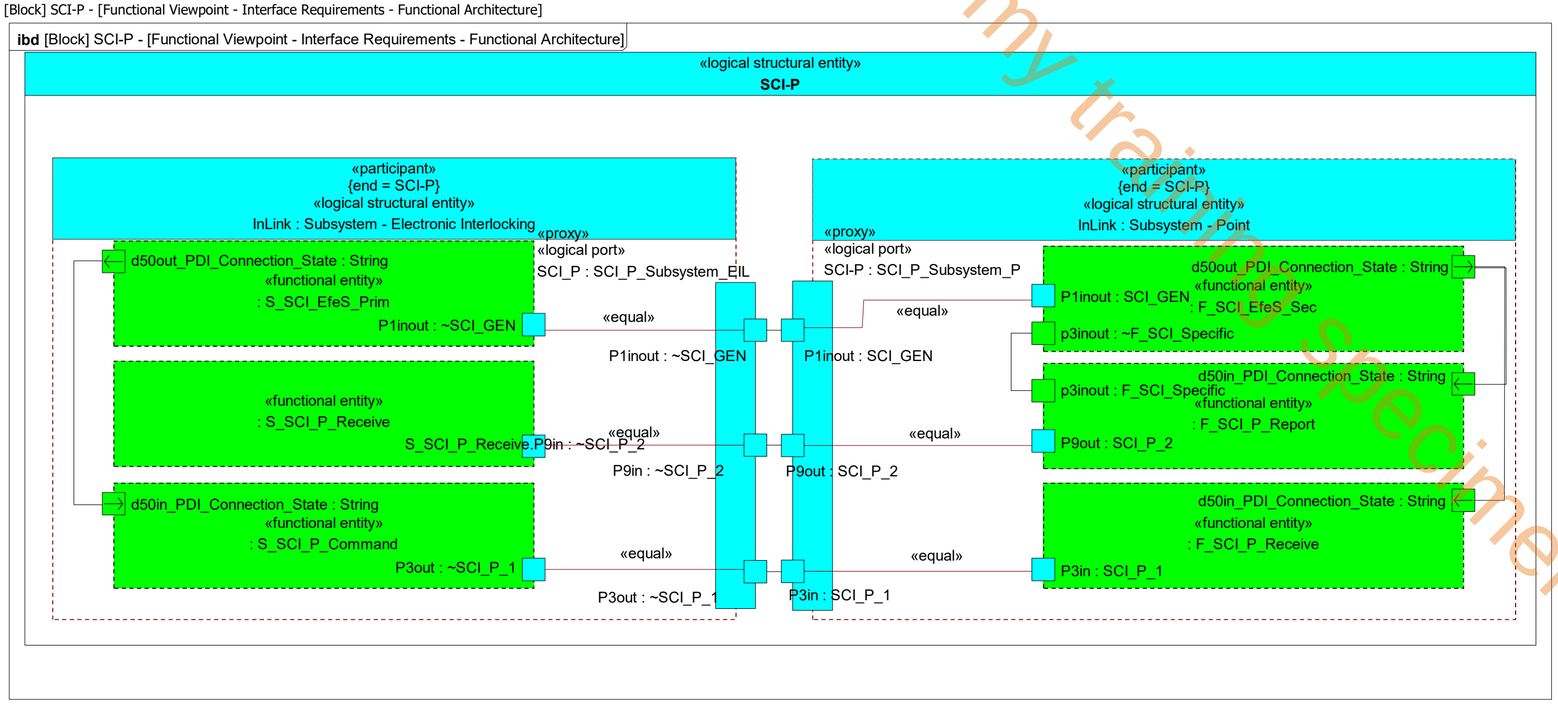
ID	Type	Requirement Part 1	Requirement Part 2
Eu.P.7857	Def	entry/ d14out_Observed_Point_Position := "NO_END_POSITION"; Mem_Point_Position := D10in_Point_Position;{State-internal in NO_END_POSITION}	
Eu.P.7858	Def	RIGHT	
Eu.P.7859	Def	entry/ d14out_Observed_Point_Position := "RIGHT"; Mem_Point_Position := D10in_Point_Position;{State-internal in RIGHT}	
Eu.P.7860	Def	UNINTENDED_POSITION	
Eu.P.7861	Def	entry/ d14out_Observed_Point_Position := "UNINTENDED_POSITION"; Mem_Point_Position := D10in_Point_Position;{State-internal in UNINTENDED_POSITION}	
Eu.P.7862	Def	when(D10in_PM1n_Crucial_Position <> Mem_PM1n_Crucial_Position)/{OBSERVE_OVERALL_POINT_POSITION - OBSERVE_OVERALL_POINT_POSITION}	
Eu.P.7863	Def	when(D10in_PM2_Non_Crucial_Position <> Mem_PM2_Non_Crucial_Position)/{OBSERVE_OVERALL_POINT_POSITION - OBSERVE_OVERALL_POINT_POSITION}	
Eu.P.7864	Def	when(D10in_PM2n_Non_Crucial_Position <> Mem_PM2n_Non_Crucial_Position)/{OBSERVE_OVERALL_POINT_POSITION - OBSERVE_OVERALL_POINT_POSITION}	
Eu.P.7865	Def	when(D10in_Point_Position <> Mem_Point_Position)/{OBSERVE_OVERALL_POINT_POSITION - OBSERVE_OVERALL_POINT_POSITION}	
Eu.P.7866	Def	when(d51in_EST_EfeS_State = "NO_OPERATING_VOLTAGE" OR d51in_EST_EfeS_State = "BOOTING" OR d51in_EST_EfeS_State = "FALLBACK_MODE")/{OBSERVE_OVERALL_POINT_POSITION - OBSERVE_OVERALL_POINT_POSITION}	
Eu.P.7867	Info	F_Observe_Time_Values	
Eu.P.7868	Req	[Block] F_Observe_Movement_Failed [Functional Viewpoint - Subsystem Requirements - Functional Entity] 	
Eu.P.7869	Def	d12in_Required_PM_Position	
Eu.P.7870	Def	d13out_Observed_Movement_Failed	
Eu.P.7871	Def	D20in_Con_tmax_PM_Operation	The port D20in_Con_tmax_PM_Operation refines the time value Con_tmax_Point_Operation.
Eu.P.7872	Info	F_Observe_Movement_Failed - Behaviour	
Eu.P.7873	Req	Functional Viewpoint - Subsystem Requirements - Functional Entity STD 4 	This state machine diagram describes the requirements for the following functionalities: - monitors the Con_tmax_PM_Operation and forwards an overrun as a Movent Failure - observes and forward a Movement failure
Eu.P.7874	Def	Initial0	
Eu.P.7875	Def	/{Initial0 - OBSERVE_MOVEMENT_FAILURE}	
Eu.P.7876	Def	OBSERVE_MOVEMENT_FAILURE	
Eu.P.7877	Def	IDLE	
Eu.P.7878	Def	entry/d13out_Observed_Movement_Failed := FALSE;{State-internal in IDLE}	
Eu.P.7879	Def	when(d12in_Required_PM_Position = "LEFT" OR d12in_Required_PM_Position = "RIGHT")/{IDLE - OBSERVING_MOVEMENT_FAILURE}	
Eu.P.7880	Def	Initial1	
Eu.P.7881	Def	/{Initial1 - IDLE}	
Eu.P.7882	Def	OBSERVING_MOVEMENT_FAILURE	

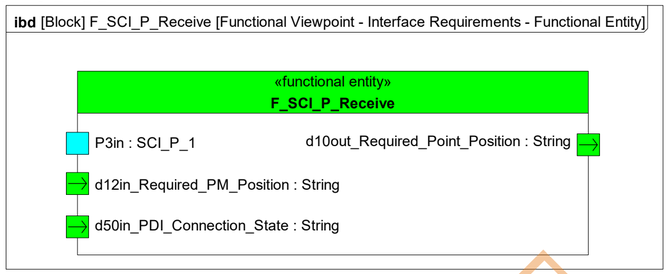
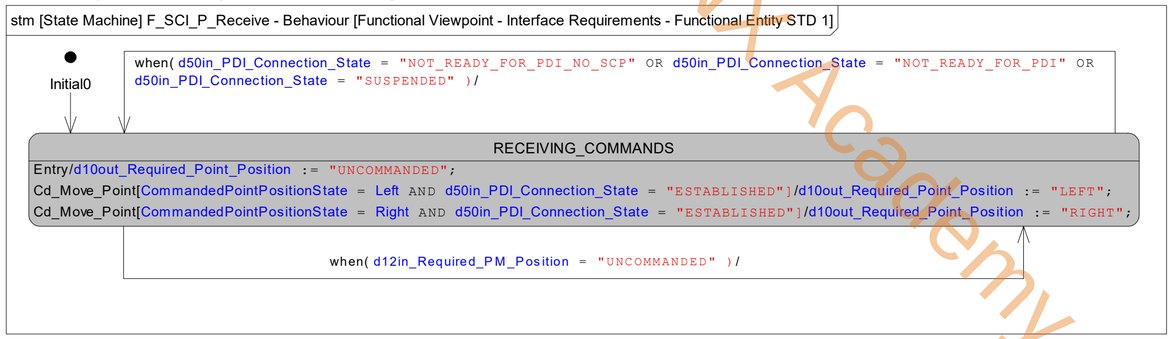
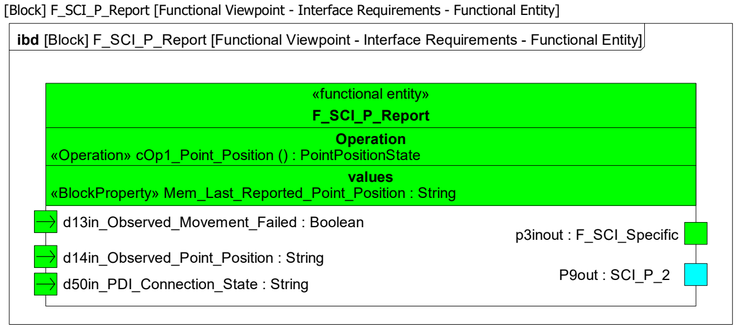
ID	Type	Requirement Part 1	Requirement Part 2
Eu.P.7883	Def	after(D20in_Con_tmax_PM_Operation)/ d13out_Observed_Movement_Failed := TRUE;{State-internal in OBSERVING_MOVEMENT_FAILURE}	
Eu.P.7884	Def	when(d12in_Required_PM_Position = "LEFT")/{OBSERVING_MOVEMENT_FAILURE - OBSERVING_MOVEMENT_FAILURE}	
Eu.P.7885	Def	when(d12in_Required_PM_Position = "RIGHT")/{OBSERVING_MOVEMENT_FAILURE - OBSERVING_MOVEMENT_FAILURE}	
Eu.P.7886	Def	when(d12in_Required_PM_Position = "UNCOMMANDED")/{OBSERVING_MOVEMENT_FAILURE - IDLE}	
Eu.P.7890	Head	**3.4 Subsystem - Point - Interfaces**	
Eu.P.7909	Head	**3.4.1 SCI-P (Subsystem - Electronic Interlocking)**	
Eu.P.8023	Head	**3.4.1.1 SCI-P - Logical Viewpoint**	
Eu.P.8024	Head	**3.4.1.1.1 SCI-P - Logical Context**	
Eu.P.8025	Def	[Package] SCI-P - Logical Context [Logical Viewpoint - Interface Definition]  	
Eu.P.8008	Head	**3.4.1.2 SCI-P - Information Flows**	
Eu.P.8010	Def	[Package] SCI-P - Information Flows [Interface Requirements - Information Objects]  	

ID	Type	Requirement Part 1	Requirement Part 2
Eu.P.8009	Def	[Package] SCI-P - Information Flows [Interface Requirements - Direction of Information Objects]  	
Eu.P.8011	Def	Cd_Move_Point	Command (Cd) from Subsystem - Electronic Interlocking to Subsystem - Point to move the Point into the commanded position.
Eu.P.8012	Def	Msg_Movement_Failed	Message (Msg) from Subsystem - Point to Subsystem - Electronic Interlocking that the ongoing moving failed. The Subsystem - Point has abandoned the moving for one of the following reasons: - The time period Con_tmax_Point_Operation has expired. - The point movement has failed. Note: The conditions defining a failed moment can be supplier and IM specific. They may include, for example, a failure of the star point buildup or a situation in which only non-driven point machines have not yet reached the commanded end position.
Eu.P.8013	Def	Msg_Point_Position	Message (Msg) from Subsystem - Point to Subsystem - Electronic Interlocking about the current Point position.
Eu.P.8014	Info	The generic commands and messages through the SCI_P_Subsystem_EIL are specified in [Eu.Doc.119].	
Eu.P.7910	Head	**3.4.1.3 SCI-P - Functional Viewpoint**	
Eu.P.8006	Head	**3.4.1.3.1 SCI-P - Functional Partitioning**	

ID	Type	Requirement Part 1	Requirement Part 2
Eu.P.8007	Def	[Package] SCI-P - Functional Partitioning [Functional Viewpoint - Interface Requirements]	
Eu.P.7911	Head	**3.4.1.3.2 SCI-P - Functional Architecture**	
Eu.P.7912	Info	SCI-P	
Eu.P.7913	Def	[Block] SCI-P - [Functional Viewpoint - Interface Requirements - Functional Architecture]	
Eu.P.7914	Head	**3.4.1.3.3 SCI-P - Functional Entities**	
Eu.P.7915	Info	F_SCI_P_Receive	

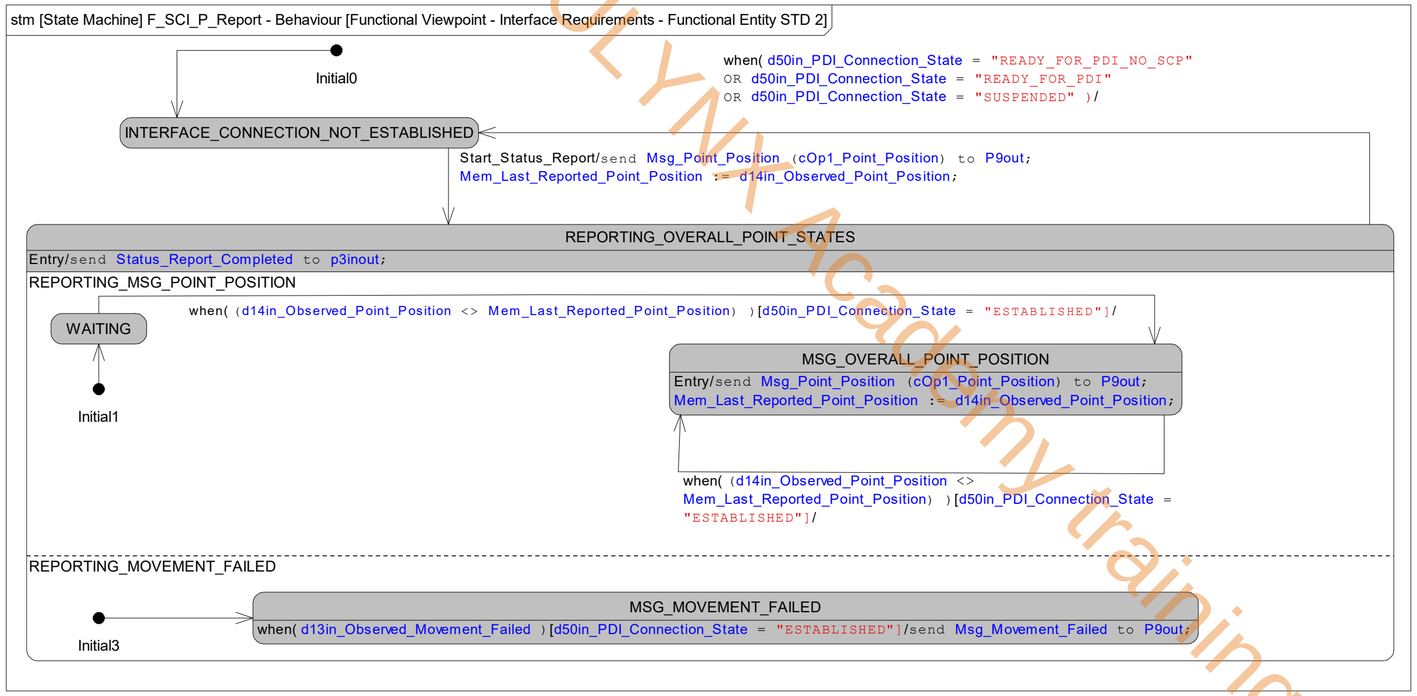


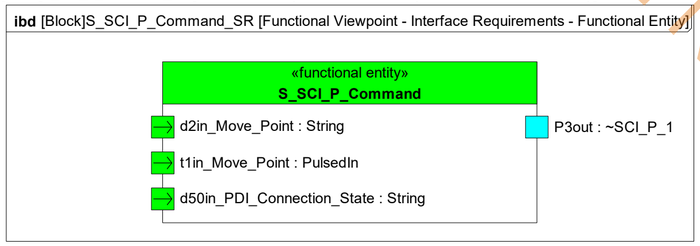
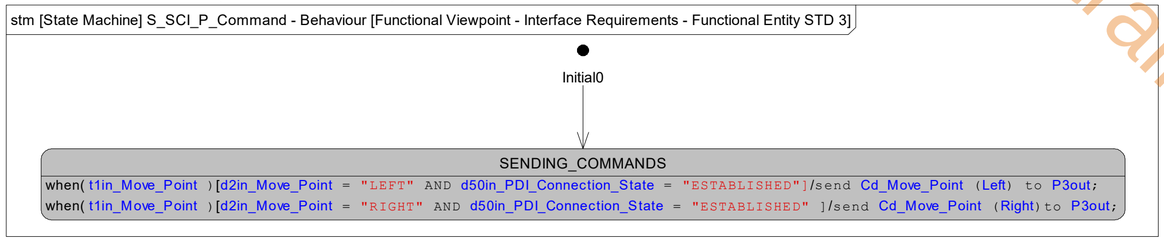


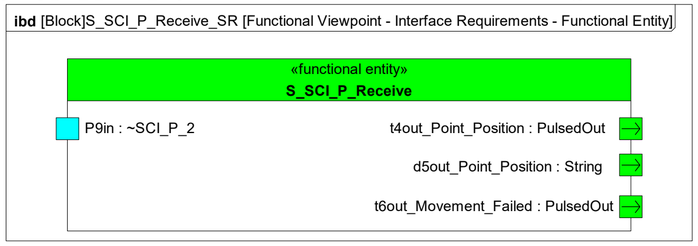
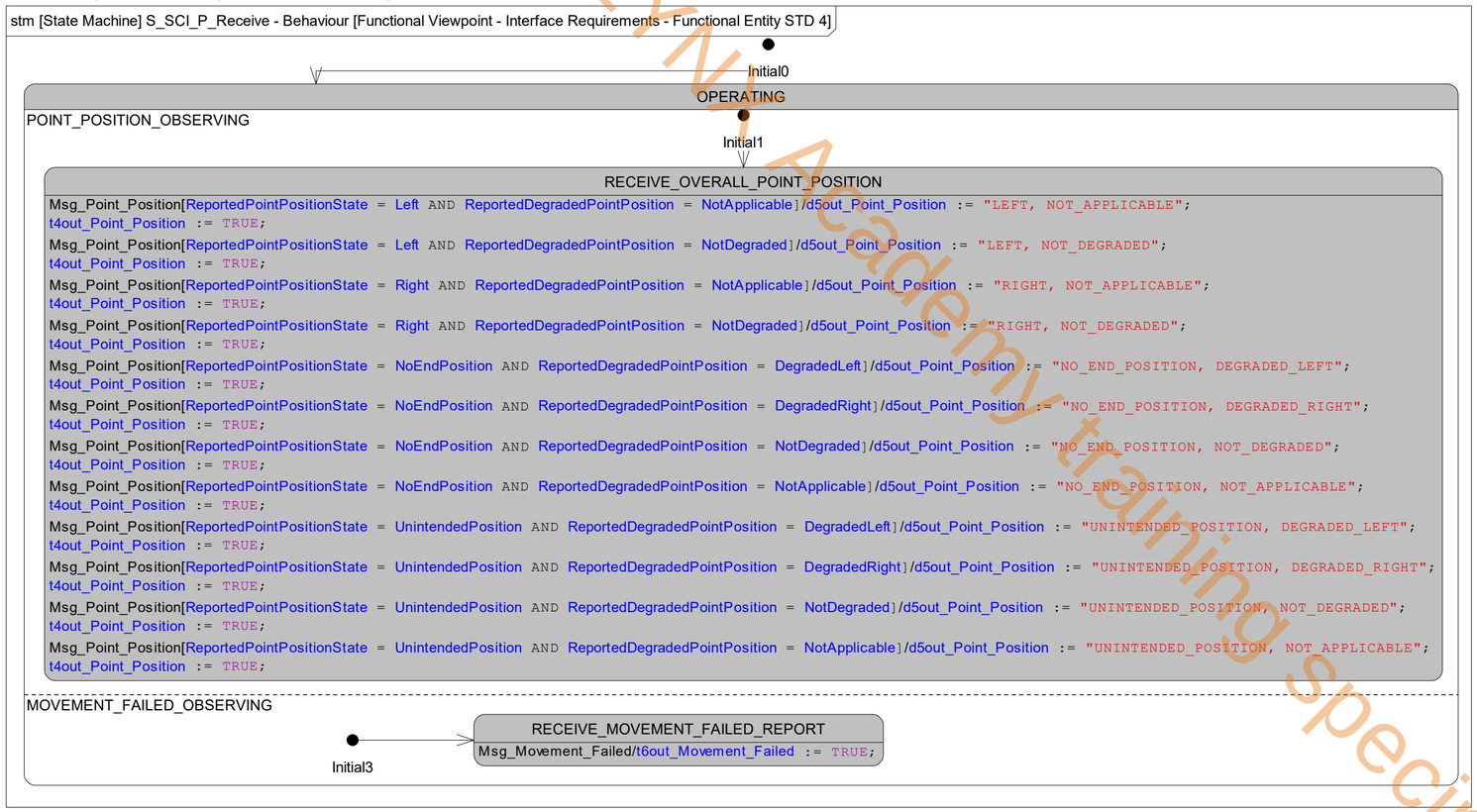
ID	Type	Requirement Part 1	Requirement Part 2
Eu.P.7916	Req	[Block] F_SCI_P_Receive [Functional Viewpoint - Interface Requirements - Functional Entity] 	
Eu.P.7917	Def	d10out_Required_Point_Position	The port d10out_Required_Point_Positiont contains the Point Target Position. The following values are valid: "LEFT", "RIGHT".
Eu.P.7918	Def	d12in_Required_PM_Position	
Eu.P.7919	Def	d50in_PDI_Connection_State	
Eu.P.7920	Info	F_SCI_P_Receive - Behaviour	
Eu.P.7921	Req	Functional Viewpoint - Interface Requirements - Functional Entity STD 1 	This state machine diagram describes the requirements for the following functionalities: - receives the commanded point position state and forwards it to the internal logic of the Subsystem - Point
Eu.P.7922	Def	Initial0	
Eu.P.7923	Def	/{Initial0 - RECEIVING_COMMANDS}	
Eu.P.7924	Def	RECEIVING_COMMANDS	
Eu.P.7925	Def	Cd_Move_Point[CommandedPointPositionState = Left AND d50in_PDI_Connection_State = "ESTABLISHED"]/d10out_Required_Point_Position := "LEFT";{State-internal in RECEIVING_COMMANDS}	
Eu.P.7926	Def	Cd_Move_Point[CommandedPointPositionState = Right AND d50in_PDI_Connection_State = "ESTABLISHED"]/d10out_Required_Point_Position := "RIGHT";{State-internal in RECEIVING_COMMANDS}	
Eu.P.7927	Def	entry/d10out_Required_Point_Position := "UNCOMMANDED";{State-internal in RECEIVING_COMMANDS}	
Eu.P.7928	Def	when(d12in_Required_PM_Position = "UNCOMMANDED")/{RECEIVING_COMMANDS - RECEIVING_COMMANDS}	
Eu.P.7929	Def	when(d50in_PDI_Connection_State = "NOT_READY_FOR_PDI_NO_SCP" OR d50in_PDI_Connection_State = "NOT_READY_FOR_PDI" OR d50in_PDI_Connection_State = "SUSPENDED")/{RECEIVING_COMMANDS - RECEIVING_COMMANDS}	
Eu.P.7930	Def	P3in	The port P3in exchanges information objects according to SCI_P_1.
Eu.P.7931	Info	F_SCI_P_Report	
Eu.P.7932	Req	[Block] F_SCI_P_Report [Functional Viewpoint - Interface Requirements - Functional Entity] 	

ID	Type	Requirement Part 1	Requirement Part 2

Eu.P.7933	Def	/* cOp1_Point_Position */
		
		if (d14in_Observed_Point_Position = "LEFT") then
		return PointPositionState.Left;
		elseif (d14in_Observed_Point_Position = "RIGHT") then
		return PointPositionState.Right;
		elseif (d14in_Observed_Point_Position = "NO_END_POSITION") then
		return PointPositionState.NoEndPosition;
		elseif (d14in_Observed_Point_Position = "UNINTENDED_POSITION") then
		return PointPositionState.UnintendedPosition;
		end if
			cOp1_Point_Position

Eu.P.7934	Def	d13in_Observed_Movement_Failed	
Eu.P.7935	Def	d14in_Observed_Point_Position	
Eu.P.7936	Def	d50in_PDI_Connection_State	
Eu.P.7937	Info	F_SCI_P_Report - Behaviour	
Eu.P.7938	Req	Functional Viewpoint - Interface Requirements - Functional Entity STD 2  	This state machine diagram describes the requirements for the following functionalities: - receives the observed Point Position status from internal logic and reports this to the Subsystem - Electronic Interlocking - receives the observed Movement Failed status from internal logic and reports this to the Subsystem - Electronic Interlocking
Eu.P.7939	Def	Initial0	
Eu.P.7940	Def	/{Initial0 - INTERFACE_CONNECTION_NOT_ESTABLISHED}	
Eu.P.7941	Def	INTERFACE_CONNECTION_NOT_ESTABLISHED	
Eu.P.7942	Def	Start_Status_Report/send Msg_Point_Position (cOp1_Point_Position) to P9out; Mem_Last_Reported_Point_Position := d14in_Observed_Point_Position;{INTERFACE_CONNECTION_NOT_ESTABLISHED - REPORTING_OVERALL_POINT_STATES}	
Eu.P.7943	Def	REPORTING_OVERALL_POINT_STATES	
Eu.P.7944	Def	REPORTING_MOVEMENT_FAILED	
Eu.P.7945	Def	Initial3	
Eu.P.7946	Def	/{Initial3 - MSG_MOVEMENT_FAILED}	
Eu.P.7947	Def	MSG_MOVEMENT_FAILED	
Eu.P.7948	Def	when(d13in_Observed_Movement_Failed)[d50in_PDI_Connection_State = "ESTABLISHED"]/send Msg_Movement_Failed to P9out;{State-internal in MSG_MOVEMENT_FAILED}	
Eu.P.7949	Def	REPORTING_MSG_POINT_POSITION	
Eu.P.7950	Def	Initial1	

ID	Type	Requirement Part 1	Requirement Part 2
Eu.P.7951	Def	/{Initial1 - WAITING}	
Eu.P.7952	Def	MSG_OVERALL_POINT_POSITION	
Eu.P.7953	Def	entry/send Msg_Point_Position (cOp1_Point_Position) to P9out; Mem_Last_Reported_Point_Position := d14in_Observed_Point_Position;{State-internal in MSG_OVERALL_POINT_POSITION}	
Eu.P.7954	Def	when((d14in_Observed_Point_Position <> Mem_Last_Reported_Point_Position))[d50in_PDI_Connection_State = "ESTABLISHED"]/{MSG_OVERALL_POINT_POSITION - MSG_OVERALL_POINT_POSITION}	
Eu.P.7955	Def	WAITING	
Eu.P.7956	Def	when((d14in_Observed_Point_Position <> Mem_Last_Reported_Point_Position))[d50in_PDI_Connection_State = "ESTABLISHED"]/{WAITING - MSG_OVERALL_POINT_POSITION}	
Eu.P.7957	Def	entry/send Status_Report_Completed to p3inout;{State-internal in REPORTING_OVERALL_POINT_STATES}	
Eu.P.7958	Def	when(d50in_PDI_Connection_State = "READY_FOR_PDI_NO_SCP" OR d50in_PDI_Connection_State = "READY_FOR_PDI" OR d50in_PDI_Connection_State = "SUSPENDED")/{REPORTING_OVERALL_POINT_STATES - INTERFACE_CONNECTION_NOT_ESTABLISHED}	
Eu.P.7959	Def	p3inout	
Eu.P.7960	Def	P9out	The port P9out exchanges information objects according to SCI_P_2.
Eu.P.7961	Info	S_SCI_P_Command	
Eu.P.7962	Req	[Block]S_SCI_P_Command_SR [Functional Viewpoint - Interface Requirements - Functional Entity] 	
Eu.P.7963	Def	d2in_Move_Point	
Eu.P.7964	Def	d50in_PDI_Connection_State	
Eu.P.7965	Def	P3out	The port P3out exchanges information objects according to SCI_P_1.
Eu.P.7966	Info	S_SCI_P_Command - Behaviour	
Eu.P.7967	Req	Functional Viewpoint - Interface Requirements - Functional Entity STD 3 	This state machine diagram describes the requirements for the following functionalities: - receives the Moving Command to be set from internal logic and commands this to the Subsystem - Point
Eu.P.7968	Def	Initial0	
Eu.P.7969	Def	/{Initial0 - SENDING_COMMANDS}	
Eu.P.7970	Def	SENDING_COMMANDS	
Eu.P.7971	Def	when(t1in_Move_Point)[d2in_Move_Point = "RIGHT" AND d50in_PDI_Connection_State = "ESTABLISHED"]/send Cd_Move_Point (Right)to P3out;{State-internal in SENDING_COMMANDS}	
Eu.P.7972	Def	when(t1in_Move_Point)[d2in_Move_Point = "LEFT" AND d50in_PDI_Connection_State = "ESTABLISHED"]/send Cd_Move_Point (Left) to P3out;{State-internal in SENDING_COMMANDS}	
Eu.P.7973	Def	t1in_Move_Point	
Eu.P.7974	Info	S_SCI_P_Receive	

ID	Type	Requirement Part 1	Requirement Part 2
Eu.P.7975	Req	[Block]S_SCI_P_Receive_SR [Functional Viewpoint - Interface Requirements - Functional Entity] 	
Eu.P.7976	Def	d5out_Point_Position	
Eu.P.7977	Def	P9in	The port P9in exchanges information objects according to SCI_P_2.
Eu.P.7978	Info	S_SCI_P_Receive - Behaviour	
Eu.P.7979	Req	Functional Viewpoint - Interface Requirements - Functional Entity STD 4 	This state machine diagram describes the requirements for the following functionalities: - receives the reported point position status and forwards it to the internal logic - receives the reported movement failed report and forwards it to the internal logic
Eu.P.7980	Def	Initial0	
Eu.P.7981	Def	/{Initial0 - OPERATING}	
Eu.P.7982	Def	OPERATING	
Eu.P.7983	Def	MOVEMENT_FAILED_OBSERVING	
Eu.P.7984	Def	Initial3	
Eu.P.7985	Def	/{Initial3 - RECEIVE_MOVEMENT_FAILED_REPORT}	
Eu.P.7986	Def	RECEIVE_MOVEMENT_FAILED_REPORT	
Eu.P.7987	Def	Msg_Movement_Failed/t6out_Movement_Failed := TRUE;{State-internal in RECEIVE_MOVEMENT_FAILED_REPORT}	

ID	Type	Requirement Part 1	Requirement Part 2
Eu.P.7988	Def	POINT_POSITION_OBSERVING	
Eu.P.7989	Def	Initial1	
Eu.P.7990	Def	/{Initial1 - RECEIVE_OVERALL_POINT_POSITION}	
Eu.P.7991	Def	RECEIVE_OVERALL_POINT_POSITION	
Eu.P.7992	Def	Msg_Point_Position[ReportedPointPositionState = Left AND ReportedDegradedPointPosition = NotApplicable]/d5out_Point_Position := "LEFT, NOT_APPLICABLE"; t4out_Point_Position := TRUE;{State-internal in RECEIVE_OVERALL_POINT_POSITION_AND_DEGRADED_POINT_POSITION_REPORT}	
Eu.P.7993	Def	Msg_Point_Position[ReportedPointPositionState = UnintendedPosition AND ReportedDegradedPointPosition = DegradedRight]/d5out_Point_Position := "UNINTENDED_POSITION, DEGRADED_RIGHT"; t4out_Point_Position := TRUE;{State-internal in RECEIVE_OVERALL_POINT_POSITION_AND_DEGRADED_POINT_POSITION_REPORT}	
Eu.P.7994	Def	Msg_Point_Position[ReportedPointPositionState = UnintendedPosition AND ReportedDegradedPointPosition = NotDegraded]/d5out_Point_Position := "UNINTENDED_POSITION, NOT_DEGRADED"; t4out_Point_Position := TRUE;{State-internal in RECEIVE_OVERALL_POINT_POSITION_AND_DEGRADED_POINT_POSITION_REPORT}	
Eu.P.7995	Def	Msg_Point_Position[ReportedPointPositionState = UnintendedPosition AND ReportedDegradedPointPosition = NotApplicable]/d5out_Point_Position := "UNINTENDED_POSITION, NOT_APPLICABLE"; t4out_Point_Position := TRUE;{State-internal in RECEIVE_OVERALL_POINT_POSITION_AND_DEGRADED_POINT_POSITION_REPORT}	
Eu.P.7996	Def	Msg_Point_Position[ReportedPointPositionState = Left AND ReportedDegradedPointPosition = NotDegraded]/d5out_Point_Position := "LEFT, NOT_DEGRADED"; t4out_Point_Position := TRUE;{State-internal in RECEIVE_OVERALL_POINT_POSITION_AND_DEGRADED_POINT_POSITION_REPORT}	
Eu.P.7997	Def	Msg_Point_Position[ReportedPointPositionState = Right AND ReportedDegradedPointPosition = NotApplicable]/d5out_Point_Position := "RIGHT, NOT_APPLICABLE"; t4out_Point_Position := TRUE;{State-internal in RECEIVE_OVERALL_POINT_POSITION_AND_DEGRADED_POINT_POSITION_REPORT}	
Eu.P.7998	Def	Msg_Point_Position[ReportedPointPositionState = Right AND ReportedDegradedPointPosition = NotDegraded]/d5out_Point_Position := "RIGHT, NOT_DEGRADED"; t4out_Point_Position := TRUE;{State-internal in RECEIVE_OVERALL_POINT_POSITION_AND_DEGRADED_POINT_POSITION_REPORT}	
Eu.P.7999	Def	Msg_Point_Position[ReportedPointPositionState = NoEndPosition AND ReportedDegradedPointPosition = DegradedLeft]/d5out_Point_Position := "NO_END_POSITION, DEGRADED_LEFT"; t4out_Point_Position := TRUE;{State-internal in RECEIVE_OVERALL_POINT_POSITION_AND_DEGRADED_POINT_POSITION_REPORT}	
Eu.P.8000	Def	Msg_Point_Position[ReportedPointPositionState = NoEndPosition AND ReportedDegradedPointPosition = DegradedRight]/d5out_Point_Position := "NO_END_POSITION, DEGRADED_RIGHT"; t4out_Point_Position := TRUE;{State-internal in RECEIVE_OVERALL_POINT_POSITION_AND_DEGRADED_POINT_POSITION_REPORT}	
Eu.P.8001	Def	Msg_Point_Position[ReportedPointPositionState = NoEndPosition AND ReportedDegradedPointPosition = NotDegraded]/d5out_Point_Position := "NO_END_POSITION, NOT_DEGRADED"; t4out_Point_Position := TRUE;{State-internal in RECEIVE_OVERALL_POINT_POSITION_AND_DEGRADED_POINT_POSITION_REPORT}	
Eu.P.8002	Def	Msg_Point_Position[ReportedPointPositionState = NoEndPosition AND ReportedDegradedPointPosition = NotApplicable]/d5out_Point_Position := "NO_END_POSITION, NOT_APPLICABLE"; t4out_Point_Position := TRUE;{State-internal in RECEIVE_OVERALL_POINT_POSITION_AND_DEGRADED_POINT_POSITION_REPORT}	
Eu.P.8003	Def	Msg_Point_Position[ReportedPointPositionState = UnintendedPosition AND ReportedDegradedPointPosition = DegradedLeft]/d5out_Point_Position := "UNINTENDED_POSITION, DEGRADED_LEFT"; t4out_Point_Position := TRUE;{State-internal in RECEIVE_OVERALL_POINT_POSITION_AND_DEGRADED_POINT_POSITION_REPORT}	
Eu.P.8004	Def	t4out_Point_Position	
Eu.P.8005	Def	t6out_Movement_Failed	
Eu.P.8028	Head	**3.4.2 SMI-P (Subsystem - Maintenance and Data Management)**	
Eu.P.8029	Info	The generic FlowSpecification and the related FlowProperties through SMI-P are specified in [Eu.Doc.120].	
Eu.P.8026	Head	**3.4.3 SDI-P (Subsystem - Maintenance and Data Management)**	
Eu.P.8027	Info	The generic data points through the SDI-P are specified in [Eu.Doc.94]. The specific data points through the SDI-P are specified in [Eu.Doc.80].	
Eu.P.8030	Head	**3.4.4 SSI-P (Subsystem - Security Services Platform)**	
Eu.P.8031	Info	The generic content through SSI-P is specified in [Eu.Doc.117].	
Eu.P.7907	Head	**3.4.5 P4 (Basic Data Identifier)**	
Eu.P.7908	Info	The generic FlowSpecification and the related FlowProperties through P4 are specified in [Eu.Doc.20].	
Eu.P.7891	Head	**3.4.6 P1 (Maintainer)**	
Eu.P.7892	Info	The generic FlowProperties through P1 are specified in [Eu.Doc.20].	
Eu.P.7893	Info	Maintainer	Definition of the InformationFlow (by FlowSpecification) for Maintenance/Operation/Display P1 (Maintainer).
Eu.P.7894	Def	End_Position_L	Displays the status of the detection of point end position on the left hand.
Eu.P.7895	Def	End_Position_R	Displays the status of the detection of point end position on the right hand.
Eu.P.7896	Def	P3_Moving_N	Displays the state for each respective physical Point Machine Output (N > 0). (Moving, Stopped).
Eu.P.7897	Def	P3_Status_N	Displays the state for each respective physical Point Machine Input (N > 0). (End_Position, No_End_Position, Unintended_Position).
Eu.P.7898	Def	Point_Moving	Displays the moving of the point at the local status display.
Eu.P.7900	Head	**3.4.7 P3 (Point machine)**	
Eu.P.7901	Head	Point_machine	Definition of the InformationFlow (by FlowSpecification) for the Control Interfaces P3 (Point machine). Note: The behaviour of the interfaces P3 is described

ID	Type	Requirement Part 1	Requirement Part 2
			generically. The Subsystem - Point needs to be able to write and to read the generic information objects of the statuses from the Point machine.
Eu.P.7903	Def	Information_End_Position_Detected	Information object from Point machine to Subsystem - Point that the Point machine has reliably reached an End position (left hand position or right hand position).
Eu.P.7904	Def	Information_No_End_Position	Information object from Point machine to Subsystem - Point that the Point machine has No end position.
Eu.P.7905	Def	Moving	Information object from Subsystem - Point to Point machine to move the Point machine to an end position (left hand position or right hand position).
Eu.P.7906	Def	Stop_Moving	Information object from Subsystem - Point to Point machine to stop Moving the Point machine.
Eu.P.233	Head	**4 RAMSS requirements**	
Eu.P.2987	Info	The requirements for reliability, availability, maintainability, safety and security are specified in [Eu.Doc.20]	
Eu.P.3244	Head	**5 Technical Requirements**	
Eu.P.3245	Info	The generic technical requirements are specified in [Eu.Doc.20]	
Eu.P.3246	Head	**5.1 Specific technical interface requirements**	
Eu.P.3247	Head	**5.1.1 Interface to the Point of Service Signalling (PoS-Signalling)**	
Eu.P.3248	Req	Via the technical interface **PoS-Signalling**, the data of the functional interface "SCI-P" shall be exchanged with the Subsystem - Electronic Interlocking as specified in [Eu.Doc.92].	
Eu.P.3249	Req	Via the technical interface **PoS-Signalling**, the data of the functional interface "SMI-P" shall be exchanged with the Subsystem - Maintenance and Data Management as specified in [Eu.Doc.76].	
Eu.P.3250	Req	Via the technical interface **PoS-Signalling**, the data of the functional interface "SDI-P" shall be exchanged with the Subsystem - Maintenance and Data Management as specified in [Eu.Doc.77].	
Eu.P.6207	Req	Via the technical interface **PoS-Signalling**, the data of the functional interface "SSI-P" shall be exchanged with the Subsystem - Maintenance and Data Management as specified in [Eu.Doc.117].	
Eu.P.3251	Head	**5.1.2 Interface to the point machine**	
Eu.P.3252	Req	The technical requirements for both implementation variants shall be defined by national specifications. Note: In future phases of the System Pillar, national specifications will be replaced by harmonised specifications.	
Eu.P.4946	Info	Power property assumptions • The P3 interface may be implemented with separate power supplies for driving and detection. • The power supply for detection is provided by the Subsystem - Point. • A loss of the power supply for detection will report as “Information_No_End_Position” on P3. • When a crank handle is inserted to the Point Machine, typically its power supply for driving is disconnected. (Note: A point machine may not be implemented with the function to monitor its power supply for driving).	
Eu.P.4941	Info	P3 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 P3 interface.	
Eu.P.4942	Req	On a Point machine with detection functionality only, the information objects Moving and Stop_Moving are not available on P3.	
Eu.P.3253	Head	**5.2 Time behaviour**	
Eu.P.3254	Req	The time values defined in the chapter Functional requirements specification (Eu.P.2286) shall be configured for the operation of the Subsystem - Point.	
Eu.P.3262	Head	**5.2.1 Response times**	
Eu.P.3263	Req	The Subsystem - Point shall send the corresponding message telegram to the Subsystem - Electronic Interlocking within 250 ms after successful change of state, according to specific use cases.	
Eu.P.3264	Req	The Subsystem - Point shall start the reversal operation within 500 ms after receiving a command telegram.	
Eu.P.3255	Head	**5.3 Configuration and engineering data**	
Eu.P.3256	Head	**5.3.1 Specific data**	
Eu.P.3257	Req	The engineering and configuration data for the Subsystem - Point shall include as a minimum the following information:	
Eu.P.3258	Req	• The applicable time values defined in chapter Definition of time values (Eu.P.2286).	
Eu.P.3259	Info	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:	
Eu.P.3260	Req	• configuration data, such as the IP addresses of the Subsystem - Electronic Interlocking , the value of the diagnostic data points with attribute type 'configuration', is not safety-relevant. This data shall be used to calculate the CSNS.	
Eu.P.3261	Req	• The remaining configuration data is currently categorised as safety-relevant. This data shall be used to calculate the CSS.	

ID	Type	Requirement Part 1	Requirement Part 2
Eu.P.4546	Req	• The engineering data is safety-relevant. This data shall be used to calculate the CSS.	
Eu.P.7244	Head	**5.3.2 Value configuration**	
Eu.P.7245	Req	**Con_tmax_Point_Operation** The time value shall be configured in accordance with: Configurable resolution: steps of 100 ms. Configurable range: from 100 ms up to 30 s Con_tmax_Point_Operation is defined in Eu.P.2439.	