EULYNX Initiative

Europe's Rail Joint Undertaking

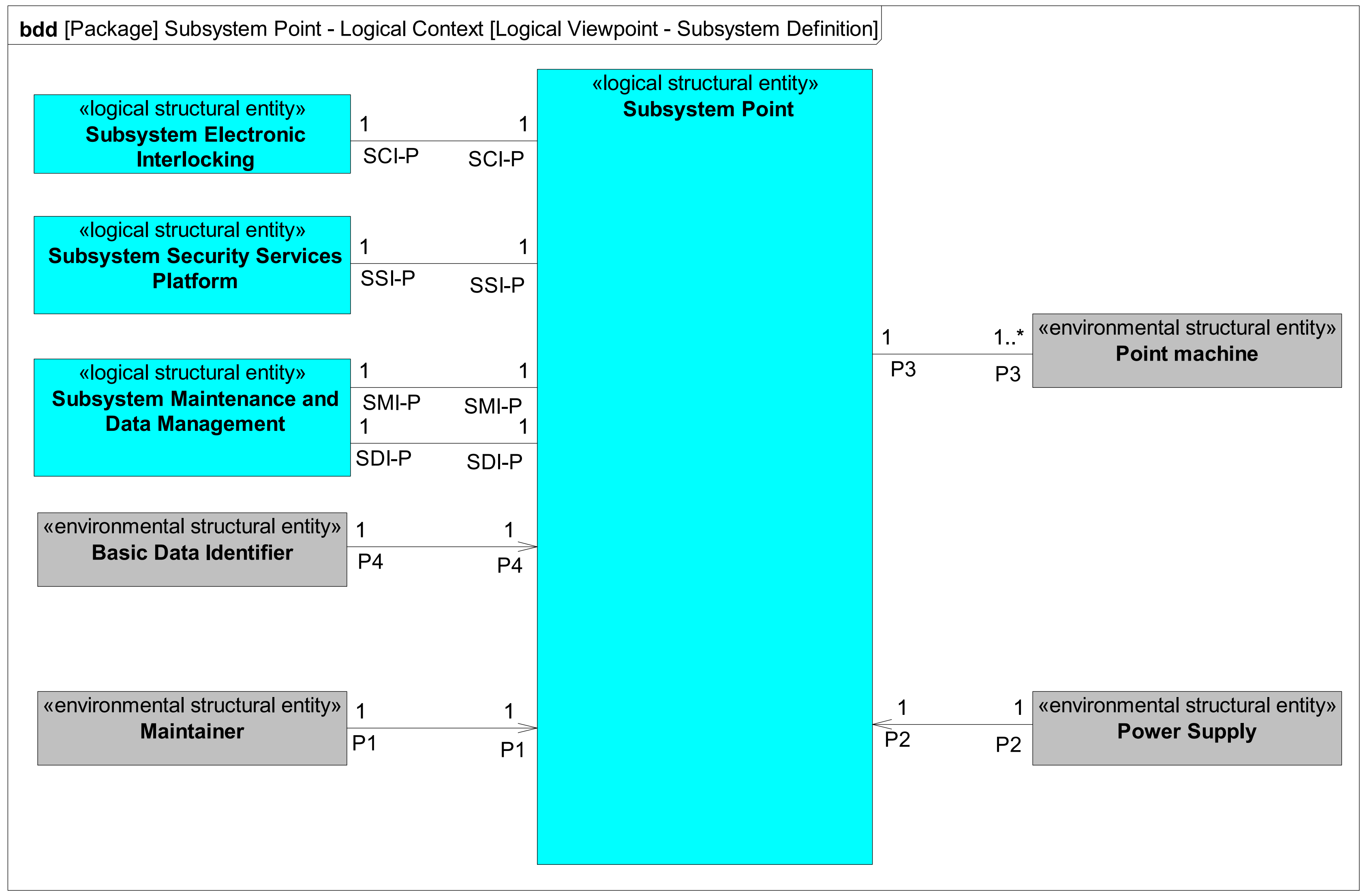
Requirements specification for subsystem Point

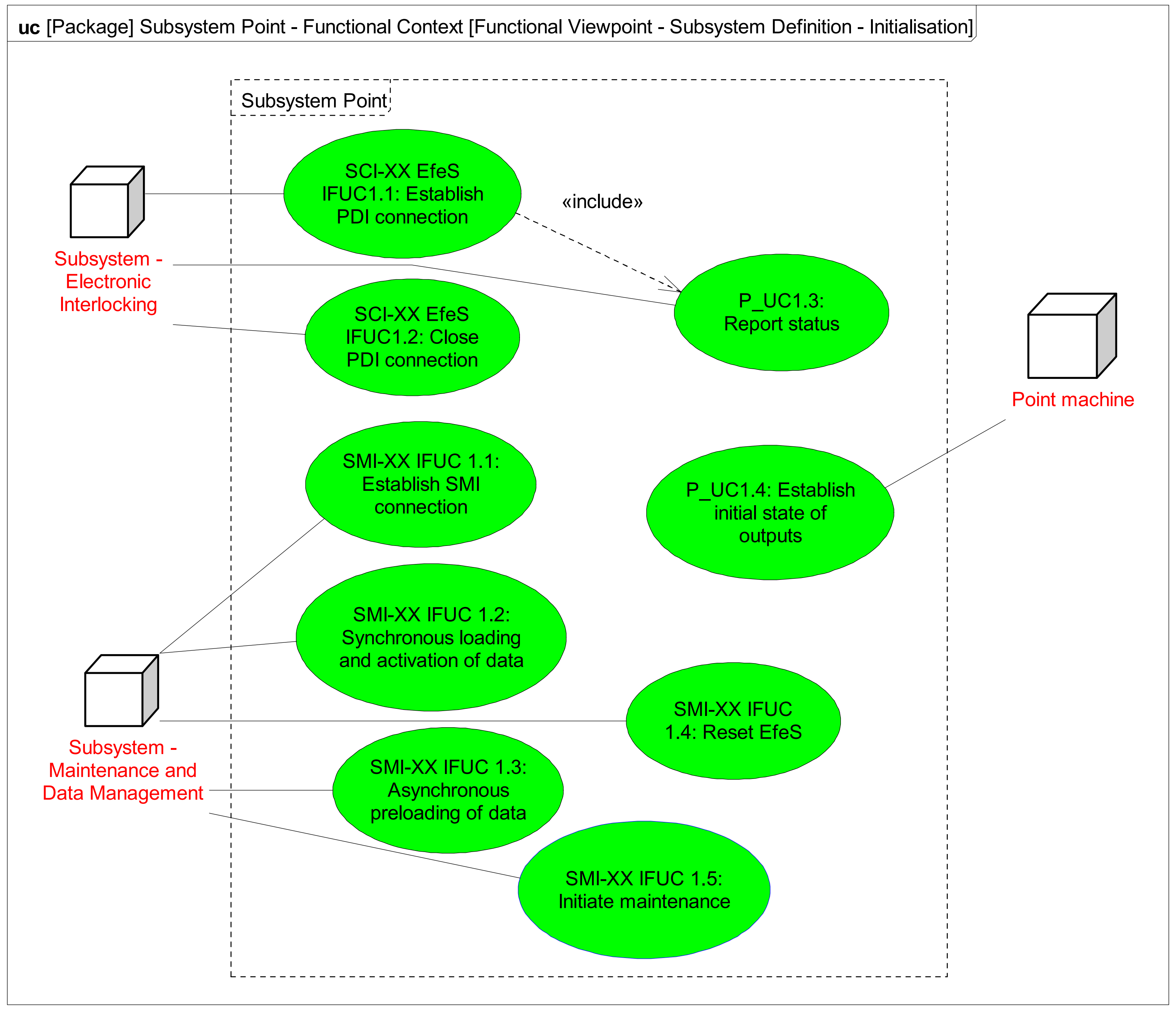
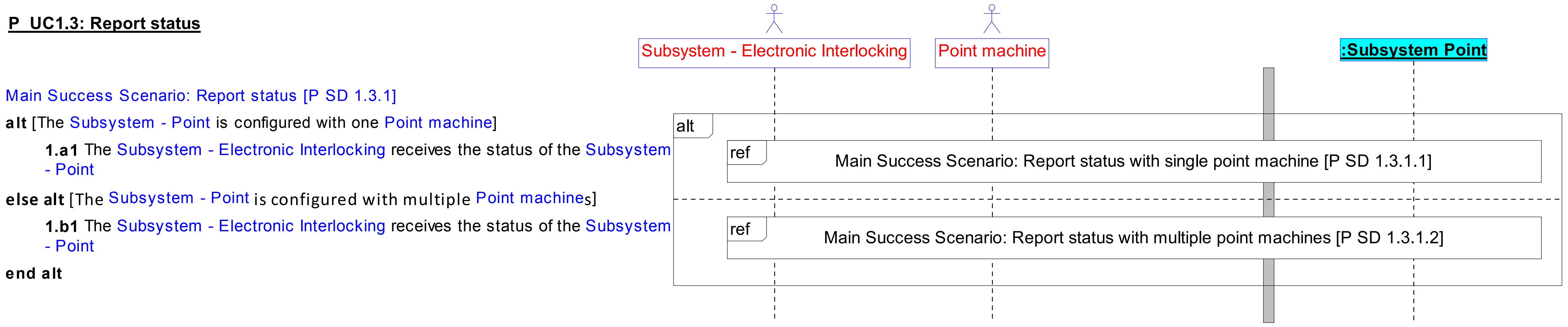
Document number: Eu.Doc.36
Version: 4.3 (0.A)

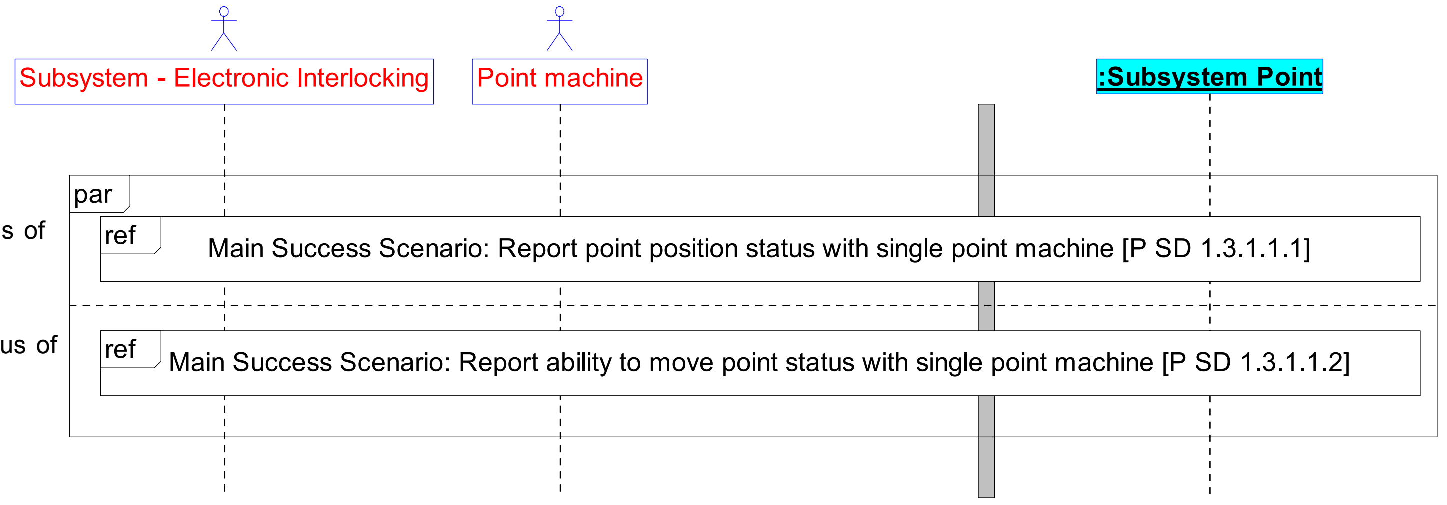
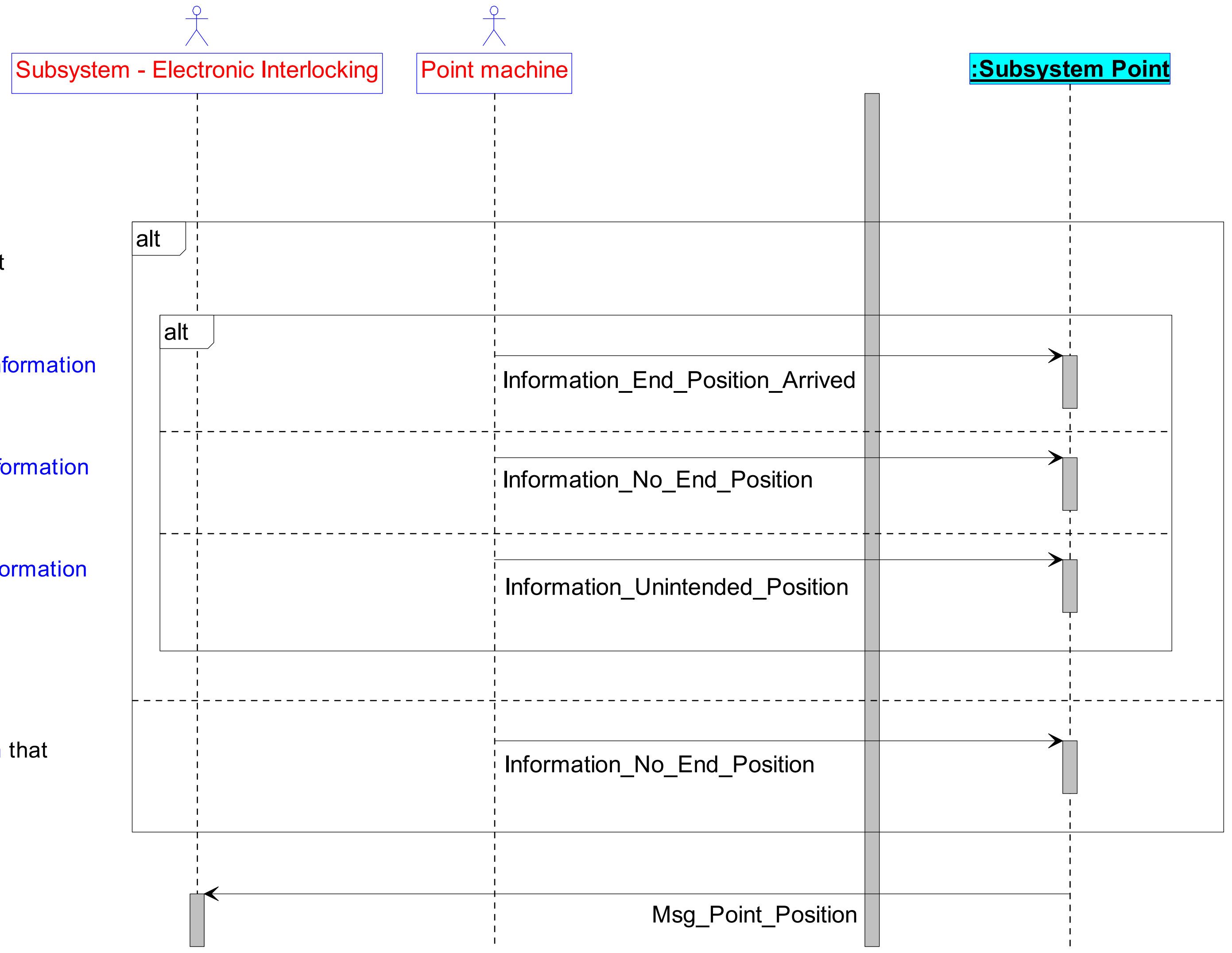
Contents

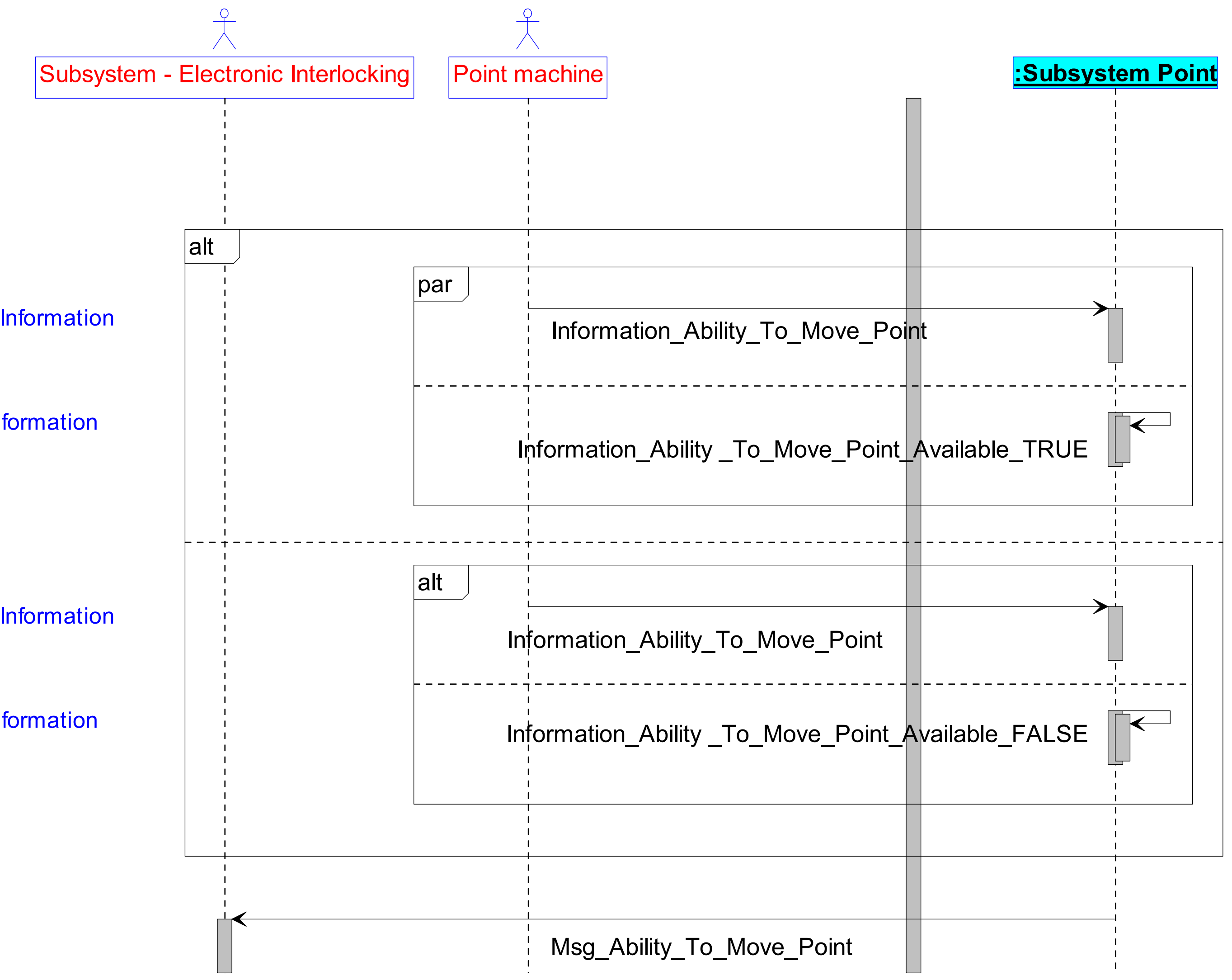
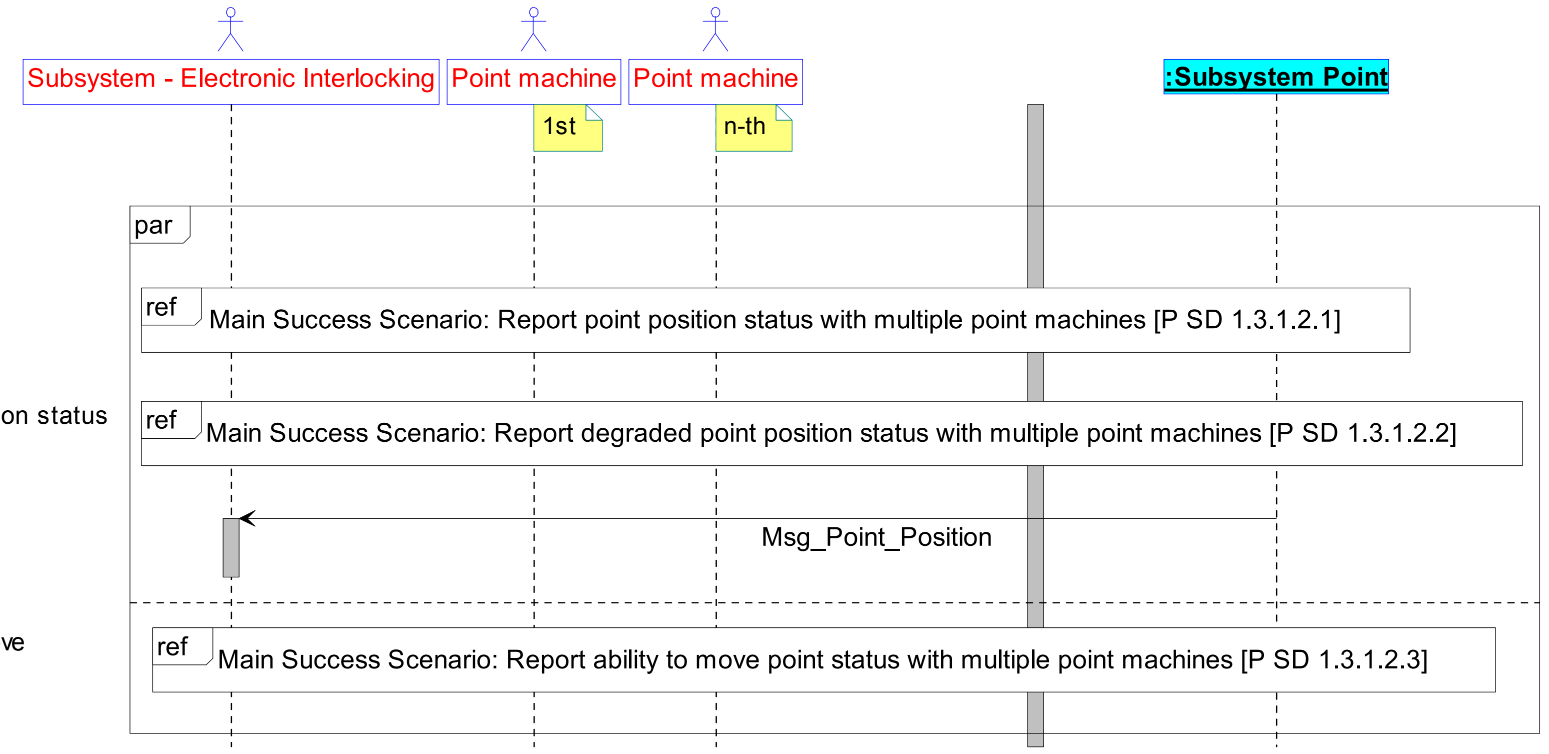
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.1	Head	**1 Introduction**		
Eu.P.2	Head	**1.1 Release information**		
Eu.P.3	Info	[Eu.Doc.36] Requirements specification for subsystem Point CENELEC Phase: 4 Version: 4.3 (0.A) Approval date: 15.06.2023		
Eu.P.3032	Info	**Version history**		
Eu.P.6591	Info	version number: 4.0 (0.A) date: 16.05.2022 author: Andreas Staudte 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: EUP-383, EUP-384, EUP-385, EUP-386, EUP-387, EUP-391, EUP-393		
Eu.P.6701	Info	version number: 4.1 (0.A) date: 17.04.2023 author: Philipp Wolber, Filip Giering 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: changes: EUP-396, EUP-397, EUP-389, EUP-405, EUP-406, EUP-409, EUP-412, EUP-413, EUP-415, EUP-416, EUP-418, EUP-421, EUP-429, EUP-431, EUP-433, EUP-434, EUP-436, EUP-437, EUP-441, EUP-442, EUP-443, EUP-444, EUP-445, EUP-446, EUP-448, EUP-449, EUP-451, EUP-452		
Eu.P.7097	Info	version number: 4.2 (0.A) date: 11.05.2023 author: Philipp Wolber, Filip Giering, Dominik Smajgl 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: EUP-453, EUP-454, EUP-455, EUP-456, EUP-458, EUP-460, EUP-461, EUP-462, EUP-463, EUP-469, EUP-470, EUP-472, EUP-475, EUP-476, EUP-481		
Eu.P.7153	Info	version number: 4.2 (1.A) date: 01.06.2023 author: Philipp Wolber, Filip Giering, Dominik Smajgl 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: EUP-457, EUP-459, EUP-467, EUP-468, EUP-471, EUP-473, EUP-478, EUP-479, EUP-482, EUP-483, EUP-488, EUP-489, EUP-491, EUP-493		
Eu.P.7198	Info	version number: 4.3 (0.A) date: 28.06.2023 author: Philipp Wolber, Filip Giering 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: EUP-502, EUP-503, EUP-504, EUP-505, EUP-506, EUP-507, EUP-509, EUP-510, EUP-513		
Eu.P.7	Head	**1.2 Impressum**		
Eu.P.8	Info	Publishers: **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.P.9	Info	Responsible for this document: EU-Rail System Pillar Trackside Assets Control and Supervision domain		
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 intended for the following users: • safety authorities • infrastructure managers • safety assessors • signalling system suppliers • validators		
Eu.P.14	Info	This document is the basis for the implementation by the supplier and for approval by the infrastructure manager.		
Eu.P.7144	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.		

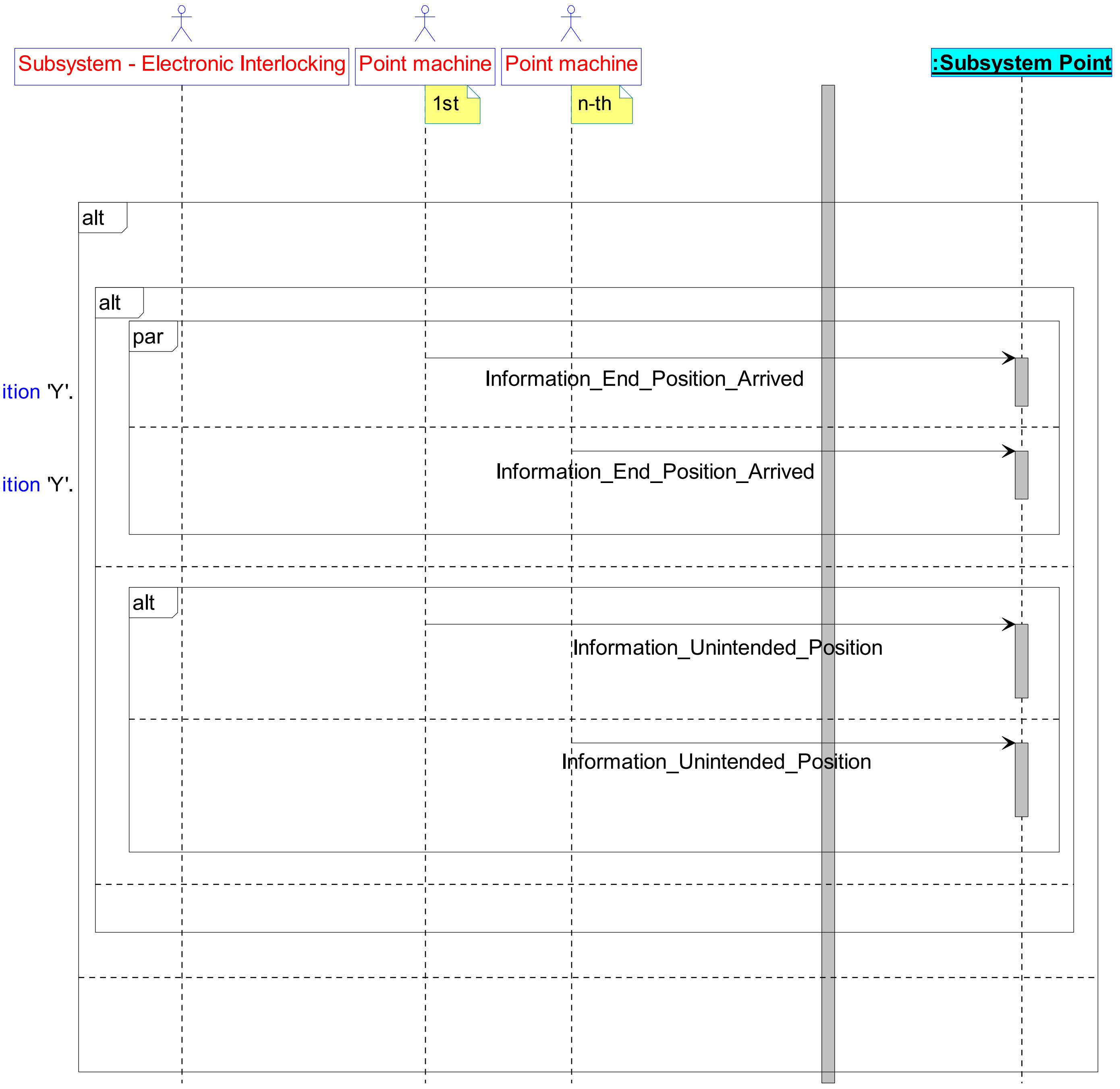
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.15	Head	**1.4 Applicable standards and regulations**		
Eu.P.314	Info	The applicable standards and regulations used in EULYNX are listed in the EULYNX Reference Document List [Eu.Doc.12].		
Eu.P.35	Head	**1.5 Applicable documents**		
Eu.P.36	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.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.57	Head	**2 Conditions of use**		
Eu.P.4952	Req	All references to Eu.Doc.20 refer to version 4.0 (3.A) of that document.		
Eu.P.6374	Req	All references to Eu.Doc.119 refer to version 1.0 (3.A) of that document.		
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.6159	Head	**2.1 Functional Packages**		
Eu.P.6160	Info	The specifications in this document are divided into functional packages. There are two types of packages related to the product capabilities.		
Eu.P.6161	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.P.6162	Info	'Optional package': One or more packages that can be optionally implemented in addition to one or more basic packages.		
Eu.P.6163	Info	The specifications of the Subsystem – Point are divided into the following functional packages:		
Eu.P.6164	Info	Subsystem Point for a single point machine 4 wire (basic package) [Basic 4-wire single P].		
Eu.P.7154	Info	Subsystem Point for a single point machine non 4 wire (basic package) [Basic non-4-wire single P].		
Eu.P.6165	Info	Subsystem Point for multiple point machines 4 wire (basic package) [Basic 4-wire multiple P].		
Eu.P.7155	Info	Subsystem Point for multiple point machines non 4 wire (basic package) [Basic non-4-wire multiple P].		
Eu.P.6166	Info	Ability to Move functionality (optional package) [Option Able to move]. Note: This optional functional package can be combined with any of the 4 basic functional packages.		Option Able to move
Eu.P.7156	Info	Redrive functionality (optional package) [Option Redrive]. Note: This optional functional package can be combined only with the basic packages [Basic non-4-wire single P] and [Basic non-4-wire multiple P]		Option Redrive
Eu.P.7157	Info	Common Drive functionality (optional package) [Option Common Drive]. Note: This optional functional package can be combined only with the basic package [Basic non-4-wire multiple P].		Option Common Drive
Eu.P.884	Head	**3 Functional requirements specification**		
Eu.P.6039	Head	**3.1 Subsystem Point - General Infos and Assumptions**		

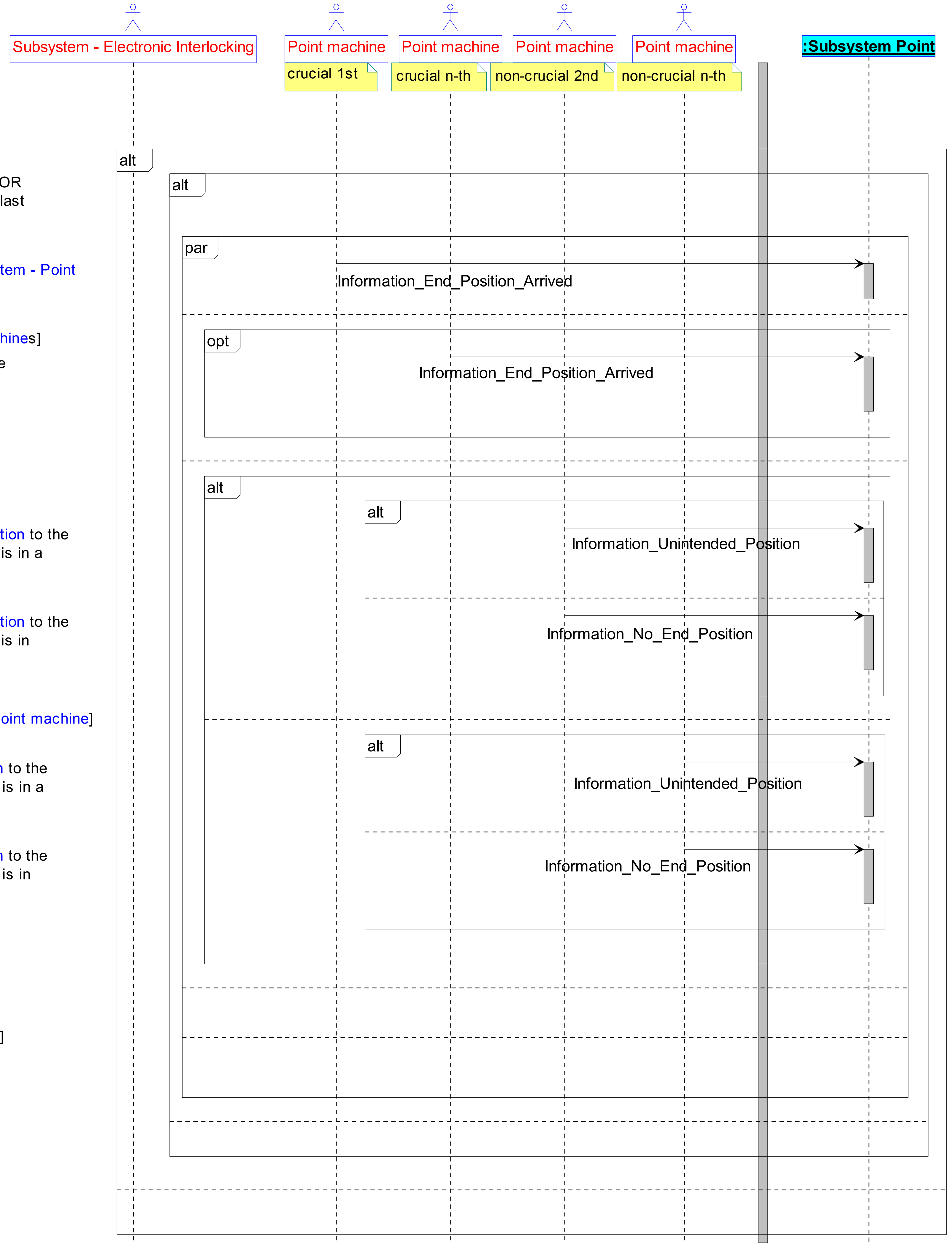
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.6292	Info	The defined model elements represent the Subsystem - Point in • The functional architectures shown in the internal block diagrams • The defined number of Point machines in the state diagrams • The timing behaviour related to individual Point Machines.		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6158	Head	**3.2 Subsystem Point - Logical Viewpoint**		
Eu.P.6282	Head	**3.2.1 Subsystem Point - Logical Context**		
Eu.P.950	Info	[Package] Subsystem Point - Logical Context [Logical Viewpoint - Subsystem Definition]  bdd [Package] Subsystem Point - Logical Context [Logical Viewpoint - Subsystem Definition] «logical structural entity» Subsystem Electronic Interlocking 1 SCI-P 1 SCI-P «logical structural entity» Subsystem Security Services Platform 1 SSI-P 1 SSI-P «logical structural entity» Subsystem Maintenance and Data Management 1 SMI-P 1 SMI-P 1 SDI-P 1 SDI-P «environmental structural entity» Basic Data Identifier 1 P4 1 P4 «environmental structural entity» Maintainer 1 P1 1 P1 «logical structural entity» Subsystem Point 1 P3 1..* P3 «environmental structural entity» Point machine 1 P2 1 P2 «environmental structural entity» Power Supply	The Subsystem - Point shall provide the technical interfaces shown in "[Package] Subsystem Point - Logical Context [Logical Viewpoint - Subsystem Definition]". Each interface shall allow the connection to the corresponding actors shown in the quantities defined in the multiplicities. The Subsystem - Point has to be able to manage and control more than one connected Point machine. It does not send each status input from the n-fold (n = 1...*) Point machine to the Subsystem - Electronic Interlocking, but instead sends one collective message.	Basic 4-wire single P Basic 4-wire multiple P Basic non-4-wire single P Basic non-4-wire multiple P
Eu.P.5780	Head	**3.3 Subsystem Point - Functional Viewpoint**		
Eu.P.2286	Head	**3.3.1 Definition of time values**		
Eu.P.3068	Info	The generic time values for SCI are specified in Eu.Doc.119.		Basic 4-wire single P Basic 4-wire multiple P Basic non-4-wire single P Basic non-4-wire multiple P
Eu.P.6376	Info	The generic time values for SMI are specified in Eu.Doc.120.		Basic 4-wire single P Basic 4-wire multiple P Basic non-4-wire single P Basic non-4-wire multiple P
Eu.P.2439	Info	Con_tmax_Point_Operation	Con_tmax_Point_Operation is a configurable timer, starting with the command Moving to the Point machines and defines the max. time span after the Point has to arrive to an End position. After that time span the command Moving to the Point machines is set to Stop moving. The time value shall be configured in accordance to: resolution: in steps of 100 ms. range: Between 100 ms and 30 s	Basic 4-wire single P Basic 4-wire multiple P Basic non-4-wire single P Basic non-4-wire multiple P
Eu.P.5782	Head	**3.3.2 Subsystem Point - Functional Context**		

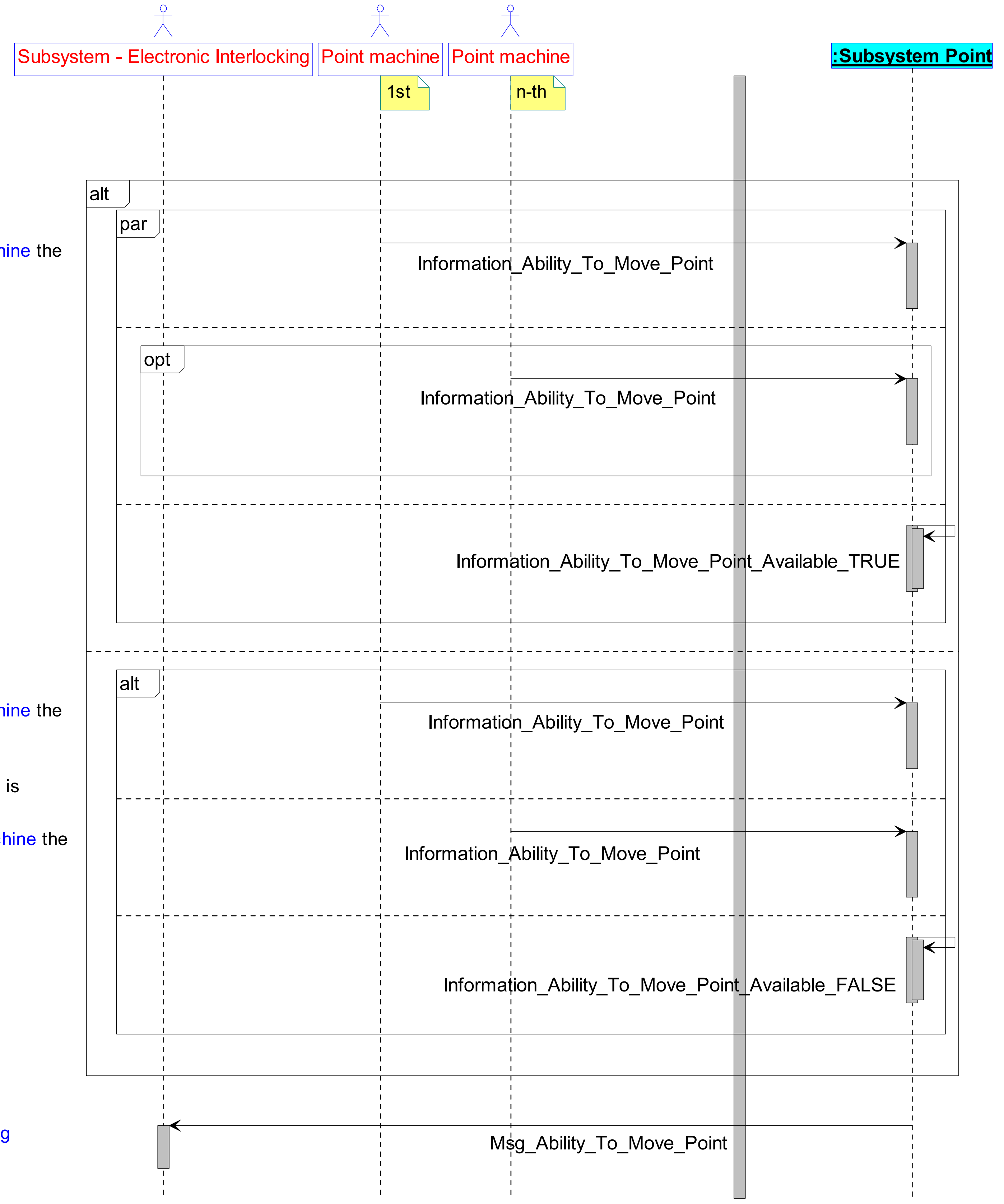
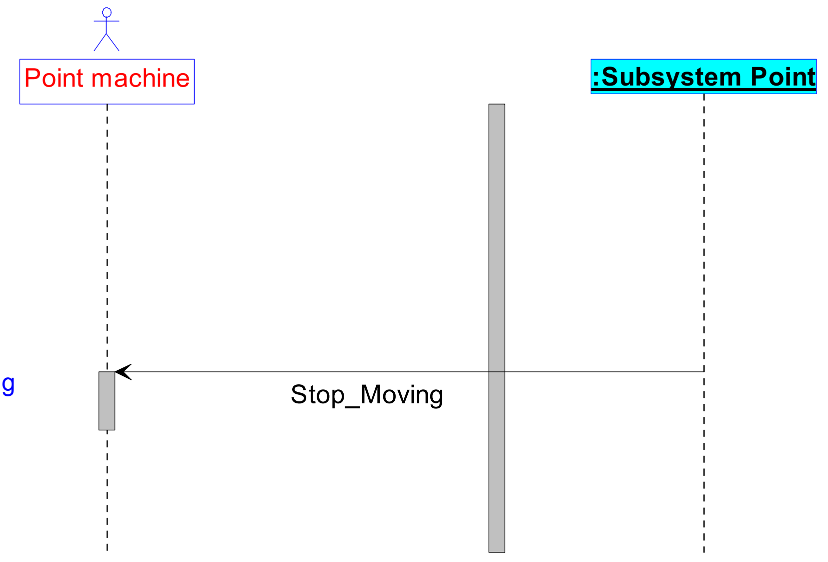
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.1467	Info	[Package] Subsystem Point - Functional Context [Functional Viewpoint - Subsystem Definition - Initialisation]  uc [Package] Subsystem Point - Functional Context [Functional Viewpoint - Subsystem Definition - Initialisation] Subsystem Point Subsystem - Electronic Interlocking SCI-XX EfeS IFUC1.1: Establish PDI connection «include» P_UC1.3: Report status SCI-XX EfeS IFUC1.2: Close PDI connection Point machine SMI-XX IFUC 1.1: Establish SMI connection P_UC1.4: Establish initial state of outputs SMI-XX IFUC 1.2: Synchronous loading and activation of data SMI-XX IFUC 1.4: Reset EfeS Subsystem - Maintenance and Data Management SMI-XX IFUC 1.3: Asynchronous preloading of data SMI-XX IFUC 1.5: Initiate maintenance		Basic 4-wire single P Basic 4-wire multiple P Basic non-4-wire single P Basic non-4-wire multiple P
Eu.P.3070	Info	The generic UseCases SCI-XX EfeS IFUC1.1: Establish PDI connection and SCI-XX EfeS IFUC1.2: Close PDI connection are specified in Eu.Doc.119. The generic UseCases SMI-XX IFUC 1.1: Establish SMI connection, SMI-XX IFUC 1.2: Synchronous loading and activation of data, SMI-XX IFUC 1.3: Asynchronous preloading of data, SMI-XX IFUC 1.4: Reset EfeS and SMI-XX IFUC 1.5: Initiate maintenance are specified in Eu.Doc.120.		Basic 4-wire single P Basic 4-wire multiple P Basic non-4-wire single P Basic non-4-wire multiple P
Eu.P.1465	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.	Basic 4-wire single P Basic 4-wire multiple P Basic non-4-wire single P Basic non-4-wire multiple P
Eu.P.5788	Info	P SD 1.3.1  **P_UC1.3: Report status** Main Success Scenario: Report status [P SD 1.3.1] **alt** [The Subsystem - Point is configured with one Point machine] **1.a1** The Subsystem - Electronic Interlocking receives the status of the Subsystem - Point **else alt** [The Subsystem - Point is configured with multiple Point machines] **1.b1** The Subsystem - Electronic Interlocking receives the status of the Subsystem - Point **end alt** Subsystem - Electronic Interlocking Point machine :Subsystem Point alt ref Main Success Scenario: Report status with single point machine [P SD 1.3.1.1] ref Main Success Scenario: Report status with multiple point machines [P SD 1.3.1.2]		Basic 4-wire single P Basic 4-wire multiple P Basic non-4-wire single P Basic non-4-wire multiple P

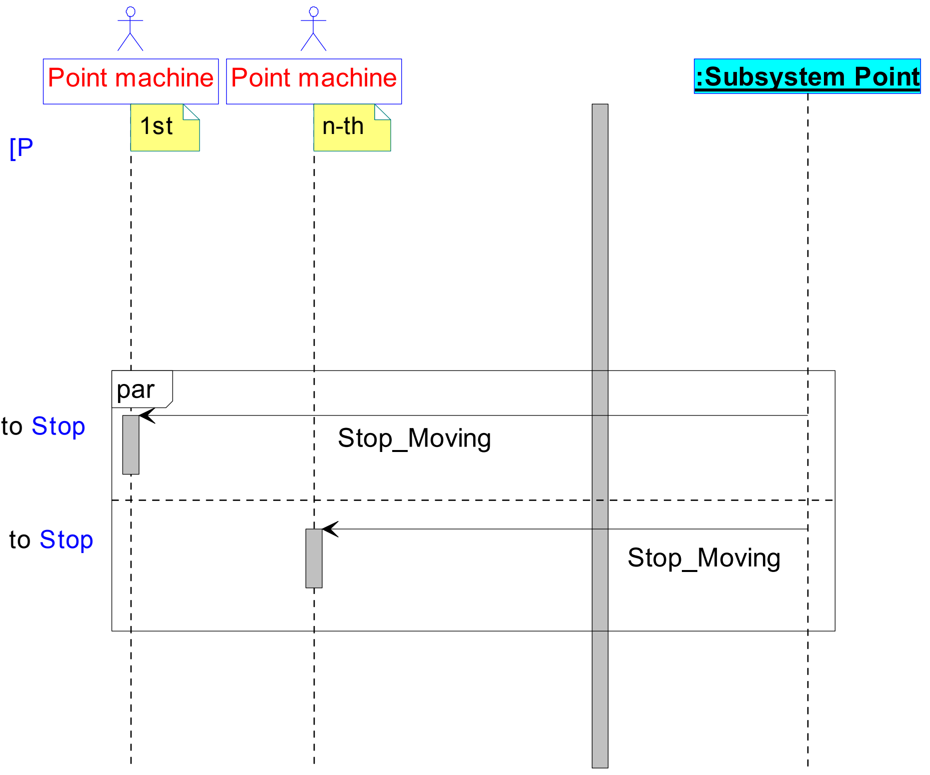
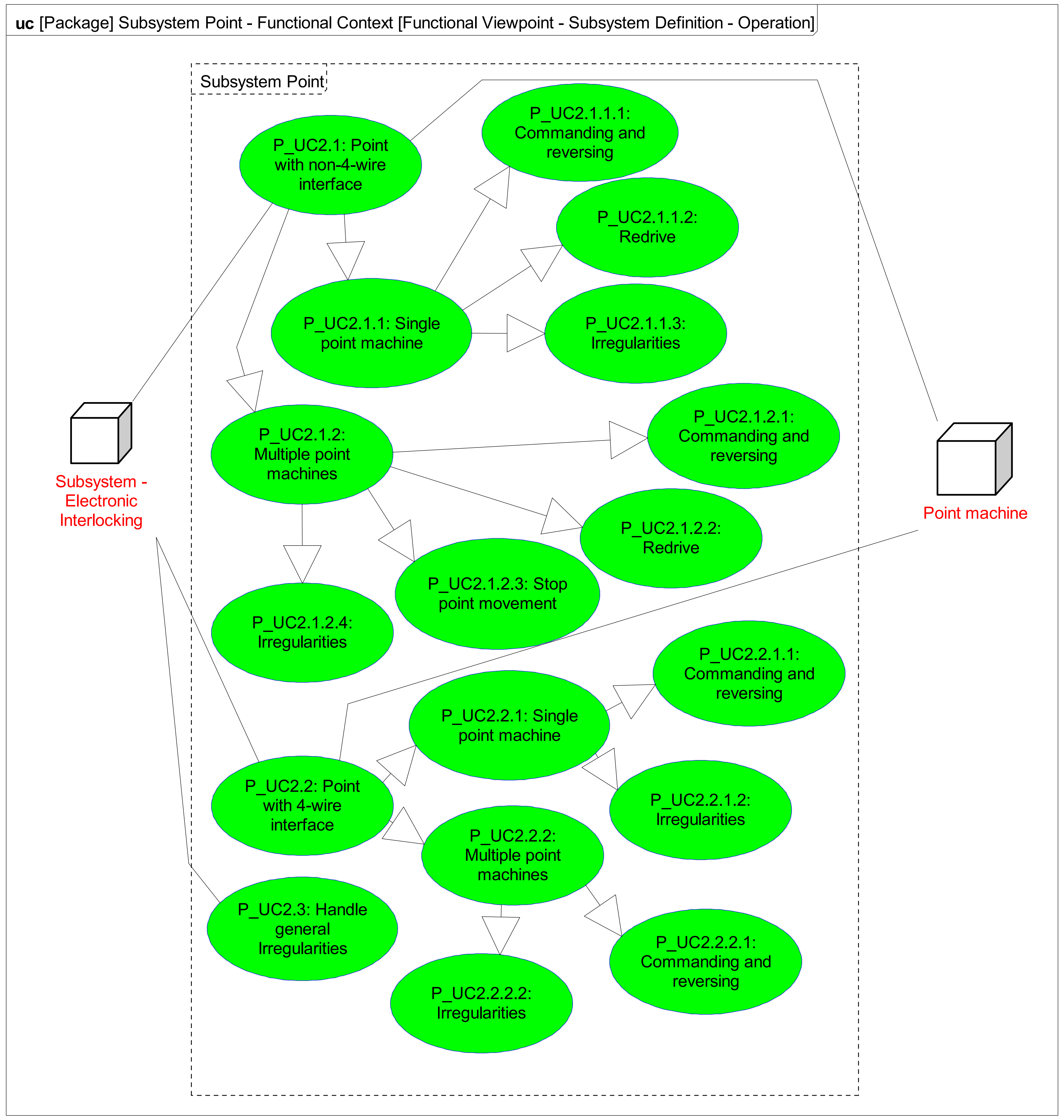
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.1110	Info	P SD 1.3.1.1 **P_UC1.3: Report status** Main Success Scenario: Report status with single point machine [P SD 1.3.1.1] **par** **1.a1** The Subsystem - Electronic Interlocking receives the point position status of the Subsystem - Point **also par** **1.b1** The Subsystem - Electronic Interlocking receives the ability to move status of the Subsystem - Point **end par**  Subsystem - Electronic Interlocking Point machine :Subsystem Point par ref Main Success Scenario: Report point position status with single point machine [P SD 1.3.1.1.1] ref Main Success Scenario: Report ability to move point status with single point machine [P SD 1.3.1.1.2]		Basic 4-wire single P Basic non-4-wire single P
Eu.P.5787	Info	P SD 1.3.1.1.1 **P_UC1.3: Report status** Main Success Scenario: Report point position status with single point machine [P SD 1.3.1.1.1] **Interaction 1.3.1.1.1.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.  Subsystem - Electronic Interlocking Point machine :Subsystem Point alt alt Information_End_Position_Arrived Information_No_End_Position Information_Unintended_Position Information_No_End_Position Msg_Point_Position	If a state change happens while establishing the PDI connection and the status report Msg_Point_Position has already been sent, a new status report Msg_Point_Position has to be sent to Subsystem - Electronic Interlocking immediately after completion of the establishment of the connection.	Basic 4-wire single P Basic non-4-wire single P

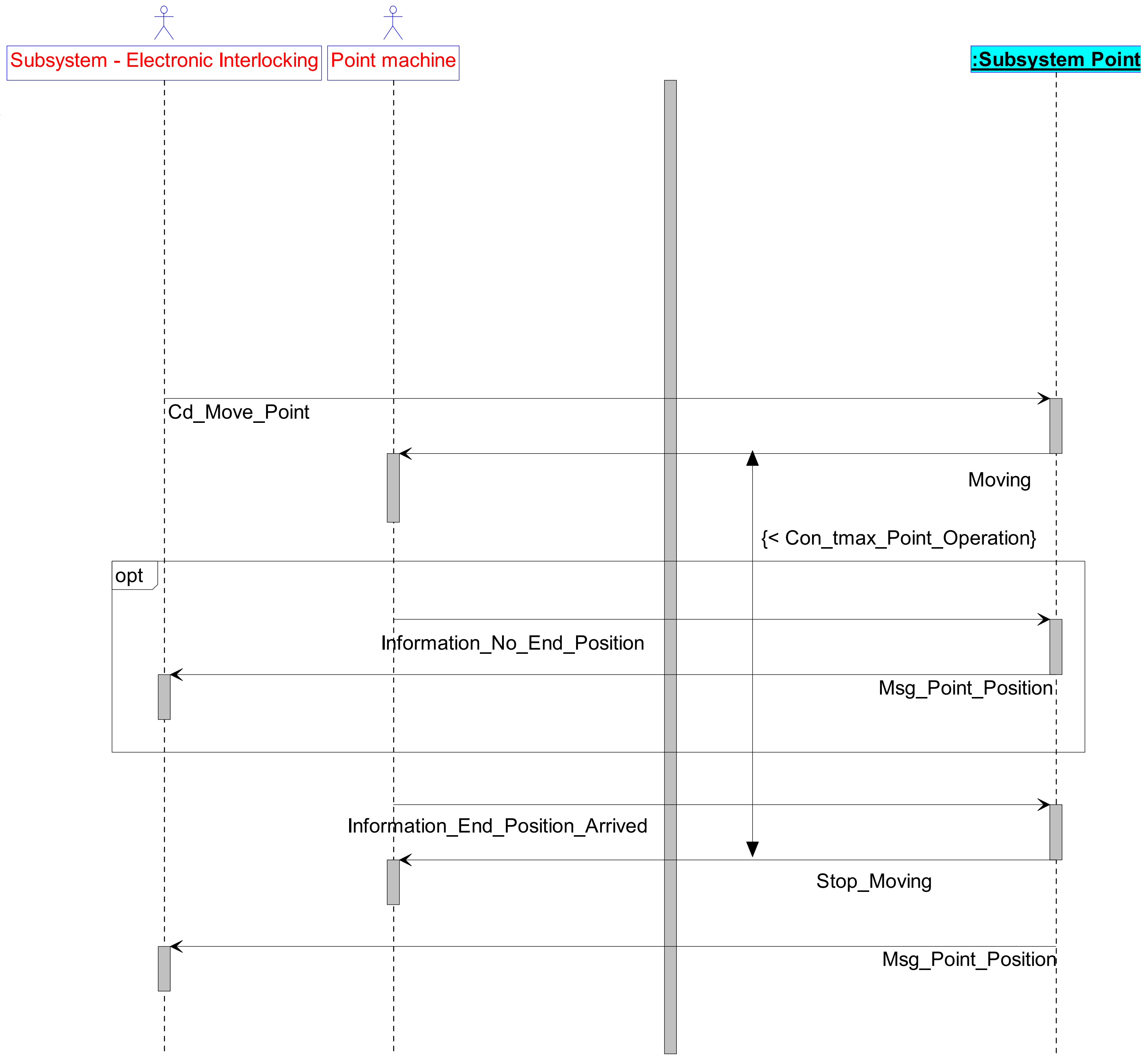
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.5784	Info	P SD 1.3.1.1.2 **P_UC1.3: Report status** Main Success Scenario: Report ability to move point status with single point machine [P SD 1.3.1.1.2] **Interaction 1.3.1.1.2.A** **alt** [The Point is Able to move point] **par** **1.a1.a1** - The Subsystem - Point receives from the Point machine the Information that the Point is Able to move point. **also par** **1.a1.a2** The Subsystem - Point receives from the internal trigger the Information that the Point is Able to move point. **end par** **else alt** [The Point is Unable to move point] **alt** [The Point machine is Unable to move point.] **1.b1.a1** - The Subsystem - Point receives from the Point machine the Information that the Point is Unable to move point. **else alt** [The internal trigger indicates Unable to move point] **1.b1.b1** The Subsystem - Point receives from the internal trigger the Information that the Point is Unable to move point. **end alt** **end alt** **Interaction 1.3.1.1.2 B** **2.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking the Ability to move point.  Subsystem - Electronic Interlocking Point machine :Subsystem Point alt par Information_Ability_To_Move_Point Information_Ability _To_Move_Point_Available_TRUE alt Information_Ability_To_Move_Point Information_Ability _To_Move_Point_Available_FALSE Msg_Ability_To_Move_Point	Only applicable if the package [Option Able to move] is used in combination with [Basic 4-wire single P] or [Basic non-4-wire single P].	Option Able to move
Eu.P.5789	Info	P SD 1.3.1.2 **P_UC1.3: Report status** Main Success Scenario: Report status with multiple point machines [P SD 1.3.1.2] **par** **par** **1.a1.a1** The Subsystem - Point receives the point position status from the Point machines **also par** **1.a1.a2** The Subsystem - Point receives the degraded point position status from the Point machines **end par** **1.a2** - The Subsystem - Point reports to the Subsystem - Electronic Interlocking the Point position. **also par** **1.b1** The Subsystem - Electronic Interlocking receives the ability to move status of the Subsystem - Point **end par**  Subsystem - Electronic Interlocking Point machine Point machine 1st n-th :Subsystem Point par ref Main Success Scenario: Report point position status with multiple point machines [P SD 1.3.1.2.1] ref Main Success Scenario: Report degraded point position status with multiple point machines [P SD 1.3.1.2.2] Msg_Point_Position ref Main Success Scenario: Report ability to move point status with multiple point machines [P SD 1.3.1.2.3]	If a state change happens while establishing the PDI connection and the status report Msg_Point_Position has already been sent, a new status report Msg_Point_Position has to be sent to Subsystem - Electronic Interlocking immediately after completion of the establishment of the connection.	Basic 4-wire multiple P Basic non-4-wire multiple P

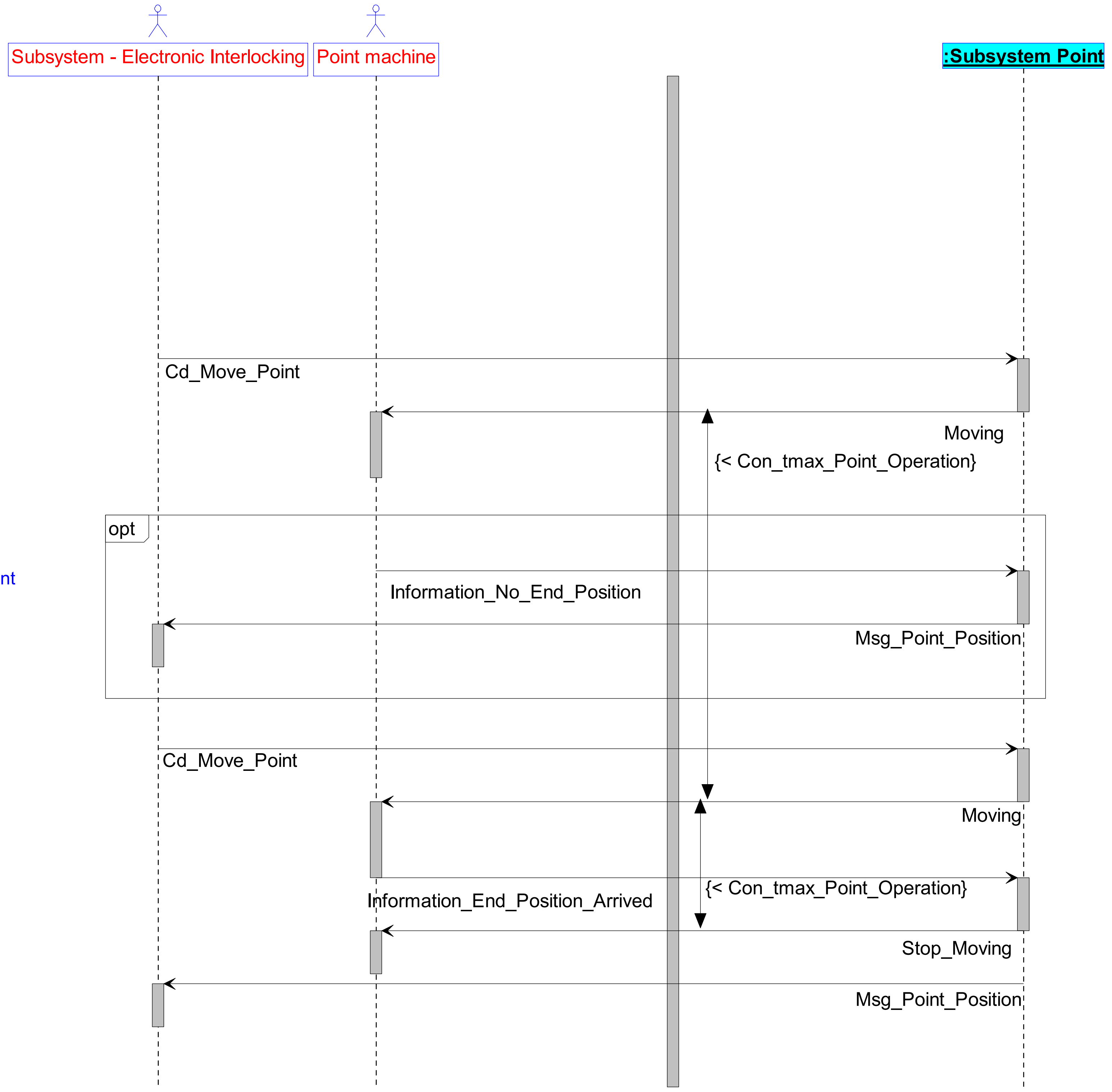
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.5786	Info	P SD 1.3.1.2.1 **P_UC1.3: Report status** Main Success Scenario: Report point position status with multiple point machines [P SD 1.3.1.2.1] **Interaction 1.3.1.2.1.A** **alt** [The Subsystem - Point is configured with a non-4-wire 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'] **par** **1.a1.a1** - The Subsystem - Point receives from the 1st Point machine the Information that the Point is in an End position 'Y'. **also par** **1.a1.a2** - The Subsystem - Point receives from the n-th Point machine the Information that the Point is in an End position 'Y'. **end par** **else alt** [The Point is in a Unintended position] **alt** [The 1st Point machine is in a Unintended position.] **1.b1.b1** - The Subsystem - Point receives from the 1st Point machine the Information that the Point is in a Unintended position. **else alt** [The n-th Point machine is in a Unintended position.] **1.b1.b2** - The Subsystem - Point receives from the n-th Point machine the Information that the Point is in a Unintended position. **end alt** **else alt** [The Point is in No end 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.c1** - The Point is in No end position. **end alt**  Subsystem - Electronic Interlocking Point machine Point machine :Subsystem Point 1st n-th alt alt par Information_End_Position_Arrived Information_End_Position_Arrived alt Information_Unintended_Position Information_Unintended_Position		Basic 4-wire multiple P Basic non-4-wire multiple P

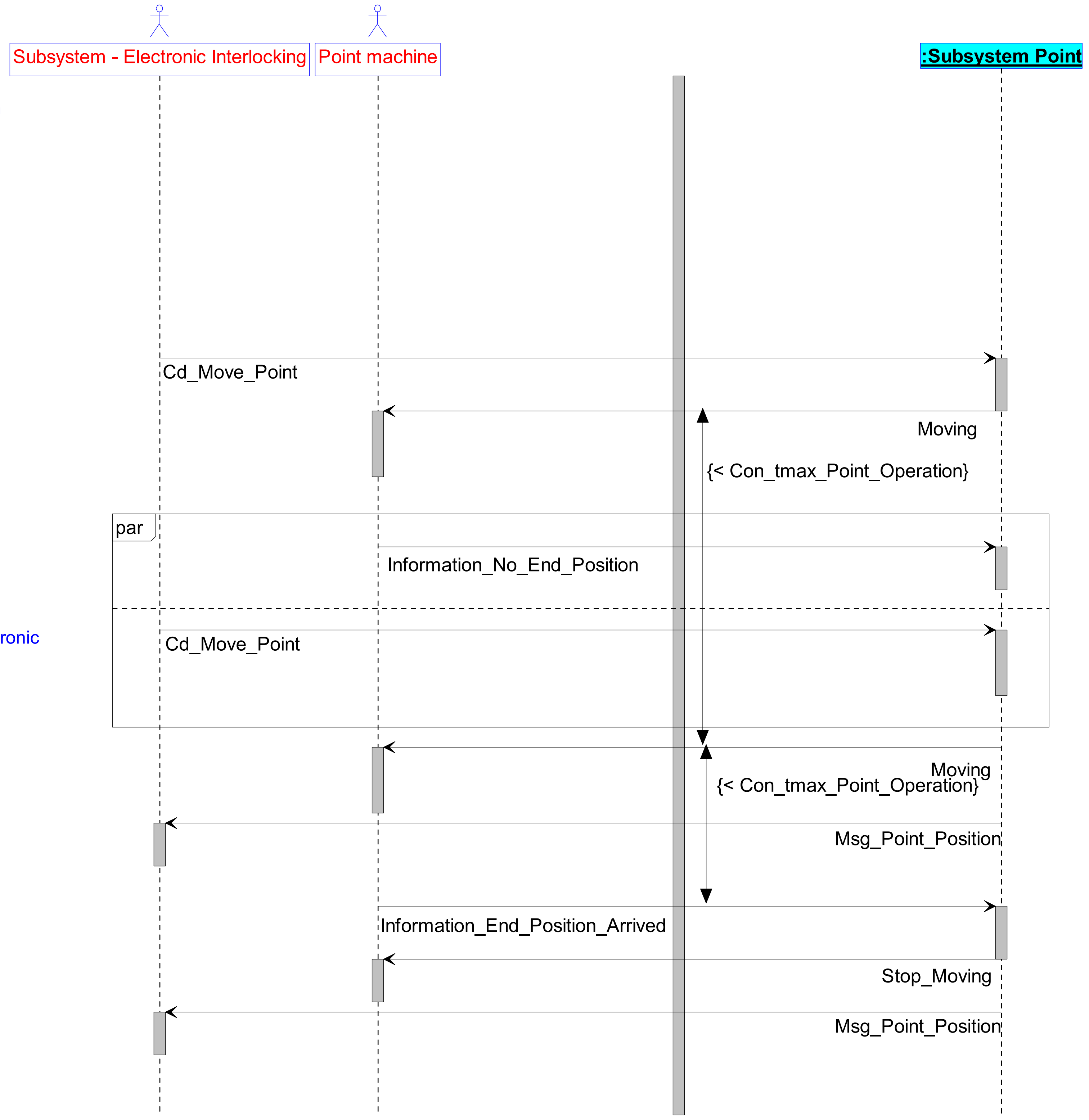
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.5785	Info	P SD 1.3.1.2.2 **P_UC1.3: Report status** Main Success Scenario: Report degraded point position status with multiple point machines [P SD 1.3.1.2.2] **Interaction 1.3.1.2.2.A** **alt** [The Subsystem - Point is configured with one or more non-crucial Point machines] **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 "X". The Point is in a Degraded point position "X"] **par** **1.a1.a1.a1** - The 1st crucial Point machine sends an Information to the Subsystem - Point indicating that the 1st crucial Point machine is in an is End position "X". **also par** **opt** [The Subsystem - Point is configured with more than one crucial Point machines] **1.a1.a1.b1.a1** - The n-th crucial Point machine sends an Information to the Subsystem - Point indicating that the n-th crucial Point machine is in an is End position "X". **end opt** **also par** **alt** **alt** **1.a1.a1.c1.a1** - The non-crucial 2nd Point machine sends an Information to the Subsystem - Point indicating that the non-crucial 2nd Point machine is in a Unintended position. **else alt** **1.a1.a1.c1.b1** - The non-crucial 2nd Point machine sends an Information to the Subsystem - Point indicating that the non-crucial 2nd Point machine is in No end position. **end alt** **else alt** [The Subsystem - Point is configured with more than one non-crucial Point machine] **alt** **1.c1.b1.a1** - The non-crucial n-th Point machine sends an Information to the Subsystem - Point indicating that the non-crucial n-th Point machine is in a Unintended position. **else alt** **1.c1.b1.b1** - The non-crucial n-th Point machine sends an Information to the Subsystem - Point indicating that the non-crucial n-th Point machine is in No end position. **end alt** **end alt** **also par** **1.a1.a1.d1** - The non-crucial 2nd Point machine is NOT in an End position "Y". **also par** [The Subsystem - Point is configured with a non-crucial n-th Point machine] **1.a1.a1.e1** - The non-crucial n-th Point machine is NOT in an End position "Y". **end par** **else alt** [The Point is not in a Degraded point position] **end alt** **else alt** [The Subsystem - Point is configured without non-crucial Point machines. The Degraded point position is not applicable.] **end alt**  Subsystem - Electronic Interlocking Point machine crucial 1st Point machine crucial n-th Point machine non-crucial 2nd Point machine non-crucial n-th :Subsystem Point alt alt par Information_End_Position_Arrived opt Information_End_Position_Arrived alt alt Information_Unintended_Position Information_No_End_Position alt Information_Unintended_Position Information_No_End_Position		Basic 4-wire multiple P Basic non-4-wire multiple P

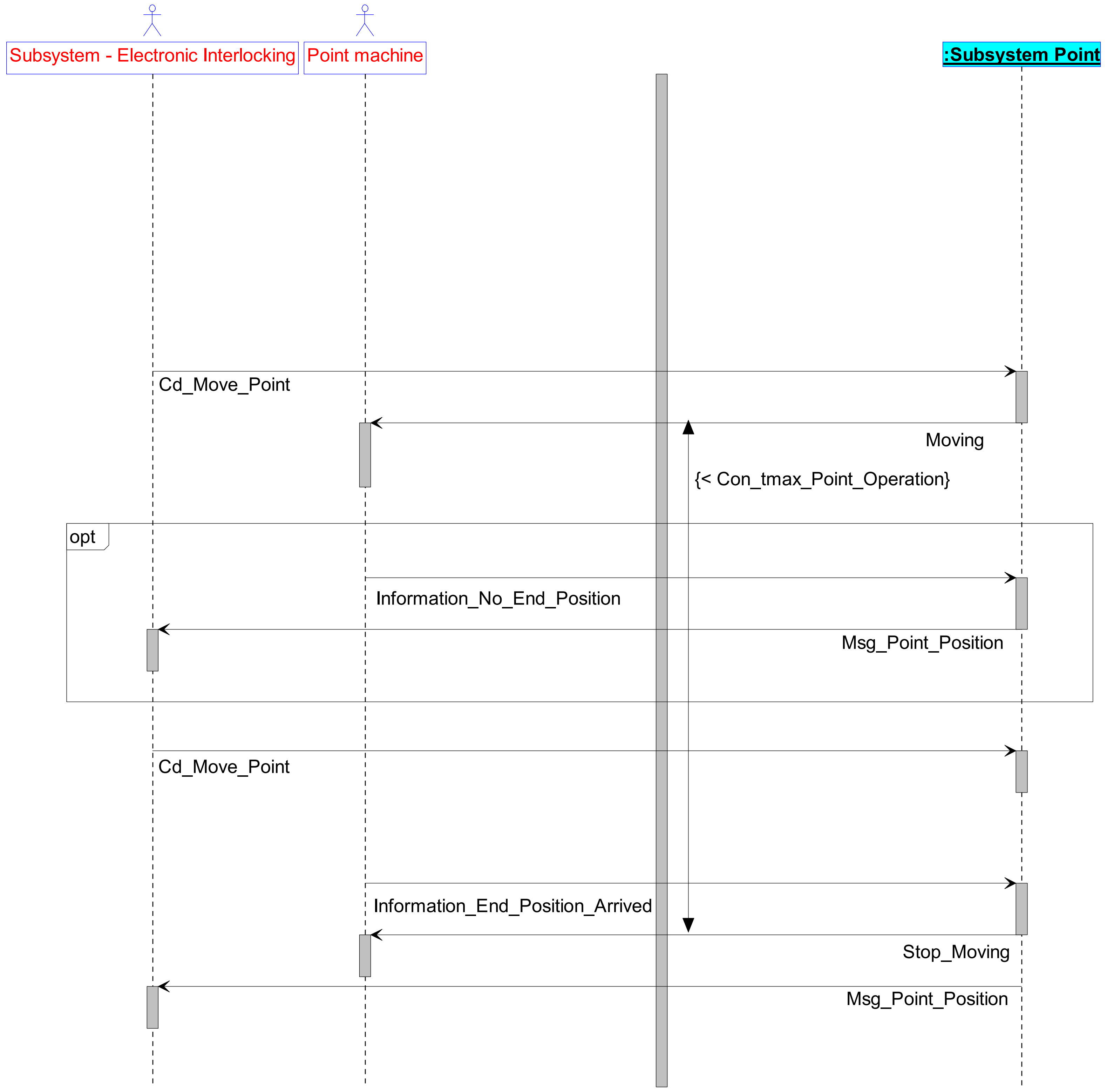
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.5783	Info	P SD 1.3.1.2.3 **P_UC1.3: Report status** Main Success Scenario: Report ability to move point status with multiple point machines [P SD 1.3.1.2.3] **Interaction 1.3.1.2.3.A** **alt** [The Point is Able to move point] **par** **1.a1.a1** - The Subsystem - Point receives from the 1st Point machine the Information that the 1st Point machine is Able to move point. **also par** **opt** [The n-th Point machine is configured with full functionality] **1.a1.b1.a1** - The Subsystem - Point receives from the n-th Point machine the Information that the n-th Point machine Able to move point. **end opt** **also par** **1.a1.c1** - The Subsystem - Point internal trigger Ability to move point available indicates availability of Ability to move point. **end par** **else alt** [The Point is Unable to move point] **alt** [The 1st Point machine is Unable to move point.] **1.b1.a1** - The Subsystem - Point receives from the 1st Point machine the Information that the 1st Point machine Unable to move point. **else alt** [The n-th Point machine is configured with full functionality and is Unable to move point.] **1.b1.b1** - The Subsystem - Point receives from the n-th Point machine the Information that the n-th Point machine Unable to move point. **else alt** [The internal trigger indicates Unable to move point] **1.b1.c1** - The Subsystem - Point internal trigger Ability to move point available indicates non-availability of Ability to move point. **end alt** **end alt** **Interaction 1.3.1.2.3.B** **2.** - The Subsystem - Point reports to the Subsystem - Electronic Interlocking the Ability to move point. **opt**  Subsystem - Electronic Interlocking Point machine Point machine 1st n-th :Subsystem Point alt par Information_Ability_To_Move_Point opt Information_Ability_To_Move_Point Information_Ability_To_Move_Point_Available_TRUE alt Information_Ability_To_Move_Point Information_Ability_To_Move_Point Information_Ability_To_Move_Point_Available_FALSE Msg_Ability_To_Move_Point	Only applicable if the package [Option Able to move] is used in combination with [Basic 4-wire multiple P] or [Basic non-4-wire multiple P].	Option Able to move
Eu.P.1109	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.	Basic 4-wire single P Basic 4-wire multiple P Basic non-4-wire single P Basic non-4-wire multiple P
Eu.P.1466	Info	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.	Basic 4-wire single P Basic non-4-wire single P

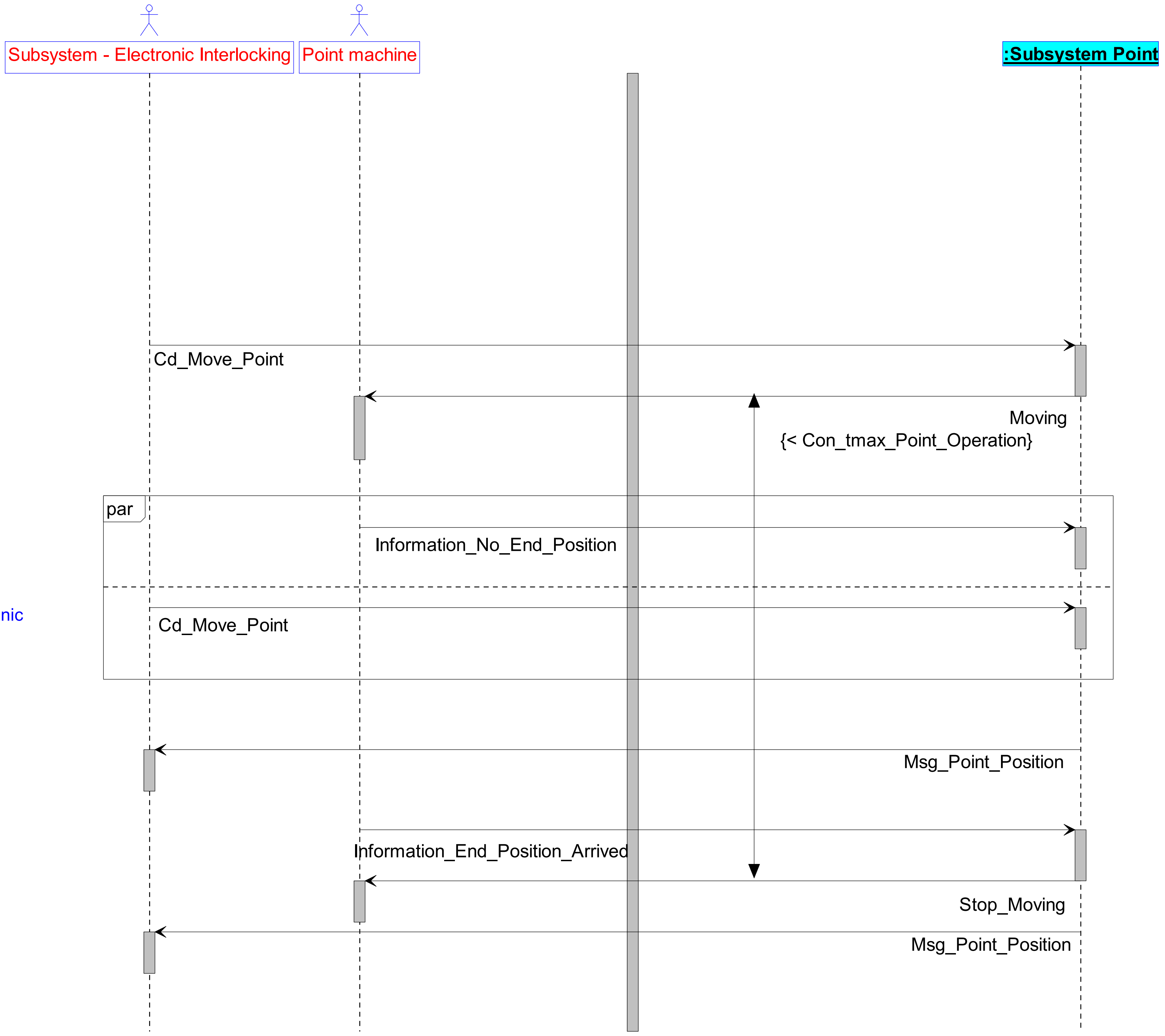
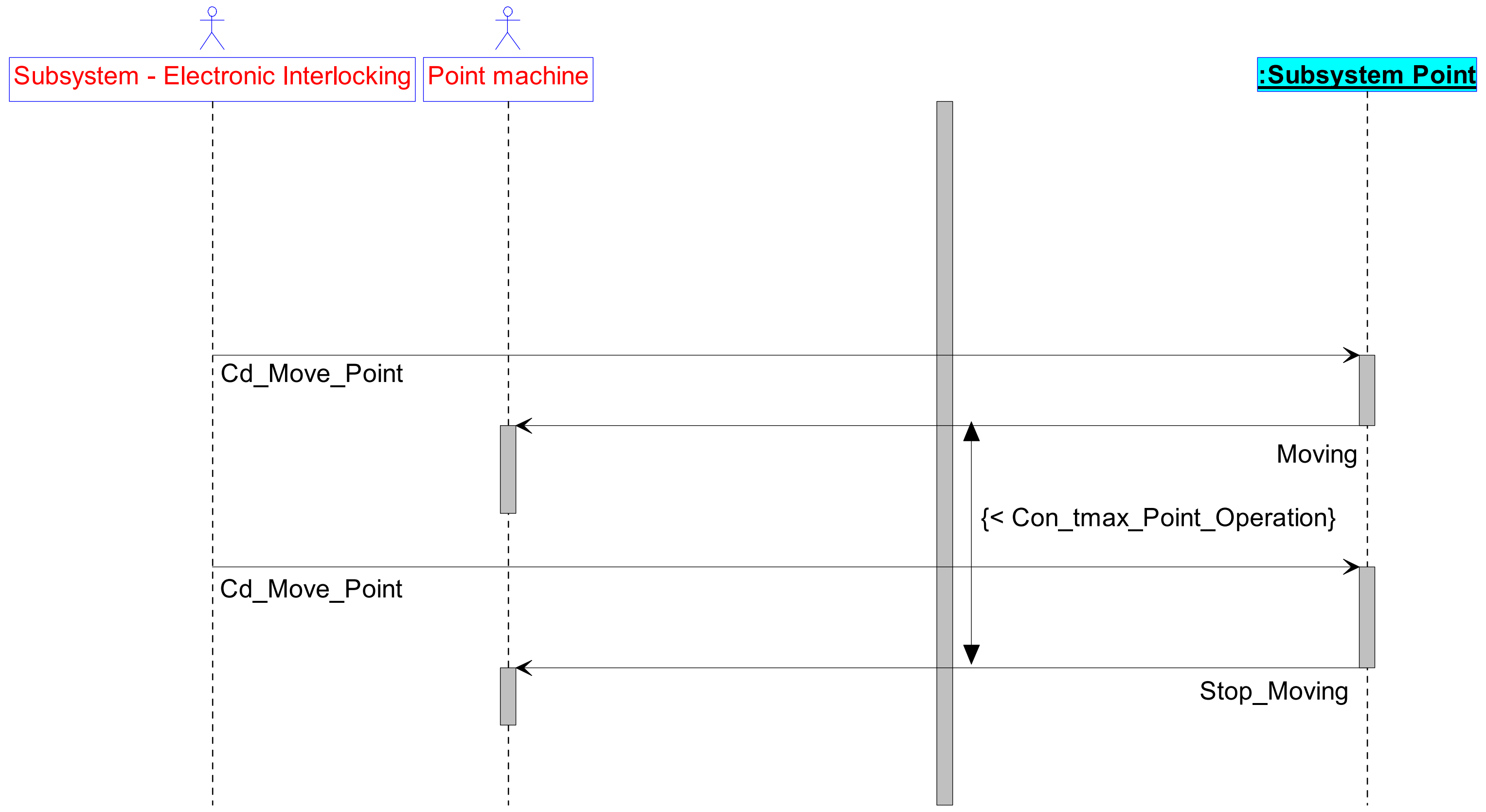
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.5790	Info	P SD 1.4.2 **P_UC1.4: Establish initial state of outputs** Main Success Scenario: Set Initial State of Outputs with multiple point machines [P SD 1.4.2] **Precondition:** The Subsystem - Point is in the state BOOTING. **Interaction 1.4.2.A:** **1.** - The Subsystem Point enters the state INITIALISING. par **2.a1** The Subsystem - Point sends the Command to the 1st Point machine to Stop moving *the* Point machine. also par **2.b1** The Subsystem - Point sends the Command to the n-th Point machine to Stop moving *the* Point machine. end par **Postconditions:** The Subsystem - Point is in the state INITIALISING. The Initial State Of Outputs of the Subsystem Point has been set.  Point machine 1st Point machine n-th :Subsystem Point par Stop_Moving Stop_Moving	Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE.	Basic 4-wire multiple P Basic non-4-wire multiple P
Eu.P.6706	Info	[Package] Subsystem Point - Functional Context [Functional Viewpoint - Subsystem Definition - Operation]  uc [Package] Subsystem Point - Functional Context [Functional Viewpoint - Subsystem Definition - Operation] Subsystem Point P_UC2.1: Point with non-4-wire interface P_UC2.1.1.1: Commanding and reversing P_UC2.1.1.2: Redrive P_UC2.1.1: Single point machine P_UC2.1.1.3: Irregularities P_UC2.1.2: Multiple point machines P_UC2.1.2.1: Commanding and reversing P_UC2.1.2.2: Redrive P_UC2.1.2.3: Stop point movement P_UC2.1.2.4: Irregularities Subsystem - Electronic Interlocking Point machine P_UC2.2.1.1: Commanding and reversing P_UC2.2.1: Single point machine P_UC2.2: Point with 4-wire interface P_UC2.2.1.2: Irregularities P_UC2.2.2: Multiple point machines P_UC2.3: Handle general Irregularities P_UC2.2.2.1: Commanding and reversing P_UC2.2.2.2: Irregularities		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P

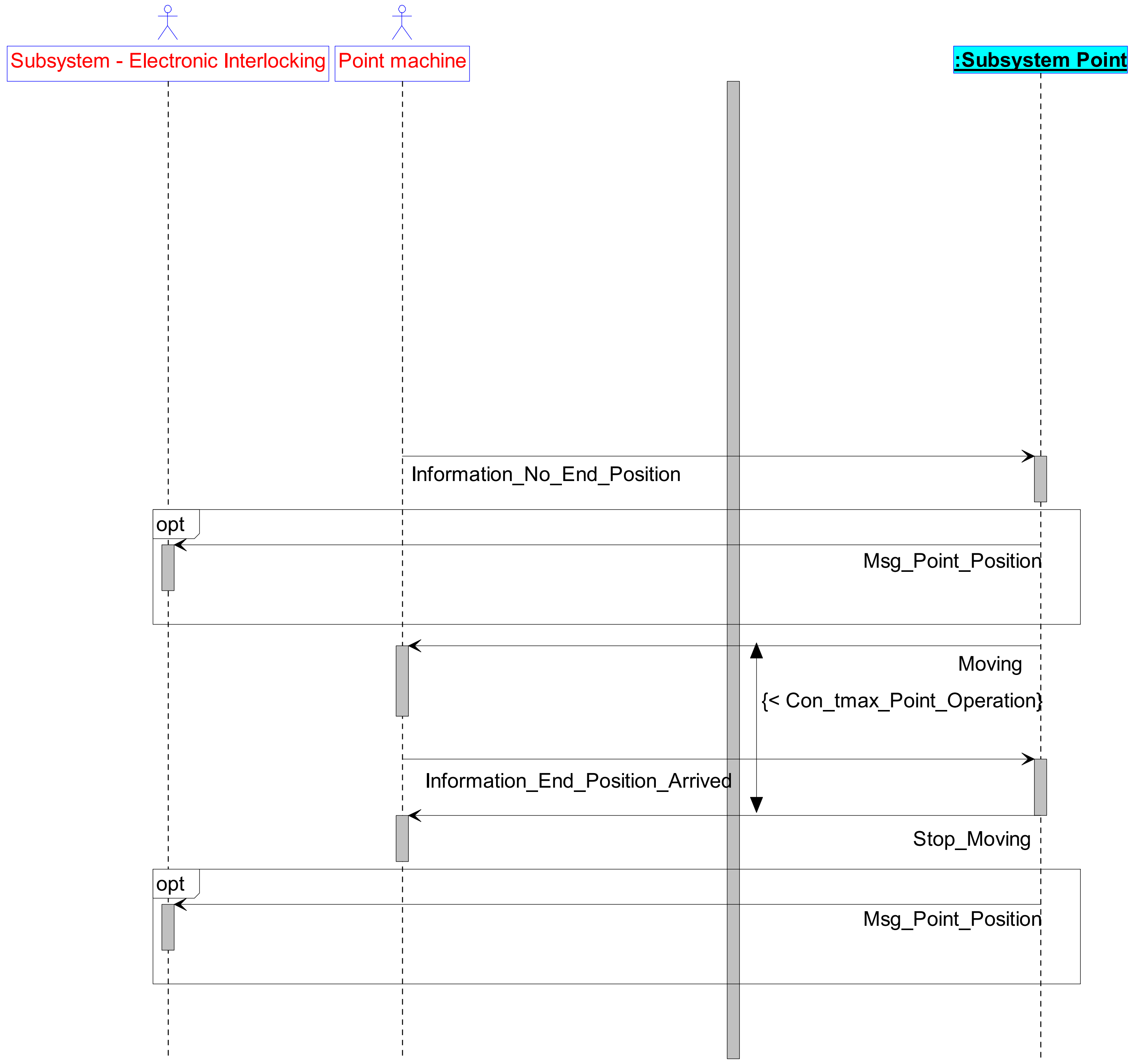
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.6732	Info	P_UC2.1: Point with non-4-wire interface	The Subsystem-UseCase "P_UC2.1: Point with non-4-wire interface" defines the behaviour of the Subsystem Point which works with a non-4-wire interface. The behaviour will be defined in the following UseCases: P_UC2.1.1: Single point machine P_UC2.1.2: Multiple point machines	Basic non-4-wire single P Basic non-4-wire multiple P
Eu.P.6720	Info	P_UC2.1.1: Single point machine	The Subsystem-UseCase "P_UC2.1.1: 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_UC2.1.1.1: Commanding and reversing P_UC2.1.1.2: Redrive P_UC2.1.1.3: Irregularities	Basic non-4-wire single P
Eu.P.6707	Info	P_UC2.1.1.1: Commanding and reversing	The Subsystem-UseCase "P_UC2.1.1.1: Commanding and reversing" defines the behaviour of commanding and reversing a single point machine via non-4-wire interface.	Basic non-4-wire single P
Eu.P.6714	Info	P SD 2.1.1.1.1 **P_UC2.1.1.1: Commanding and reversing** Main Success Scenario: Moving of the Point with a single point machine Non 4W [P SD 2.1.1.1.1] **Precondition:** The Subsystem - Point is in the state OPERATIONAL. The Subsystem - Point is configured with a non-4-wire interface to the Point machine. The Subsystem - Point is in: - an End position "Y", or - No end position, or - an Unintended position. **Interaction 2.1.1.1.1.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 the timer Con_tmax_Point_Operation. **Interaction 2.1.1.1.1.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 2.1.1.1.1.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. **6.** The timer Con_tmax_Point_Operation is reset. **7.** 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".  Subsystem - Electronic Interlocking Point machine :Subsystem Point Cd_Move_Point Moving {< Con_tmax_Point_Operation} opt Information_No_End_Position Msg_Point_Position Information_End_Position_Arrived Stop_Moving Msg_Point_Position	Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE.	Basic non-4-wire single P

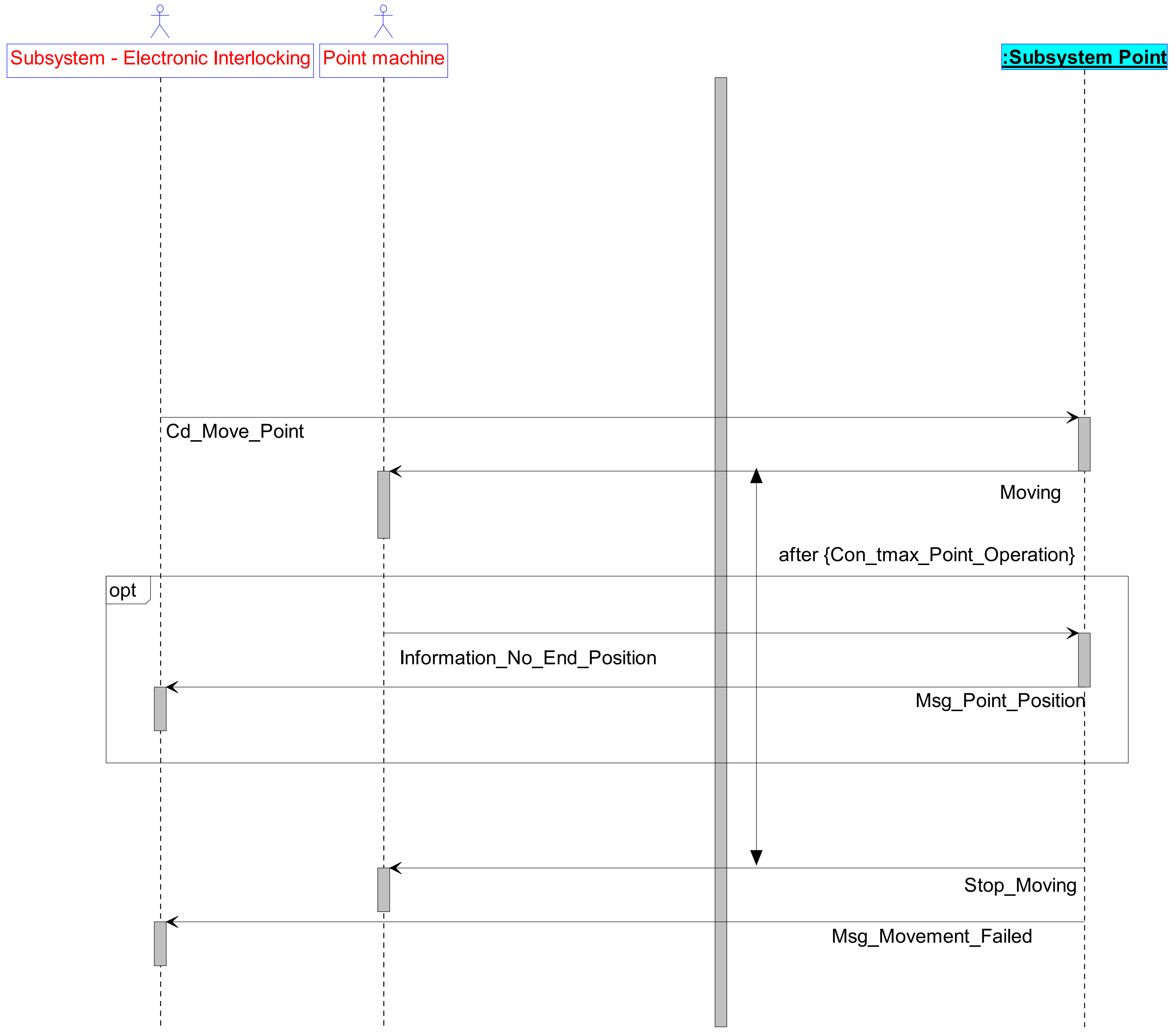
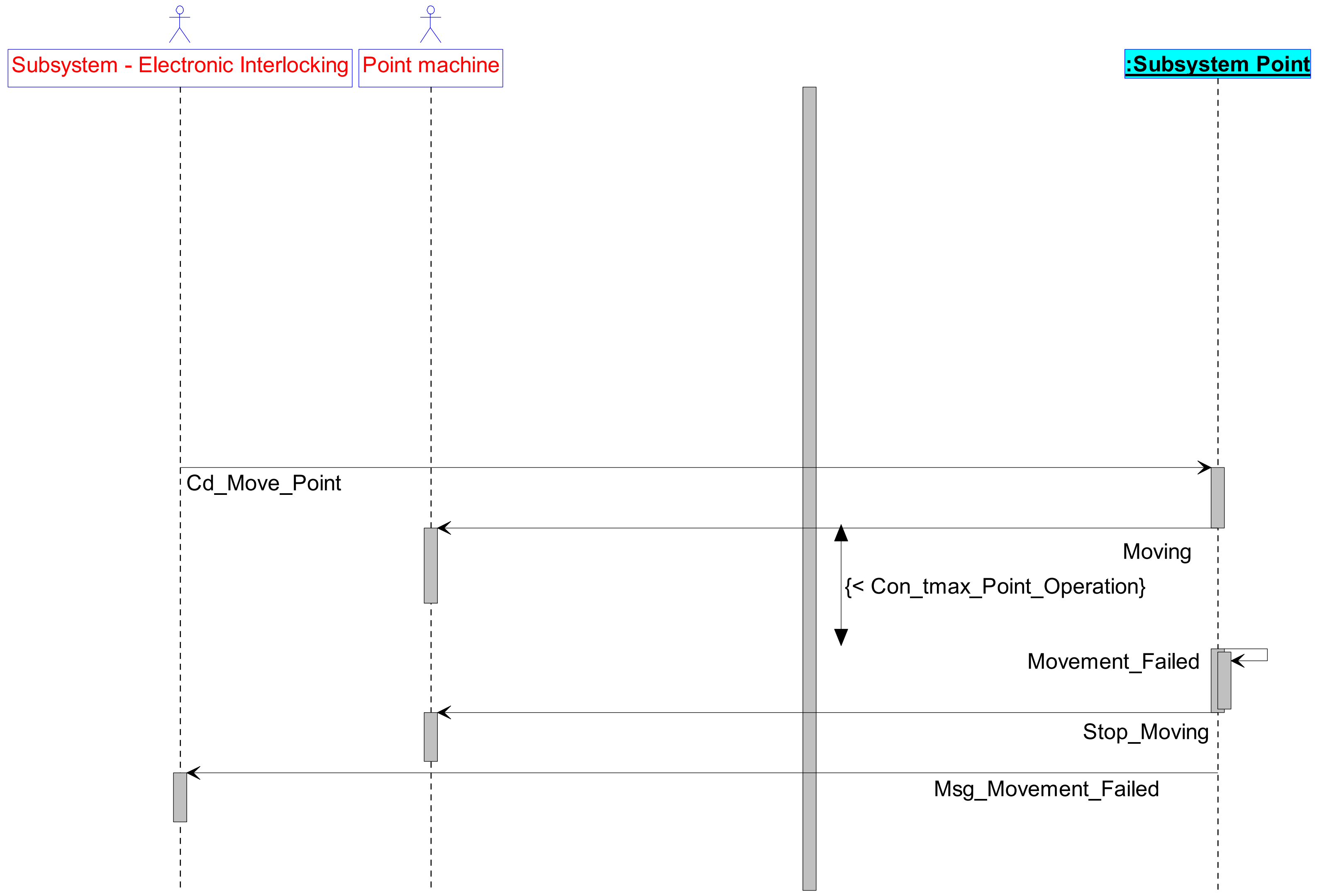
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.6713	Info	P SD 2.1.1.1.2 **P_UC2.1.1.1: Commanding and reversing** Alternative Scenario: Reversing Point Non 4W [P SD 2.1.1.1.2] **Precondition:** The Subsystem - Point is in the state OPERATIONAL. The Subsystem - Point is configured with a non-4-wire interface to the Point machine. The Subsystem - Point is in: - an End position "Y", - No end position, or - an Unintended position. **Interaction 2.1.1.1.2.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 the timer Con_tmax_Point_Operation. **Interaction 2.1.1.1.2.B:** **opt** [The Subsystem Point was previously in an End position or a 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 2.1.1.1.2.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 re-starts the timer 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. **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".  Subsystem - Electronic Interlocking Point machine :Subsystem Point Cd_Move_Point Moving {< Con_tmax_Point_Operation} opt Information_No_End_Position Msg_Point_Position Cd_Move_Point Moving {< Con_tmax_Point_Operation} Information_End_Position_Arrived Stop_Moving Msg_Point_Position	Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE.	Basic non-4-wire single P

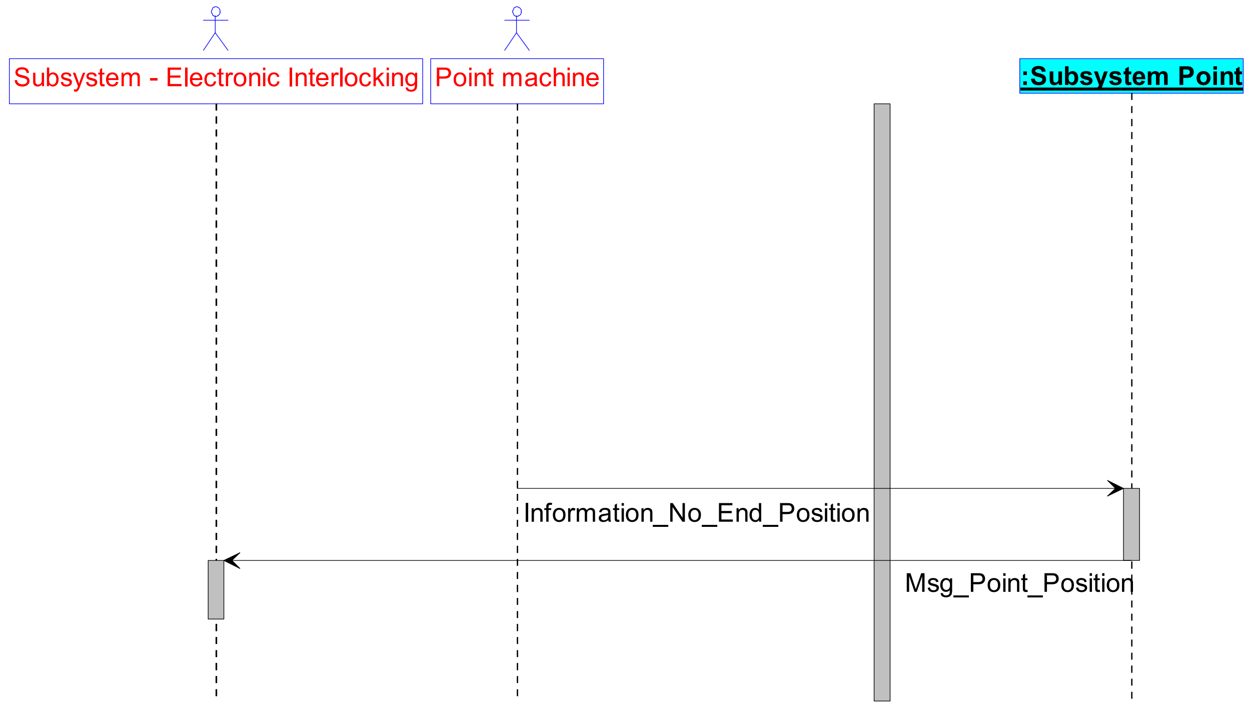
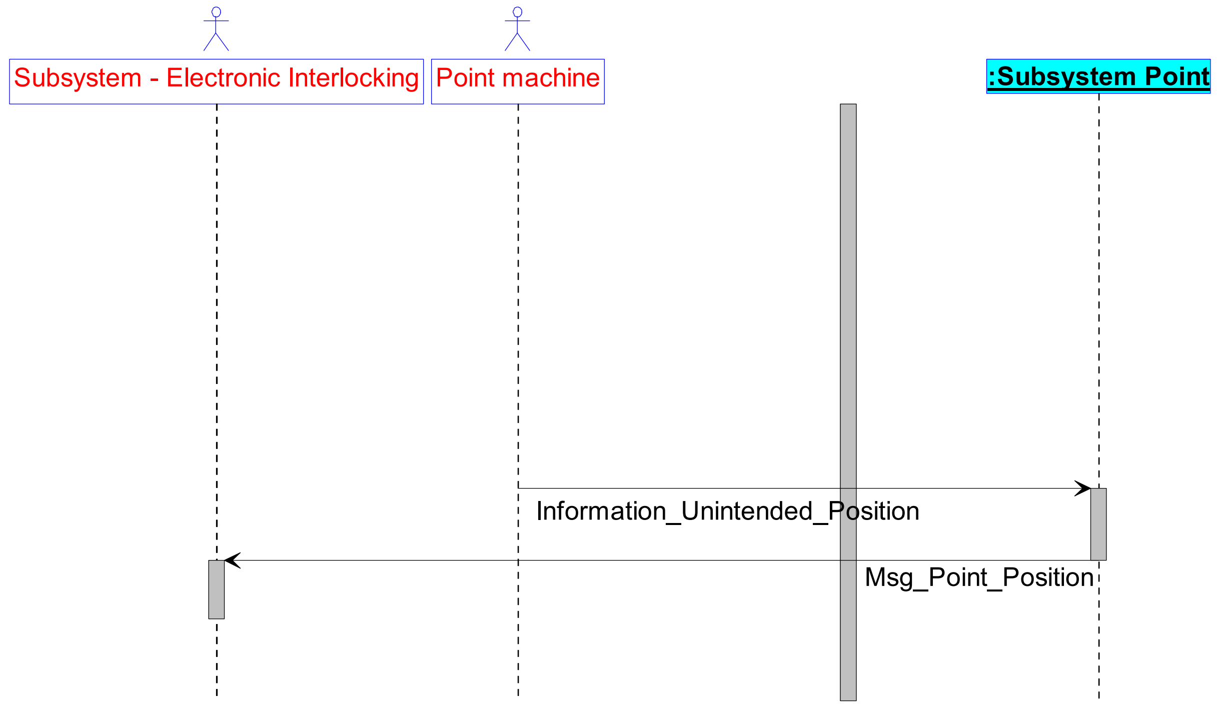
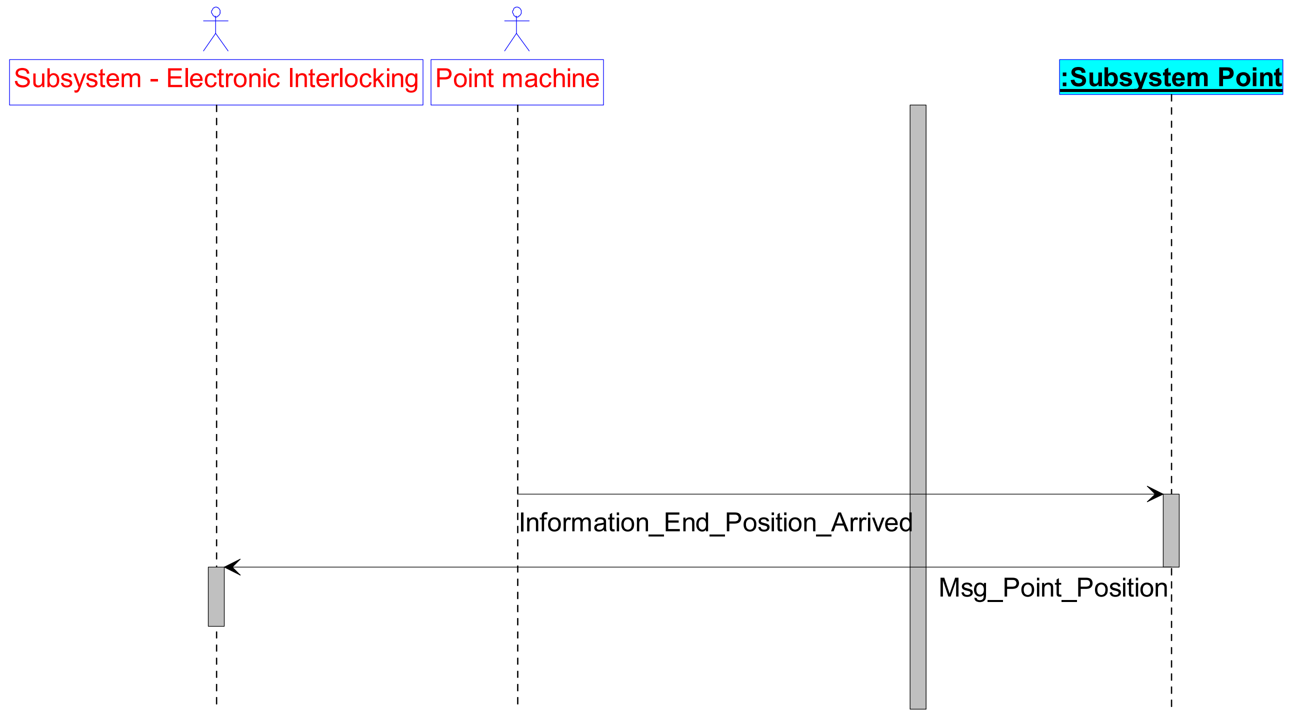
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.6711	Info	P SD 2.1.1.1.3 **P UC2.1.1.1: Commanding and reversing**  Subsystem - Electronic Interlocking Point machine :Subsystem Point Cd_Move_Point Moving {< Con_tmax_Point_Operation} par Information_No_End_Position Cd_Move_Point Moving {< Con_tmax_Point_Operation} Msg_Point_Position Information_End_Position_Arrived Stop_Moving Msg_Point_Position Alternative Scenario: Reversing Point directly after the position has been commanded Non 4W [P SD 2.1.1.1.3] **Precondition:** The Subsystem - Point is in the state OPERATIONAL. The Subsystem - Point is configured with a non-4-wire interface to the Point machine. The Subsystem - Point is in: - an End position "Y", or - an Unintended position. **Interaction 2.1.1.1.3.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 the timer Con_tmax_Point_Operation. **Interaction 2.1.1.1.3.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 re-starts the timer Con_tmax_Point_Operation. **5.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in No end position. **Interaction 2.1.1.1.3.D:** **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. **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".	Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE.	Basic non-4-wire single P

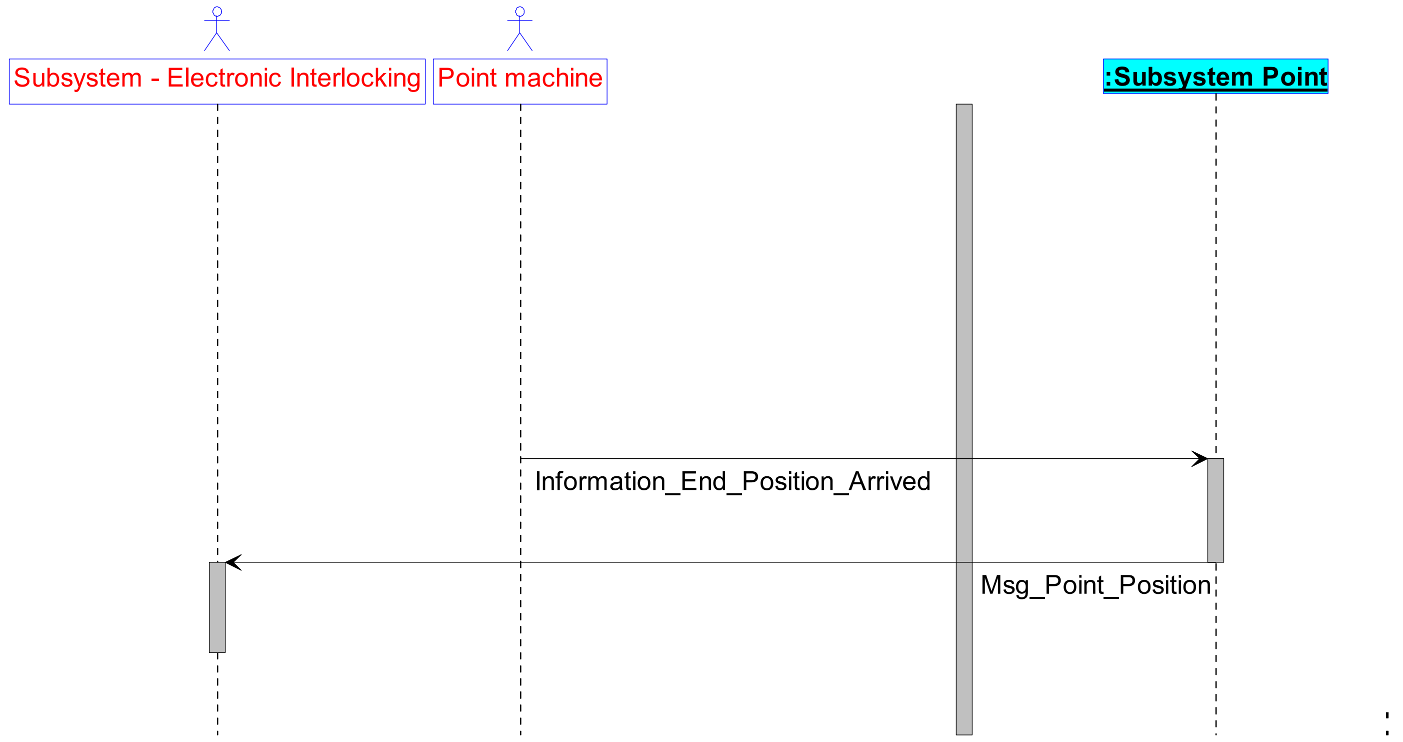
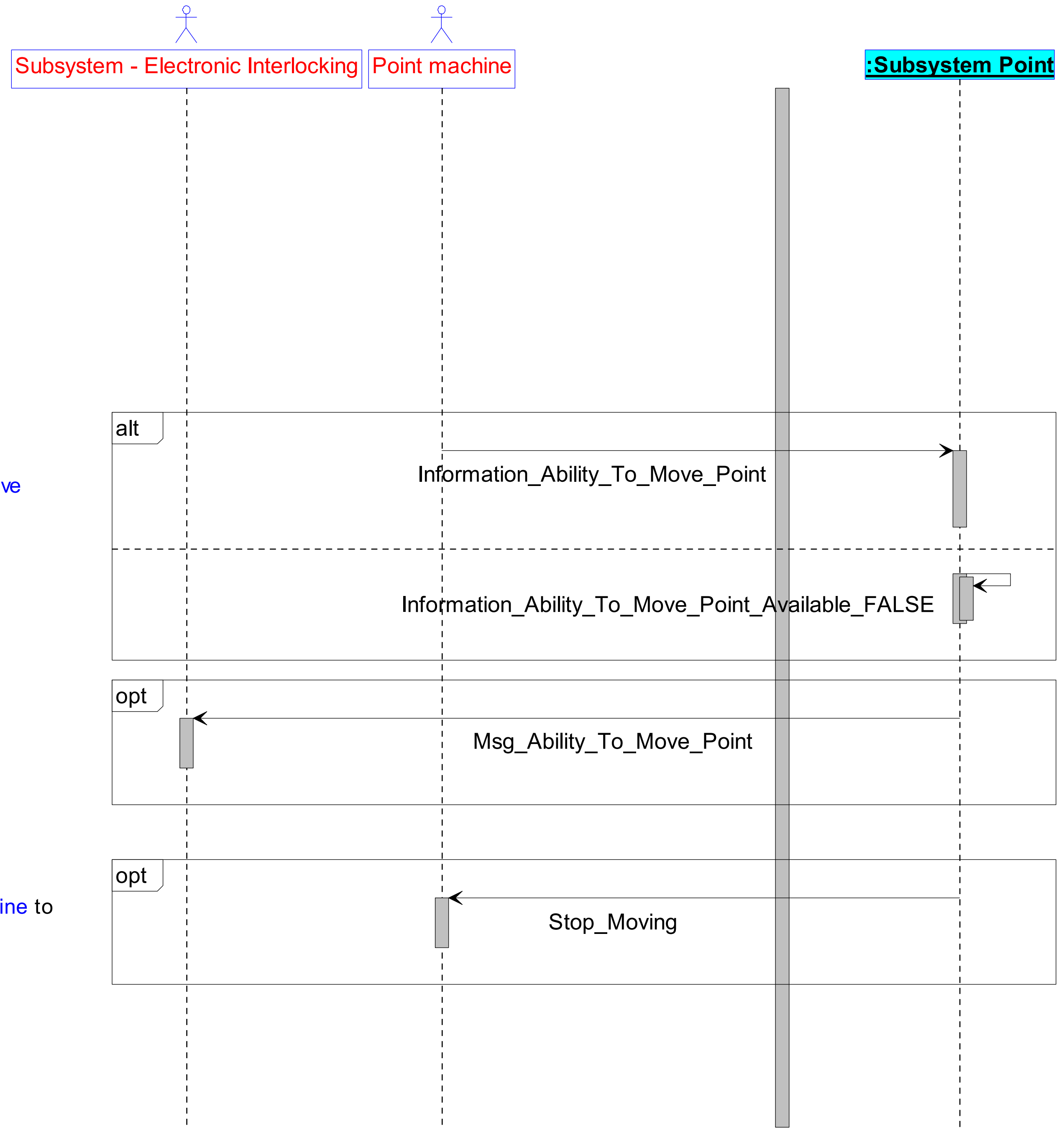
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.6708	Info	P SD 2.1.1.1.4 **P_UC2.1.1.1: Commanding and reversing** Alternative Scenario: Moving of the Point with repeated command of moving #1 Non 4W [P SD 2.1.1.1.4] **Precondition:** The Subsystem - Point is in the state OPERATIONAL. The Subsystem - Point is configured with a non-4-wire interface to the Point machine. The Subsystem - Point is in: - an End position "Y", or - No end position, or - an Unintended position. **Interaction 2.1.1.1.4.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 the timer Con_tmax_Point_Operation. **Interaction 2.1.1.1.4.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 2.1.1.1.4.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 2.1.1.1.4.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. **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".  Subsystem - Electronic Interlocking Point machine :Subsystem Point Cd_Move_Point Moving {< Con_tmax_Point_Operation} opt Information_No_End_Position Msg_Point_Position Cd_Move_Point Information_End_Position_Arrived Stop_Moving Msg_Point_Position	Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE.	Basic non-4-wire single P

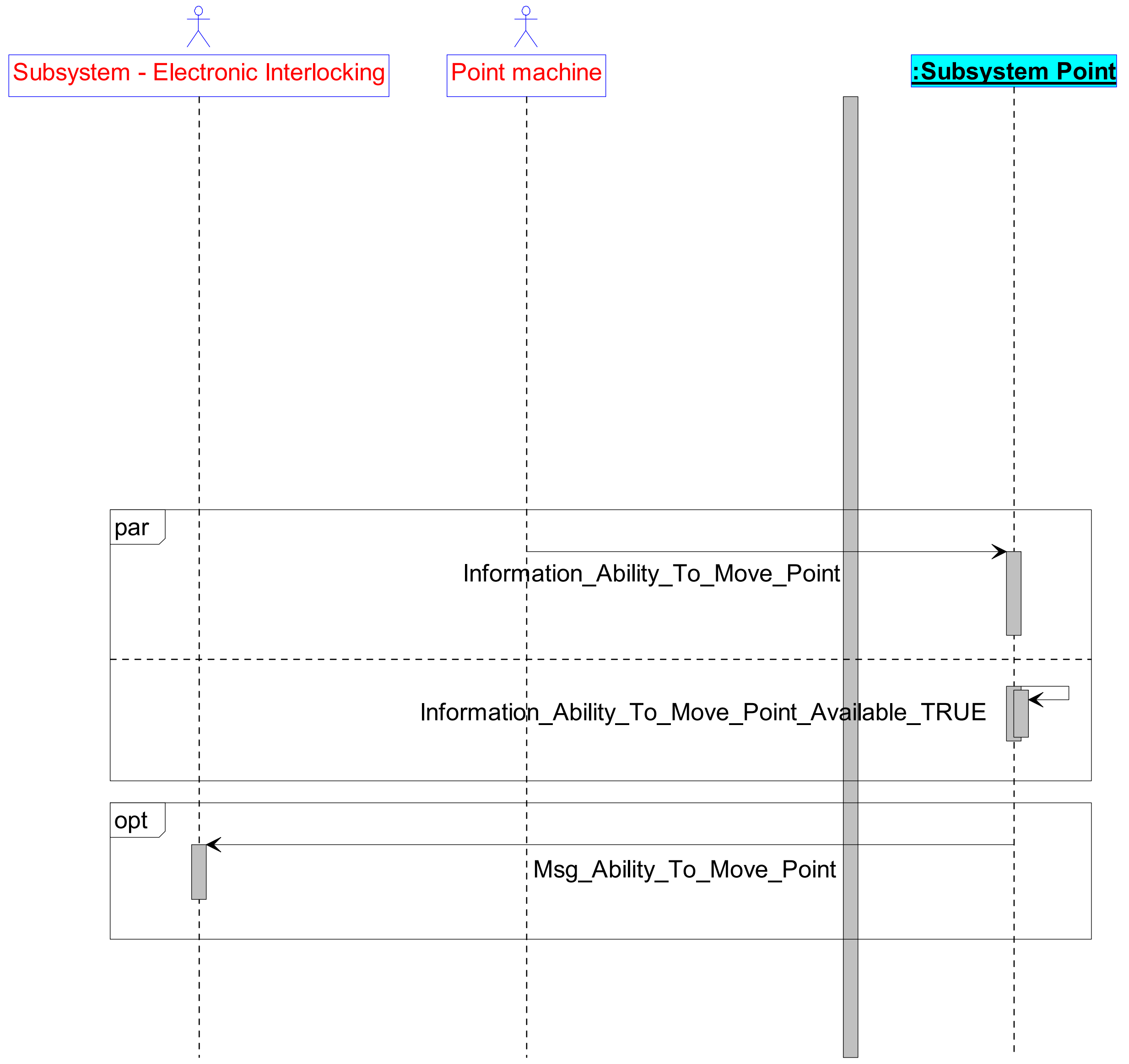
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.6709	Info	P SD 2.1.1.1.5 **P_UC2.1.1.1: Commanding and reversing** Alternative Scenario: Moving of the Point with repeated command of moving #2 Non 4W [P SD 2.1.1.1.5] **Precondition:** The Subsystem - Point is in the state OPERATIONAL. The Subsystem - Point is configured with a non-4-wire interface to the Point machine. The Subsystem - Point is in: - an End position "Y" or - an Unintended position. **Interaction 2.1.1.1.5.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 the timer Con_tmax_Point_Operation. **Interaction 2.1.1.1.5.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 2.1.1.1.5.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. **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".  Subsystem - Electronic Interlocking Point machine :Subsystem Point Cd_Move_Point Moving {< Con_tmax_Point_Operation} par Information_No_End_Position Cd_Move_Point Msg_Point_Position Information_End_Position_Arrived Stop_Moving Msg_Point_Position	Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE.	Basic non-4-wire single P
Eu.P.6712	Info	P SD 2.1.1.1.6 **P_UC2.1.1.1: Commanding and reversing** Alternative Scenario: Reversing Point directly without position change Non 4W [P SD 2.1.1.1.6] **Precondition:** The Subsystem - Point is in the state OPERATIONAL. The Subsystem - Point is in an End position "Y". **Interaction 2.1.1.1.7.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 the timer Con_tmax_Point_Operation. **Interaction 2.1.1.1.7.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 timer Con_tmax_Point_Operation is reset. **Postcondition:** The Subsystem - Point is in an End position "Y".  Subsystem - Electronic Interlocking Point machine :Subsystem Point Cd_Move_Point Moving {< Con_tmax_Point_Operation} Cd_Move_Point Stop_Moving	Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE.	Basic non-4-wire single P
Eu.P.6715	Info	P_UC2.1.1.2: Redrive	The Subsystem-UseCase "P_UC2.1.1.2: Redrive" defines the behaviour of redriving a single point machine via non-4-wire interface. Only applicable if the package [Option Redrive] is used in combination with [Basic non-4-wire single P].	Option Redrive

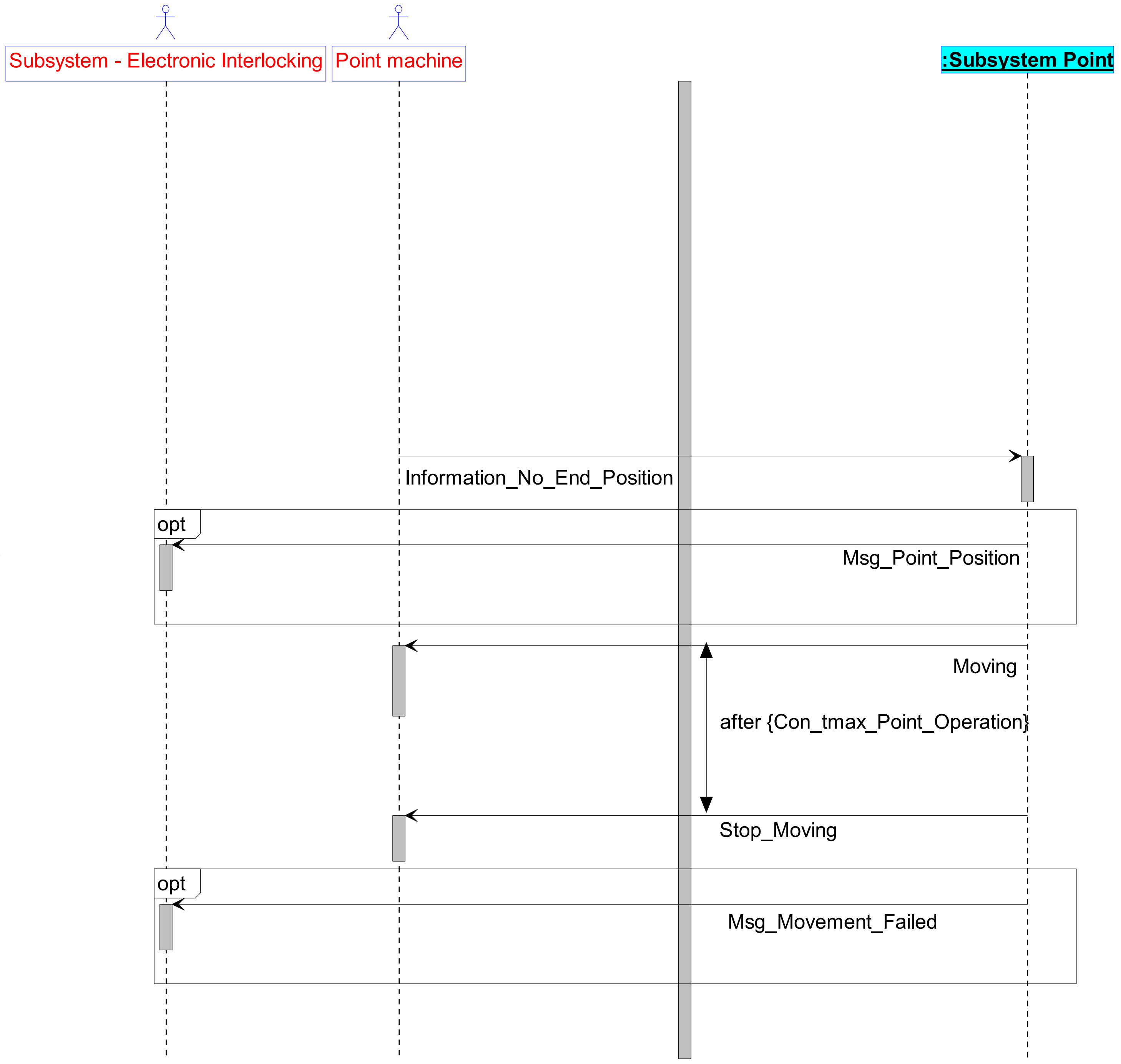
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.6717	Info	P SD 2.1.1.2.1 **P_UC2.1.1.2: Redrive** Main Success Scenario: Redrive Point after lost end position Non 4W [P SD 2.1.1.2.1] **Precondition:** The Subsystem Point is in the state INITIALISING or OPERATIONAL. The Subsystem - Point is in an End position "X". The last Cd_Move_Point Command received was also for End position "X". The last Cd_Move_Point Command received occurred since exiting the state BOOTING. The Subsystem - Point is configured with a non-4-wire interface to the Point machine. The Subsystem - Point is configured as Redrive point. **Interaction 2.1.1.2.1.A:** **1.** - The Subsystem - Point receives from the Point machine the Information that the Point machine is in No end position. **opt** [The Subsystem Point is in the state OPERATIONAL.] **2.a1** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in No end position. **end opt** **3.** The Subsystem - Point sends the Command to the Point machine to move the Point back to an End position "X". At this moment the Subsystem - Point starts the timer Con_tmax_Point_Operation. **Interaction 2.1.1.2.1.B:** **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. **opt** [The Subsystem Point is in the state OPERATIONAL.] **6.a1** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in an End position "X". **end opt** **Postcondition:** The Subsystem - Point is in an End position "X".  Subsystem - Electronic Interlocking Point machine :Subsystem Point Information_No_End_Position opt Msg_Point_Position Moving {< Con_tmax_Point_Operation} Information_End_Position_Arrived Stop_Moving opt Msg_Point_Position	Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE. Only applicable if the package [Option Redrive] is used in combination with [Basic non-4-wire single P].	Option Redrive
Eu.P.6718	Info	P_UC2.1.1.3: Irregularities	The Subsystem-UseCase "P_UC2.1.1.3: Irregularities" defines the behaviour of the Subsystem Point which works with a single point machine via non-4-wire interface, when an irregularity occurs.	Basic non-4-wire single P

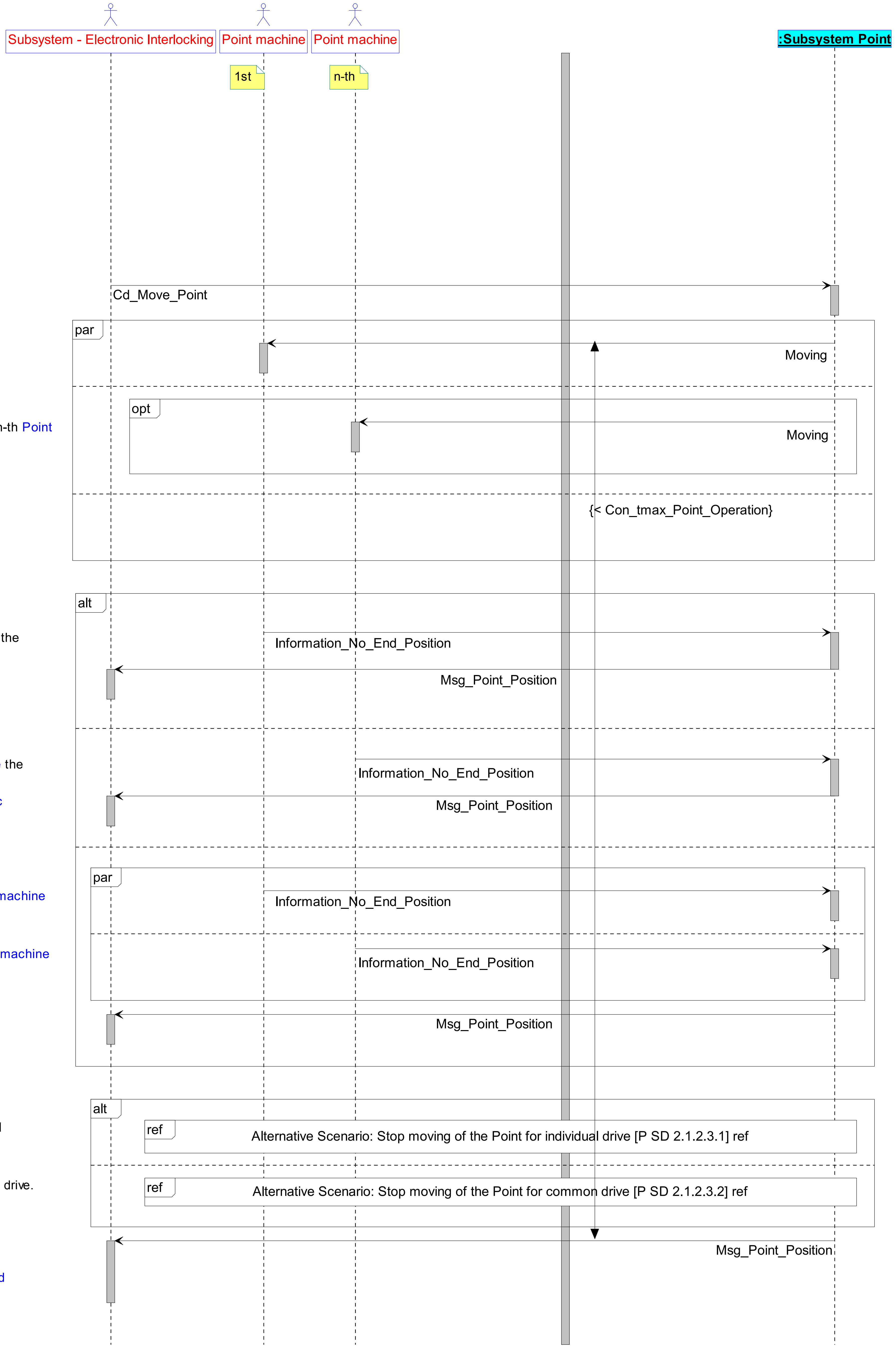
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.1284	Info	P SD 2.1.1.3.1 **P_UC2.1.1.3: Irregularities** Alternative Scenario: Handle and report Point operation timeout with position change with single point machine [P SD 2.1.1.3.1] **Precondition:** The Subsystem - Point is in the state OPERATIONAL. The Subsystem - Point is configured with a non-4-wire interface to the Point machine. The Subsystem - Point is in: - an End position "Y", - No end position, or - a Unintended position. **Interaction 2.1.1.3.1.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 the timer Con_tmax_Point_Operation. **Interaction 2.1.1.3.1.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 2.1.1.3.1.C:** **4.** - The timer Con_tmax_Point_Operation expires resulting in a Failed Movement. **5.** The Subsystem - Point sends the Command to the Point machine to stop moving the Point machine. **6.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that a Failed Movement has occurred. **Postcondition:** The Subsystem - Point is in No end position.  Subsystem - Electronic Interlocking Point machine :Subsystem Point Cd_Move_Point Moving after {Con_tmax_Point_Operation} opt Information_No_End_Position Msg_Point_Position Stop_Moving Msg_Movement_Failed	Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE.	Basic non-4-wire single P
Eu.P.6719	Info	P SD 2.1.1.3.2 **P_UC2.1.1.3: Irregularities** Alternative Scenario: Handle and report failed movement with single point machine without position change [P SD 2.1.1.3.2] **Precondition:** The Subsystem - Point is in the state OPERATIONAL. The Subsystem - Point is configured with a non-4-wire interface to the Point machine. The Subsystem - Point is in: - an End position "Y", - No end position, or - a Unintended position. **Interaction 2.1.1.3.2.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 the timer Con_tmax_Point_Operation. **Interaction 2.1.1.3.2.B:** **3.** - A failure occurred during the movement resulting in a Failed Movement. **4.** The Subsystem - Point 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:** ---  Subsystem - Electronic Interlocking Point machine :Subsystem Point Cd_Move_Point Moving {< Con_tmax_Point_Operation} Movement_Failed Stop_Moving Msg_Movement_Failed	Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE.	Basic non-4-wire single P

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.1474	Info	P SD 2.1.1.3.3 **P UC2.1.1.3: Irregularities**  Subsystem - Electronic Interlocking Point machine :Subsystem Point Information_No_End_Position Msg_Point_Position Alternative Scenario: Handle and report No end position with single point machine [P SD 2.1.1.3.3] **Precondition:** The Subsystem - Point is in the state OPERATIONAL. The Subsystem - Point is configured with a non-4-wire interface to the Point machine. The Subsystem - Point is in: - an End position "Y", or - a Unintended position **Interaction 2.1.1.3.3.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.		Basic non-4-wire single P
Eu.P.1273	Info	P SD 2.1.1.3.4 **P UC2.1.1.3: Irregularities**  Subsystem - Electronic Interlocking Point machine :Subsystem Point Information_Unintended_Position Msg_Point_Position Alternative Scenario: Handle and report Unintended position with single point machine [P SD 2.1.1.3.4] **Precondition:** The Subsystem - Point is in the state OPERATIONAL. The Subsystem - Point is configured with a non-4-wire interface to the Point machine. The Subsystem - Point is in: - an End position "Y", or - No end position. **Interaction 2.1.1.3.4.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.		Basic non-4-wire single P
Eu.P.3207	Info	P SD 2.1.1.3.5 **P UC2.1.1.3: Irregularities**  Subsystem - Electronic Interlocking Point machine :Subsystem Point Information_End_Position_Arrived Msg_Point_Position Alternative Scenario: Handle and report End Position with single point machine [P SD 2.1.1.3.5] **Precondition:** The Subsystem - Point is in the state OPERATIONAL. The Subsystem - Point is configured with a non-4-wire interface to the Point machine. The Subsystem - Point is in: - No end position, or - a Unintended position. **Interaction 2.1.1.3.5.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. **Postcondition:** The Subsystem - Point is in an End position "Y".		Basic non-4-wire single P

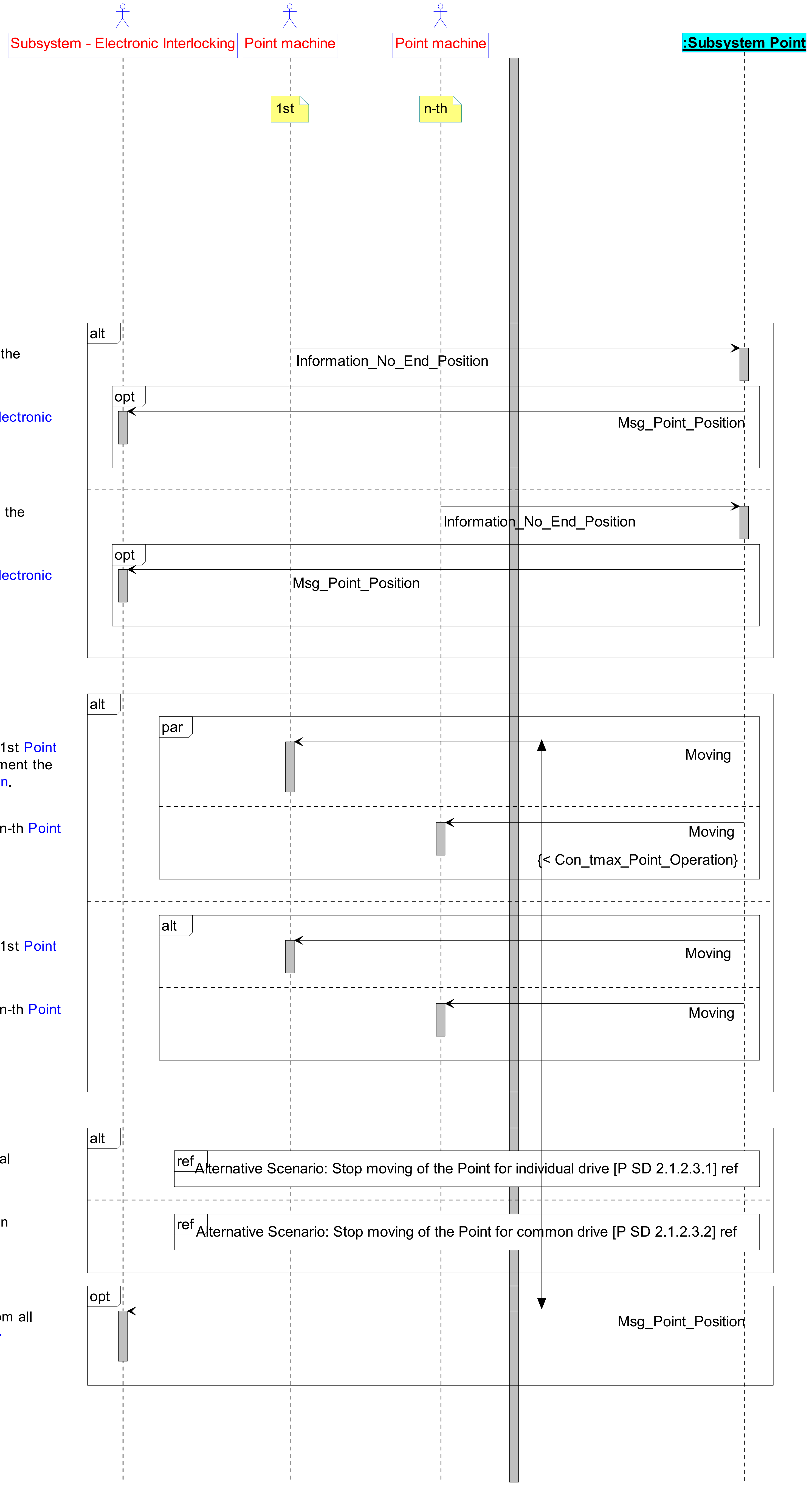
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.4761	Info	P SD 2.1.1.3.6 **P_UC2.1.1.3: Irregularities**  Subsystem - Electronic Interlocking Point machine :Subsystem Point Information_End_Position_Arrived Msg_Point_Position Alternative Scenario: Handle and report End Position out of the other End Position with single point machine [P SD 2.1.1.3.6] **Precondition:** The Subsystem - Point is in the state OPERATIONAL. The Subsystem - Point is configured with a non-4-wire interface to the Point machine. The Subsystem - Point is in an End position "Y". **Interaction 2.1.1.3.6.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".		Basic non-4-wire single P
Eu.P.5373	Info	P SD 2.1.1.3.7 **P_UC2.1.1.3: Irregularities**  Subsystem - Electronic Interlocking Point machine :Subsystem Point alt Information_Ability_To_Move_Point Information_Ability_To_Move_Point_Available_FALSE opt Msg_Ability_To_Move_Point opt Stop_Moving Alternative Scenario: Handle and report loss of ability to move point with single point machine [P SD 2.1.1.3.7] **Precondition**: The Subsystem - Point is in the state INITIALISING or OPERATIONAL. The Subsystem - Point is configured with a non-4-wire interface to the Point machine. The Subsystem - Point is configured to observe the Ability to move point. The Subsystem - Point is in: - Able to move point **Interaction 2.1.1.3.7.A**: **alt** [The Point machine is Unable to move point] **1.a1** - The Subsystem - Point detects from the switch states of the Interface P3 to the Point machine that the Point has no Ability to move point. **else alt** [The internal trigger indicates Unable to move point] **1.b1** - The Subsystem - Point internal trigger indicates that the Subsystem - Point is Unable to move point **end alt** **opt** [The Subsystem - Point is in the state OPERATIONAL] **2.a1** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that it is Unable to move point. **end opt** **Interaction 2.1.1.3.7.B**: **opt** [The point machine is in Moving] **3.a1** The Subsystem - Point sends the Command to the Point machine to Stop moving the Point machine. **end opt** **Postcondition:** The Subsystem - Point is in: - Unable to move point **Interaction 2.1.1.3.7.B**:	Only applicable if the package [Option Able to move] is used in combination with [Basic non-4-wire multiple P].	Option Able to move

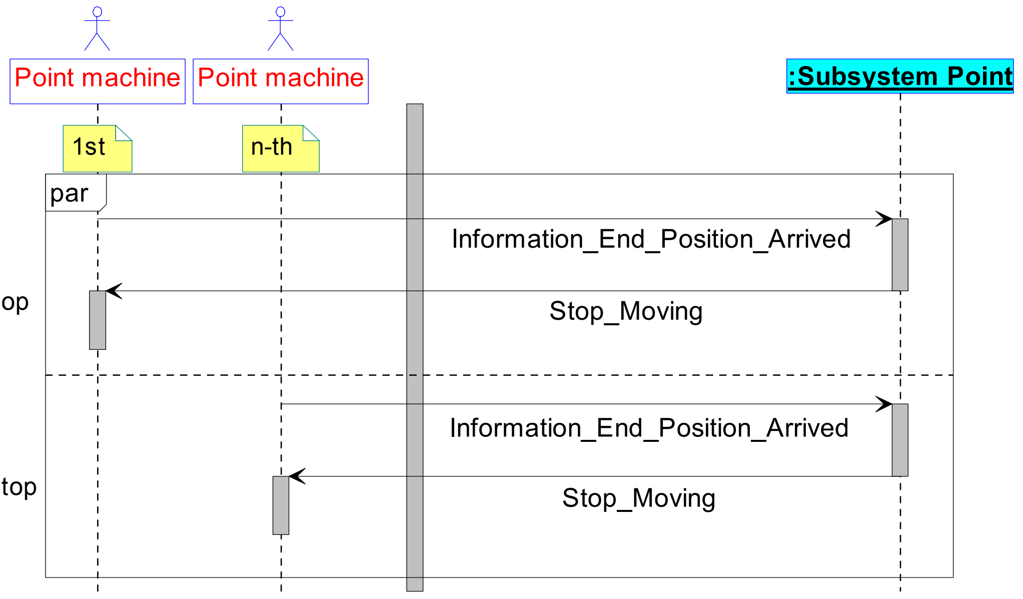
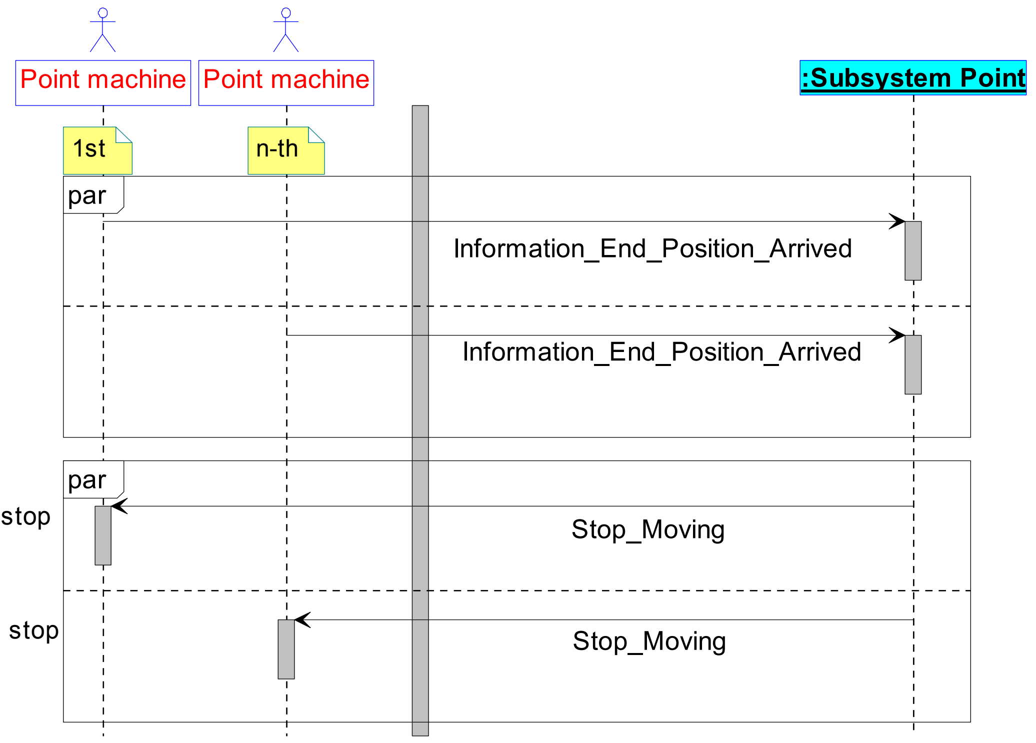
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.5797	Info	P SD 2.1.1.3.8 **P_UC2.1.1.3: Irregularities**  Subsystem - Electronic Interlocking Point machine :Subsystem Point par Information_Ability_To_Move_Point Information_Ability_To_Move_Point_Available_TRUE opt Msg_Ability_To_Move_Point Alternative Scenario: Handle and report restoring of Ability to move point with single point machine [P SD 2.1.1.3.8] **Precondition**: The Subsystem - Point is in the state INITIALISING or OPERATIONAL. The Subsystem - Point is configured with a non-4-wire interface to the Point machine. The Subsystem - Point is configured to observe the Ability to move point. The Subsystem - Point is in: - Unable to move point **Interaction 2.1.1.3.8.A**: **par** **1.a1** - The Subsystem - Point detects from the switch states of the Interface P3 to the Point machine that the Point machine has regained Ability to move point. **also par** **1.b1** - The Subsystem - Point internal trigger indicates that the Subsystem - Point is Able to move point. **end par** **opt** [The Subsystem - Point is in state OPERATIONAL] **2.a1** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that it is Able to move point. **end opt** **Postcondition:** The Subsystem - Point is in: - Able to move point		Basic non-4-wire single P Option Able to move

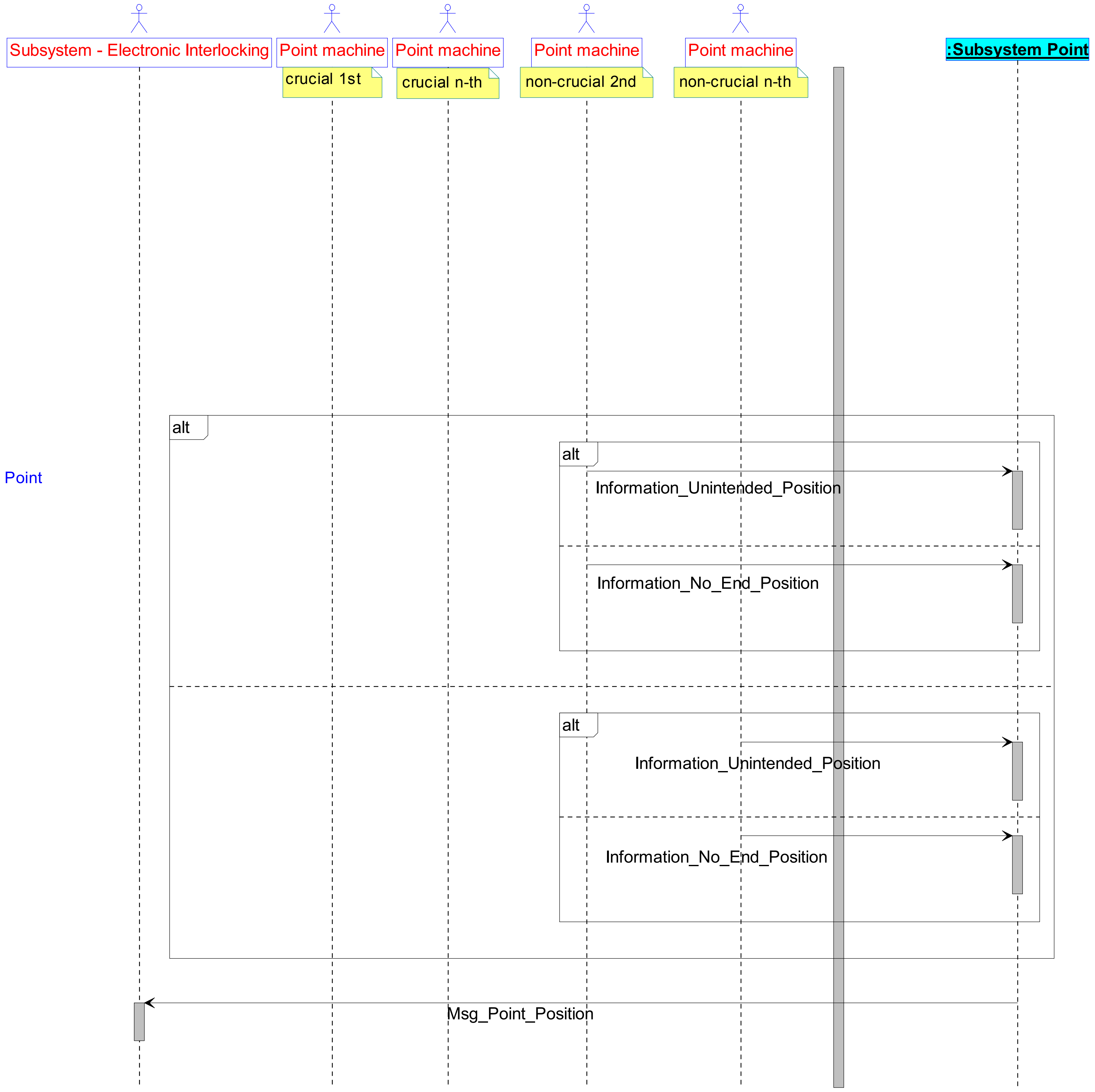
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.6716	Info	P SD 2.1.1.3.9 **P_UC2.1.1.3: Irregularities** Alternative Scenario: Handle Point operation timeout during Redrive Non 4W [P SD 2.1.1.3.9] **Precondition:** The Subsystem Point is in the state INITIALISING or OPERATIONAL. The Subsystem - Point is in an End position "X" The last Cd_Move_Point Command received was also for End position "X" The last Cd_Move_Point Command received occurred since exiting the state BOOTING. The Subsystem - Point is configured with a non-4-wire interface to the Point machine. The Subsystem - Point is configured as Redrive point. **Interaction 2.1.1.2.2.A:** **1.** - The Subsystem - Point receives from the Point machine the Information that the Point machine is in No end position. **opt** [The Subsystem Point is in the state OPERATIONAL.] **2.a1** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in No end position. **end opt** **3.** The Subsystem - Point sends the Command to the Point machine to move the Point back to an End position "X". At this moment the Subsystem - Point starts the timer Con_tmax_Point_Operation. **Interaction 2.1.1.2.2.B:** **4.** - The timer Con_tmax_Point_Operation expires resulting in a Failed Movement. **5.** The Subsystem - Point sends the Command to the Point machine to stop moving the Point machine. **opt** [The Subsystem Point is in the state OPERATIONAL.] **6.a1** The Subsystem - Point reports to the Subsystem - Electronic Interlocking to indicate that a Failed Movement has occurred. **end opt** **Postcondition:** The Subsystem - Point is in No end position.  Subsystem - Electronic Interlocking Point machine :Subsystem Point Information_No_End_Position opt Msg_Point_Position Moving after {Con_tmax_Point_Operation} Stop_Moving opt Msg_Movement_Failed	Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE. Only applicable if the package [Option Redrive] is used in combination with [Basic non-4-wire single P].	Option Redrive
Eu.P.6731	Info	P_UC2.1.2: Multiple point machines	The Subsystem-UseCase "P_UC2.1.2: Multiple point machines" defines the behaviour of the Subsystem Point which works with multiple point machines via non-4-wire interface. The behaviour will be defined in the following UseCases: P_UC2.1.2.1: Commanding and reversing P_UC2.1.2.2: Redrive P_UC2.1.2.4: Irregularities	Basic non-4-wire multiple P
Eu.P.6721	Info	P_UC2.1.2.1: Commanding and reversing	The Subsystem-UseCase "P_UC2.1.2.1: Commanding and reversing" defines the behaviour of commanding and reversing a multiple point machine via non-4-wire interface.	Basic non-4-wire multiple P

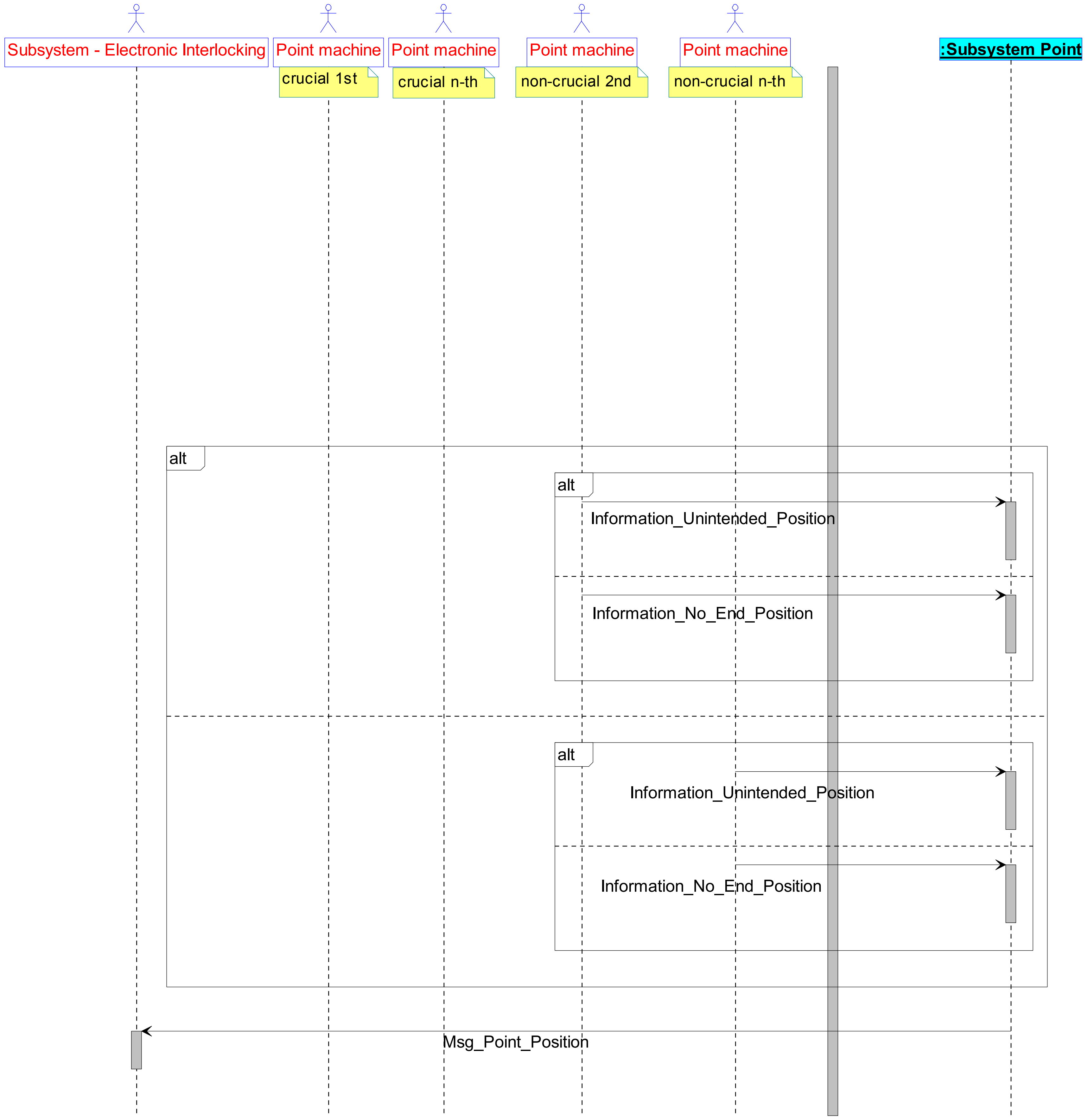
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.6722	Info	P SD 2.1.2.1.1 **P_UC2.1.2.1: Commanding and reversing** Main Success Scenario: Moving of the Point with multiple Point machines Non 4W [P SD 2.1.2.1.1] **Precondition:** The Subsystem - Point is in the state OPERATIONAL. The Subsystem - Point is configured with a non-4-wire interface to the Point machine. The Subsystem - Point is in: - an End position "Y", - No end position, or - a Unintended position. **Interaction 2.1.2.1.1.A:** **1.** - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position "X". **par** **2.a1** The Subsystem - Point sends the Command to the 1st Point machine to move the Point to an End position "X". **also par** **opt** [The n-th Point machine has drive capability] **2.b1.a1** The Subsystem - Point sends the Command to the n-th Point machine to move the Point to an End position "X". end opt **also par** **2.c1** The Subsystem - Point starts the timer Con_tmax_Point_Operation. **end par** **Interaction 2.1.2.1.1.B:** **alt** [The 1st Point machine was previously in an End position or a Unintended position and the n-th is not in a Unintended position] **3.a1** - The Subsystem - Point receives from the 1st 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. **else alt** [The n-th Point machine was previously in an End position or a Unintended position and the 1st Point machine is not in a Unintended position] **3.b1** - The Subsystem - Point receives from the n-th Point machine the Information that the Point is in No end position. **3.b2** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in No end position. **else alt** [The 1st Point machine was previously in a Unintended position and the n-th Point machine was in a Unintended position] **par** **3.c1.a1** - The Subsystem - Point receives from the 1st Point machine the Information that the Point is in No end position. **also par** **3.c1.b1** - The Subsystem - Point receives from the n-th Point machine the Information that the Point is in No end position. **end par** **3.d1** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in No end position. **end alt** **Interaction 2.1.2.1.1.C:** **alt** [The Subsystem - Point is configured for individual drive.] **4.a1** The Subsystem - Point stops moving of the Point for individual drive. **else alt** [The Subsystem - Point is configured for common drive.] **4.b1** The Subsystem - Point stops moving of the Point for common drive. **end alt** **5.** When Information_End_Position_Arrived has been received from all Point machines, the Subsystem - Point sends a Message to the Subsystem - Electronic Interlocking indicating that the Point is in an End position "X". **Postcondition:** The Subsystem - Point is in an End position "X".  Subsystem - Electronic Interlocking Point machine Point machine :Subsystem Point 1st n-th Cd_Move_Point par Moving opt Moving {< Con_tmax_Point_Operation} alt Information_No_End_Position Msg_Point_Position Information_No_End_Position Msg_Point_Position par Information_No_End_Position Information_No_End_Position Msg_Point_Position alt ref Alternative Scenario: Stop moving of the Point for individual drive [P SD 2.1.2.3.1] ref ref Alternative Scenario: Stop moving of the Point for common drive [P SD 2.1.2.3.2] ref Msg_Point_Position	Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE. The timing related to driving individual (1 to n-th) Point Machines for a single Point is supplier specific and is specified only in a general way. The timing related to individual Point Machines is not specified by EULYNX as part of the application layer, this shall be handled by the physical implementation.	Basic non-4-wire multiple P

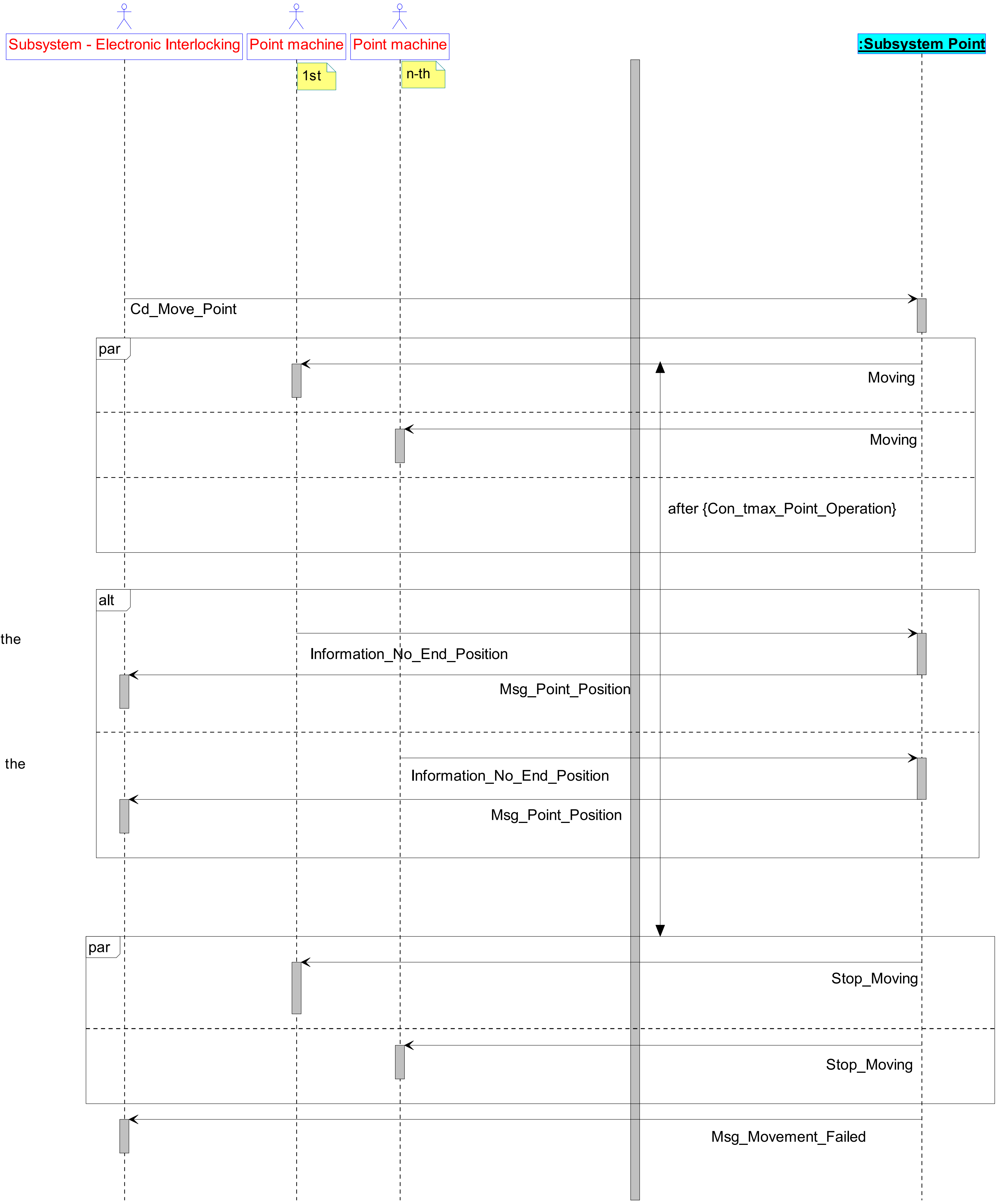
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.6723	Info	P_UC2.1.2.2: Redrive	The Subsystem-UseCase "P_UC2.1.2.2: Redrive" defines the behaviour of redriving a multiple point machine via non-4-wire interface. Only applicable if the package [Option Redrive] is used in combination with [Basic non-4-wire multiple P].	Option Redrive

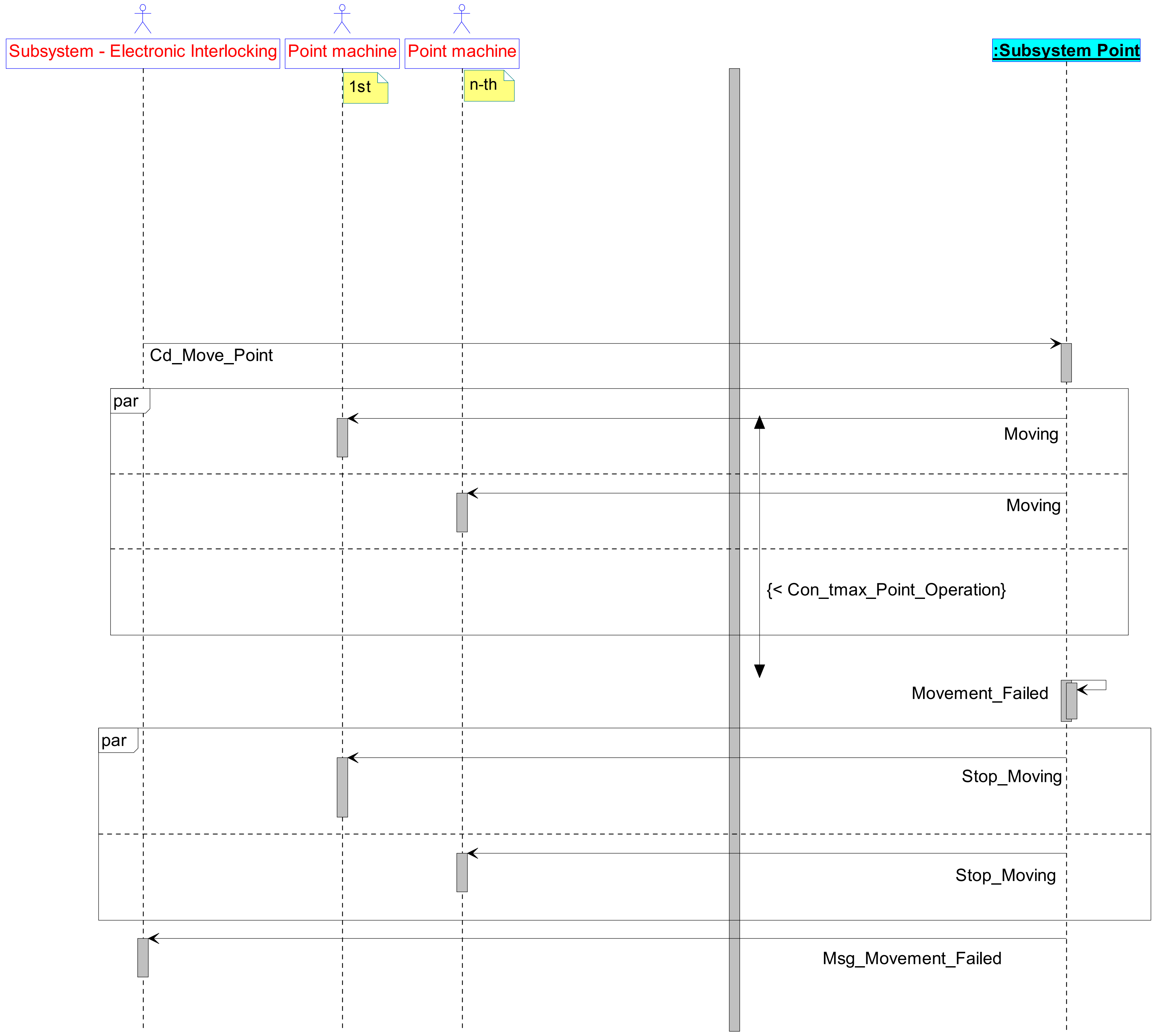
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.7098	Info	P SD 2.1.2.2.1 **P_UC2.1.2.2: Redrive** Main Success Scenario: Redrive Point after lost end position [P SD 2.1.2.2.1] **Precondition:** The Subsystem Point is in the state INITIALISING or OPERATIONAL. The Subsystem - Point is in an End position "X". The last Cd_Move_Point Command received was also for End position "X". The last Cd_Move_Point Command received occurred since exiting the state BOOTING. The Subsystem - Point is configured with a non-4-wire interface to the Point machine. The Subsystem - Point is configured as Redrive point. **Interaction 2.1.2.2.1.A:** **alt** [The 1st Point machine was previously in an End position] **1.a1** - The Subsystem - Point receives from the 1st Point machine the Information that the Point is in No end position. **opt** [The Subsystem Point is in the state OPERATIONAL.] **1.a2.a1** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in No end position. **end opt** **else alt** [The n-th Point machine was previously in an End position] **1.b1** - The Subsystem - Point receives from the n-th Point machine the Information that the Point is in No end position. **opt** [The Subsystem Point is in the state OPERATIONAL.] **1.b2.a1** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in No end position. **end opt** **end alt** **Interaction 2.1.2.2.1.B:** **alt** [The Subsystem - Point is configured for common drive.] **par** **2.a1.a1** - The Subsystem - Point sends the Command to the 1st Point machine to move the Point to an End position "X". At this moment the Subsystem - Point starts the timer Con_tmax_Point_Operation. **also par** **2.a1.b1** - The Subsystem - Point sends the Command to the n-th Point machine to move the Point to an End position "X". **end par** **else alt** [The Subsystem - Point is configured for individual drive.] **alt** [The 1st Point machine is in No end position] **2.b1.a1** - The Subsystem - Point sends the Command to the 1st Point machine to move the Point to an End position "X". **else alt** [The n-th Point machine is in No end position] **2.b1.b1** - The Subsystem - Point sends the Command to the n-th Point machine to move the Point to an End position "X". **end alt** **end alt** **Interaction 2.1.2.2.1.C:** **alt** [The Subsystem Point is configured for individual drive] **3.a1** - The Subsystem - Point stops moving of the Point for individual drive. **else alt** [The Subsystem Point is configured for common drive] **3.b1** - The Subsystem - Point stops moving of the Point for common drive. **end alt** **opt** [The Subsystem Point is in the state OPERATIONAL.] **4.a1** When Information_End_Position_Arrived has been received from all Point machines, the Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in an End position "X". **end opt** **Postcondition:** The Subsystem - Point is in an End position "X".  Subsystem - Electronic Interlocking Point machine Point machine :Subsystem Point 1st n-th alt Information_No_End_Position opt Msg_Point_Position Information_No_End_Position opt Msg_Point_Position alt par Moving Moving {< Con_tmax_Point_Operation} alt Moving Moving alt ref Alternative Scenario: Stop moving of the Point for individual drive [P SD 2.1.2.3.1] ref ref Alternative Scenario: Stop moving of the Point for common drive [P SD 2.1.2.3.2] ref opt Msg_Point_Position	Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE. Only applicable if the package [Option Redrive] is used in combination with [Basic non-4-wire multiple P].	Option Redrive

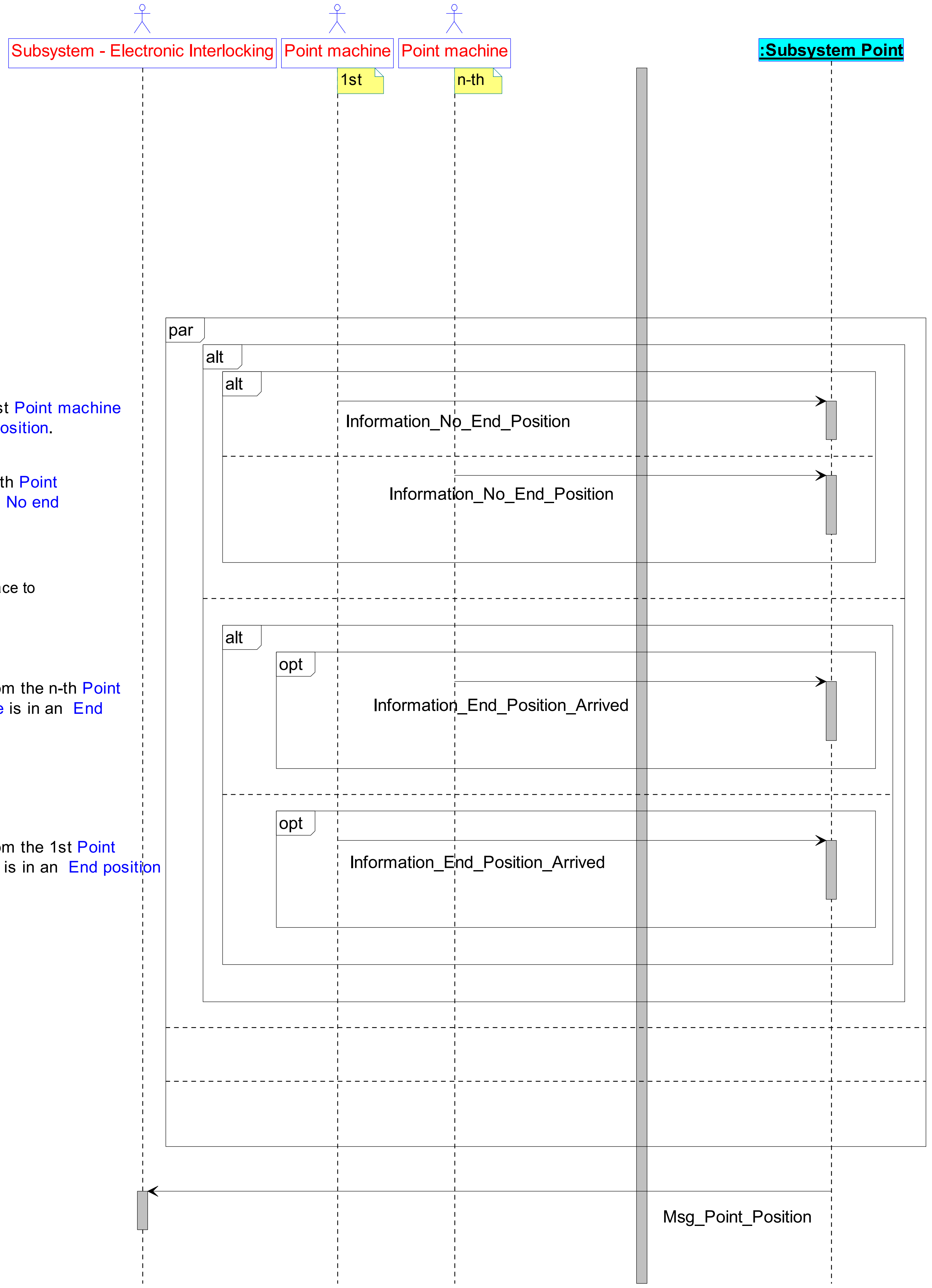
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.6726	Info	P_UC2.1.2.3: Stop point movement		Basic non-4-wire multiple P
Eu.P.6728	Info	P SD 2.1.2.3.1 **P_UC2.1.2.3: Stop point movement** Alternative Scenario: Stop moving of the Point for individual drive [P SD 2.1.2.3.1] **par** **1.a1** - The 1st Point machine sends an Information to the Subsystem - Point indicating that the Point is in an End position "X". **1.a2** The Subsystem - Point sends a Command to the 1st Point machine to stop moving the Point. **also par** **1.b1** - The n-th Point machine sends an Information to the Subsystem - Point indicating that the Point is in an End position "X". **1.b2** The Subsystem - Point sends a Command to the n-th Point machine to stop moving the Point. **end par**  Point machine Point machine :Subsystem Point 1st n-th par Information_End_Position_Arrived Stop_Moving Information_End_Position_Arrived Stop_Moving	Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE. The timing related to driving individual (1 to n-th) Point Machines for a single Point is supplier specific and is specified only in a general way. The timing related to individual Point Machines is not specified by EULYNX as part of the application layer, this shall be handled by the physical implementation.	Basic non-4-wire multiple P
Eu.P.6727	Info	P SD 2.1.2.3.2 **P_UC2.1.2.3: Stop point movement** Alternative Scenario: Stop moving of the Point for common drive [P SD 2.1.2.3.2] **par** **1.a1** - The 1st Point machine sends an Information to the Subsystem - Point indicating that the Point is in an End position "X". **also par** **1.b1** - The n-th Point machine sends an Information to the Subsystem - Point indicating that the Point is in an End position "X". **end par** **par** **2.a1** The Subsystem - Point sends a Command to the 1st Point machine to stop moving the Point. **also par** **2.b1** The Subsystem - Point sends a Command to the n-th Point machine to stop moving the Point. **end par**  Point machine Point machine :Subsystem Point 1st n-th par Information_End_Position_Arrived Information_End_Position_Arrived par Stop_Moving Stop_Moving	Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE. The timing related to driving individual (1 to n-th) Point Machines for a single Point is supplier specific and is specified only in a general way. The timing related to individual Point Machines is not specified by EULYNX as part of the application layer, this shall be handled by the physical implementation.	Option Common Drive
Eu.P.6729	Info	P_UC2.1.2.4: Irregularities	The Subsystem-UseCase "P_UC2.1.2.4: Irregularities" defines the behaviour of the Subsystem Point which works with a multiple point machine via non-4-wire interface, when an irregularity occurs.	Basic non-4-wire multiple P

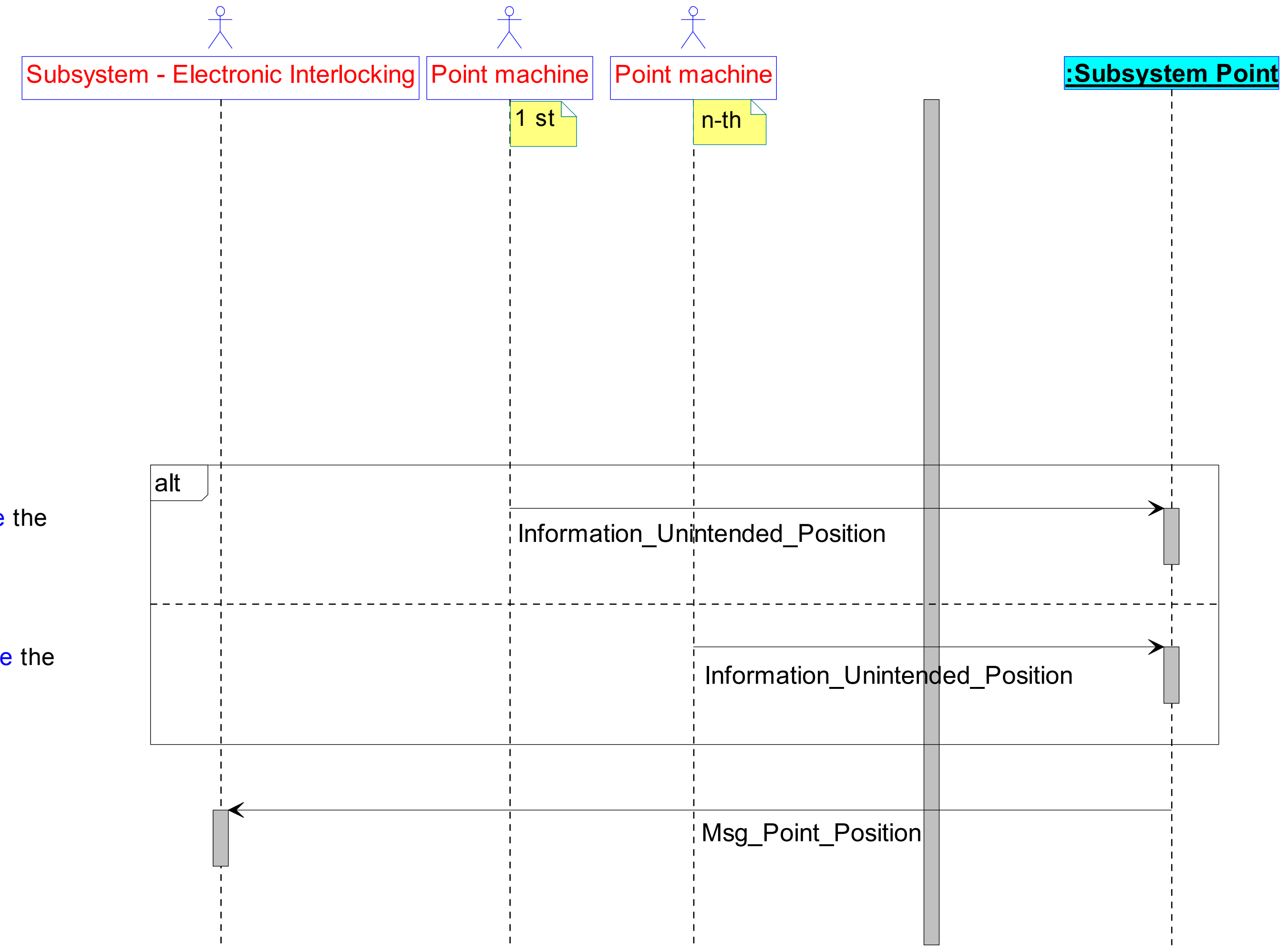
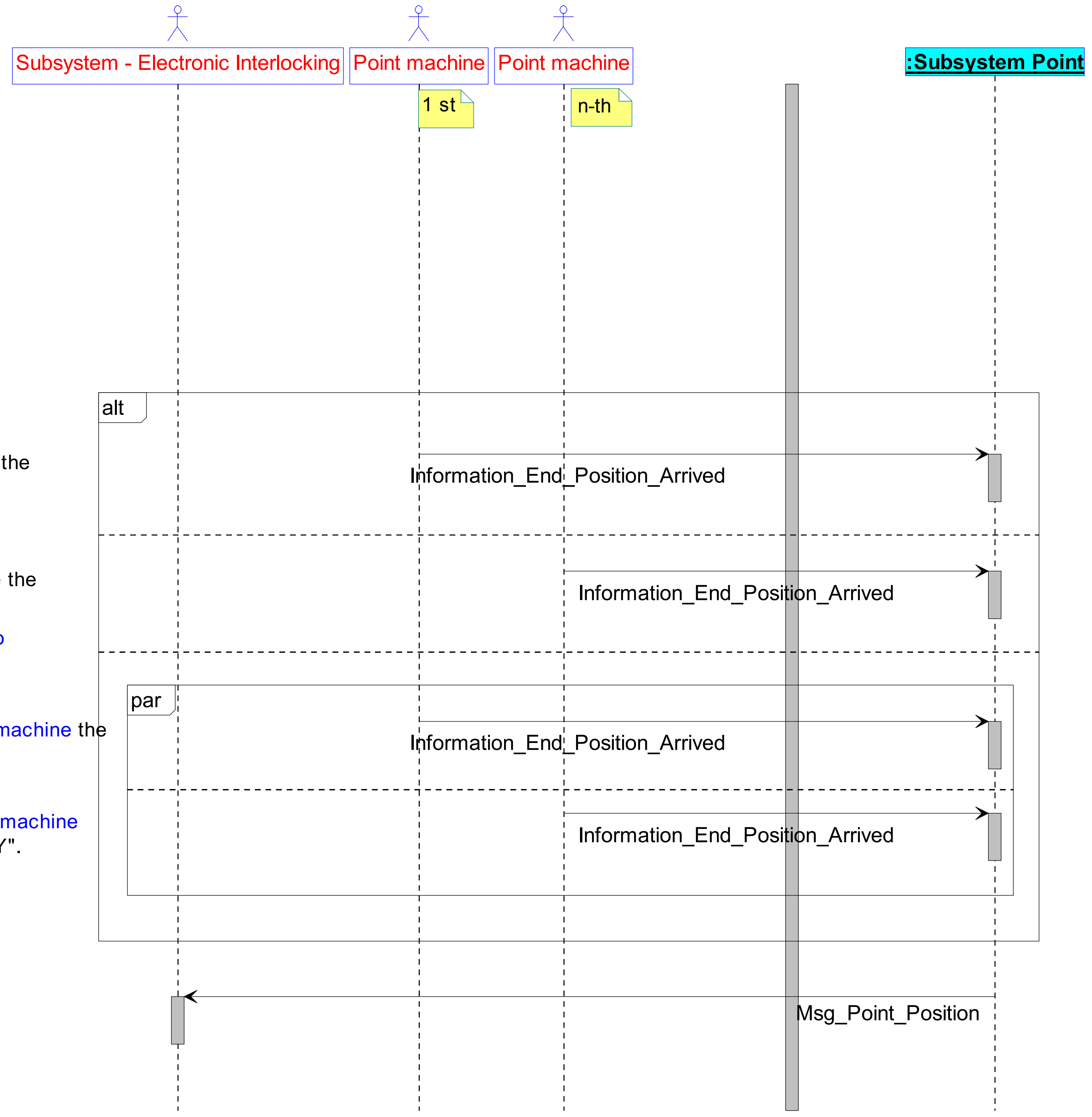
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.5801	Info	P SD 2.1.2.4.1 **P_UC2.1.2.4: Irregularities** Alternative Scenario: Handle and report Degraded Point Position with n-th Point machines [P SD 2.1.2.4.1] **Precondition:** The Subsystem - Point is in the state OPERATIONAL. The Subsystem - Point is configured with a non-4-wire interface to the Point machine. The Subsystem - Point is in a Non degraded position. The 1st crucial Point machine is in an End position "X". The Subsystem - Point is configured with only one crucial Point machine OR the Subsystem - Point is configured with more than one crucial Point machine AND the n-th crucial Point machine is in an End position "X". The non-crucial 2nd Point machine is NOT in End position "Y". If the Subsystem - Point is configured with a non-crucial n-th Point machine, it is NOT in End position "Y". **Interaction 2.1.2.4.1.A** **alt** [The non-crucial 2nd Point machine is not in an End position] **alt** [The non-crucial 2nd Point machine is in a Unintended position.] **1.a1.a1** - The Subsystem - Point receives from the non-crucial 2nd Point machine the Information that the non-crucial 2nd Point machine is in a Unintended position. **else alt** [The non-crucial 2nd Point machine is in No end position.] **1.a1.b1** - The Subsystem - Point receives from the non-crucial 2nd Point machine the Information that the non-crucial 2nd Point machine is in No end position. **end alt** **else alt** [The Subsystem - Point is configured with more than one non-crucial Point machine] **alt** [The non-crucial n-th Point machine is in a Unintended position.] **1.b1.a1** - The Subsystem - Point receives from the non-crucial n-th Point machine the Information that the non-crucial n-th Point machine is in a Unintended position. **else alt** [The non-crucial n-th Point machine is in No end position.] **1.b1.b1** - The Subsystem - Point receives from the non-crucial n-th Point machine the Information to the that the non-crucial n-th Point machine is in No end position. **end alt** **end alt** **Interaction 2.1.2.4.1.B** **2.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in a Degraded point position "X" **Postcondition**: The Subsystem - Point is in a Degraded point position "X"  Subsystem - Electronic Interlocking Point machine crucial 1st Point machine crucial n-th Point machine non-crucial 2nd Point machine non-crucial n-th :Subsystem Point alt alt Information_Unintended_Position Information_No_End_Position alt Information_Unintended_Position Information_No_End_Position Msg_Point_Position		Basic non-4-wire multiple P

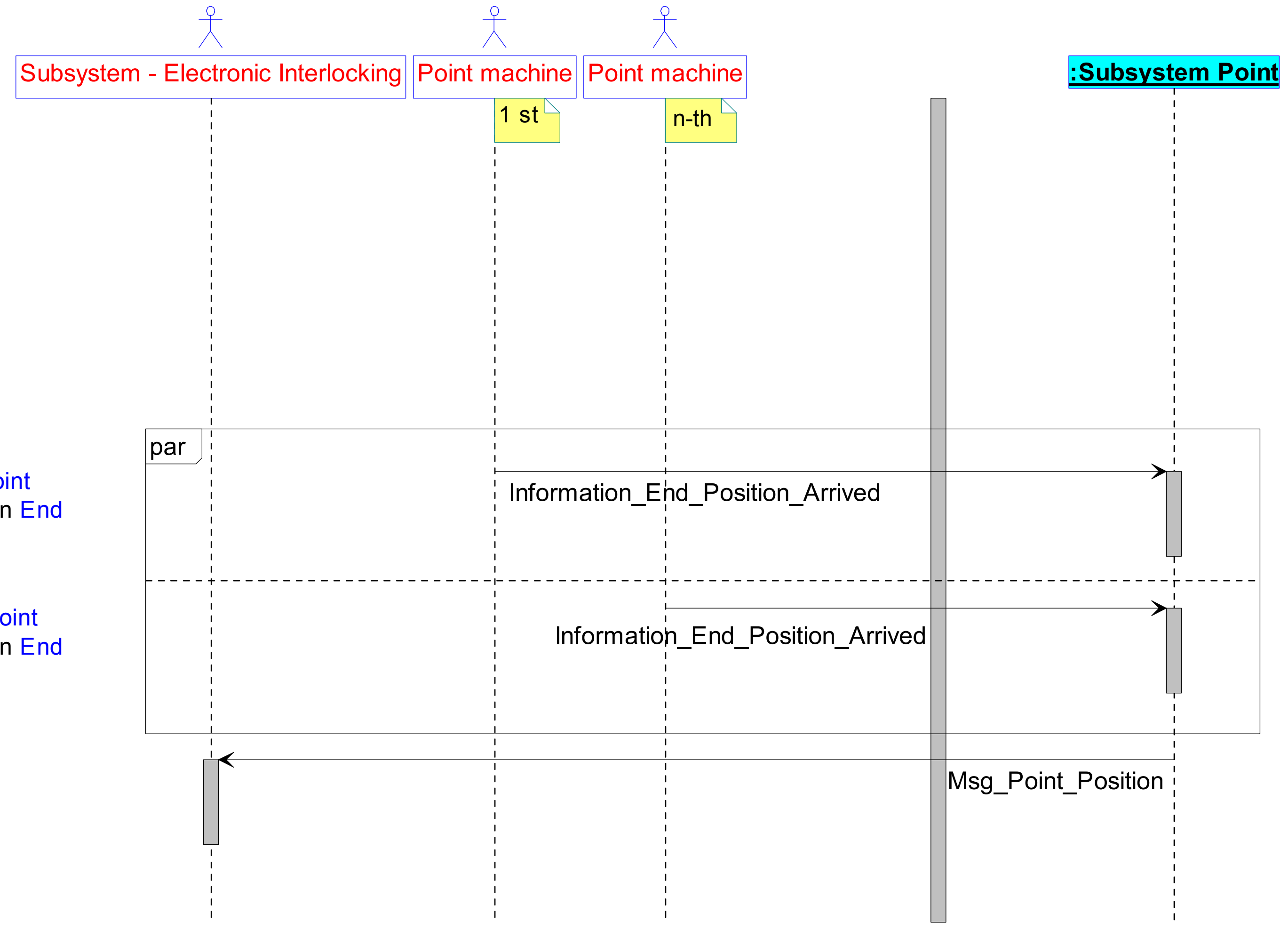
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.5806	Info	P SD 2.1.2.4.2 **P_UC2.1.2.4: Irregularities** Alternative Scenario: Handle and report non-degraded Point Position with n-th Point machines [P SD 2.1.2.4.2] **Precondition:** The Subsystem - Point is in the state OPERATIONAL. The Subsystem - Point is configured with a non-4-wire interface to the Point machine. The Subsystem - Point is in a Degraded point position "X" **Interaction 2.1.2.4.2.A** The following conditions are no longer fulfilled for either "X" or "Y". The 1st crucial Point machine is in an is End position "X". The non-crucial 2nd Point machine is NOT in End position "Y". If the Subsystem - Point is configured with a non-crucial n-th Point machine, it is NOT in End position "Y". The Subsystem - Point is configured with only one crucial Point machine OR the Subsystem - Point is configured with more than one crucial Point machine AND the n-th crucial Point machine is in an is End position "X". **alt** [The non-crucial 2nd Point machine is not in an End position] **alt** [The non-crucial 2nd Point machine is in a Unintended position.] **1.a1.a1** - The Subsystem - Point receives from the non-crucial 2nd Point machine the Information to the that the non-crucial 2nd Point machine is in a Unintended position. **else alt** [The non-crucial 2nd Point machine is in No end position.] **1.a1.b1** - The Subsystem - Point receives from the non-crucial 2nd Point machine sends an Information to the that the non-crucial 2nd Point machine is in No end position. **end alt** **else alt** [The Subsystem - Point is configured with more than one non-crucial Point machine] **alt** [The non-crucial n-th Point machine is in a Unintended position.] **1.b1.a1** - The Subsystem - Point receives from the non-crucial n-th Point machine sends an Information that the non-crucial n-th Point machine is in a Unintended position. **else alt** [The non-crucial n-th Point machine is in No end position.] **1.b1.b1** - The Subsystem - Point receives from the non-crucial n-th Point machine the Information to the that the non-crucial n-th Point machine is in No end position. **end alt** **end alt** **Interaction 2.1.2.4.2.B** **2.** - The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is not in a Degraded point position "X" **Postcondition**: The Subsystem - Point is in a Non degraded position.  Subsystem - Electronic Interlocking Point machine crucial 1st Point machine crucial n-th Point machine non-crucial 2nd Point machine non-crucial n-th :Subsystem Point alt alt Information_Unintended_Position Information_No_End_Position alt Information_Unintended_Position Information_No_End_Position Msg_Point_Position		Basic non-4-wire multiple P

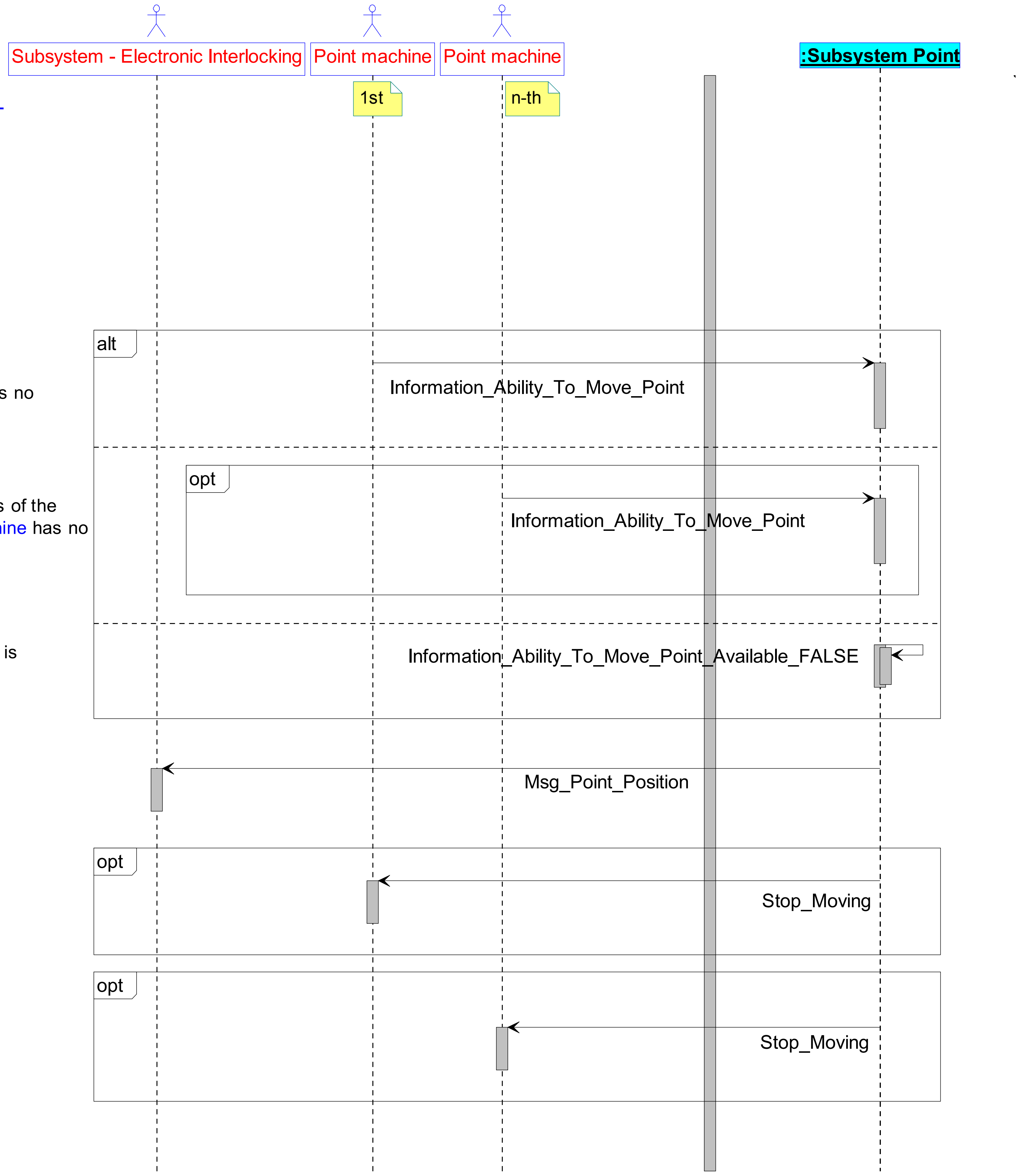
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.1475	Info	P SD 2.1.2.4.3 **P UC2.1.2.4: Irregularities**  Subsystem - Electronic Interlocking Point machine Point machine :Subsystem Point 1st n-th Cd_Move_Point par Moving Moving after {Con_tmax_Point_Operation} alt Information_No_End_Position Msg_Point_Position Information_No_End_Position Msg_Point_Position par Stop_Moving Stop_Moving Msg_Movement_Failed Alternative Scenario: Handle and report Point operation timeout with n-th Point machines with position change [P SD 2.1.2.4.3] **Precondition:** The Subsystem - Point is in the state OPERATIONAL. The Subsystem - Point is configured with a non-4-wire interface to the Point machine. The Subsystem - Point is in: - an End position "Y", - No end position, or - a Unintended position. **Interaction 2.1.2.4.3.A:** **1.** - The Subsystem - Electronic Interlocking sends a Command to the Subsystem - Point to move the Point to an End position "X". **par** **2.a1** The Subsystem - Point sends the Command to the 1st Point machine to move the 1st Point machine to an End position "X". **also par** **2.b1** The Subsystem - Point sends the Command to the n-th Point machine to move the n-th Point machine to an End position "X". **also par** **2.c1** The Subsystem - Point starts the Timer for moving Con_tmax_Point_Operation. **end par** **Interaction 2.1.2.4.3.B:** **alt** [The 1st Point machine was previously in an End position or a Unintended position] **3.a1** - The Subsystem - Point receives from the 1st Point machine the Information that the 1st Point machine is in No end position. **3.a2** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in No end position. **else alt** [The n-th Point machine was previously in an End position or a Unintended position] **3.b1** - The Subsystem - Point receives from the n-th Point machine the Information that the n-th Point machine is in No end position. **3.b2** The Subsystem - Point reports to the Subsystem - Electronic Interlocking indicating that the Point is in No end position. **end alt** **Interaction 2.1.2.4.3.C:** **4.** - The timer Con_tmax_Point_Operation expires resulting in a Failed Movement. **par** **5.a1** The Subsystem - Point sends the Command to the 1st Point machine to stop moving the 1st Point machine. The timer Con_tmax_Point_Operation is reset. **also par** **5.b1** The Subsystem - Point sends the Command to the n-th Point machine to stop moving the n-th Point machine. **end par** **6.** 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. The timing related to driving individual (1 to n-th) Point Machines for a single Point is supplier specific and is specified only in a general way. The timing related to individual Point Machines is not specified by EULYNX as part of the application layer, this shall be handled by the physical implementation.	Basic non-4-wire multiple P

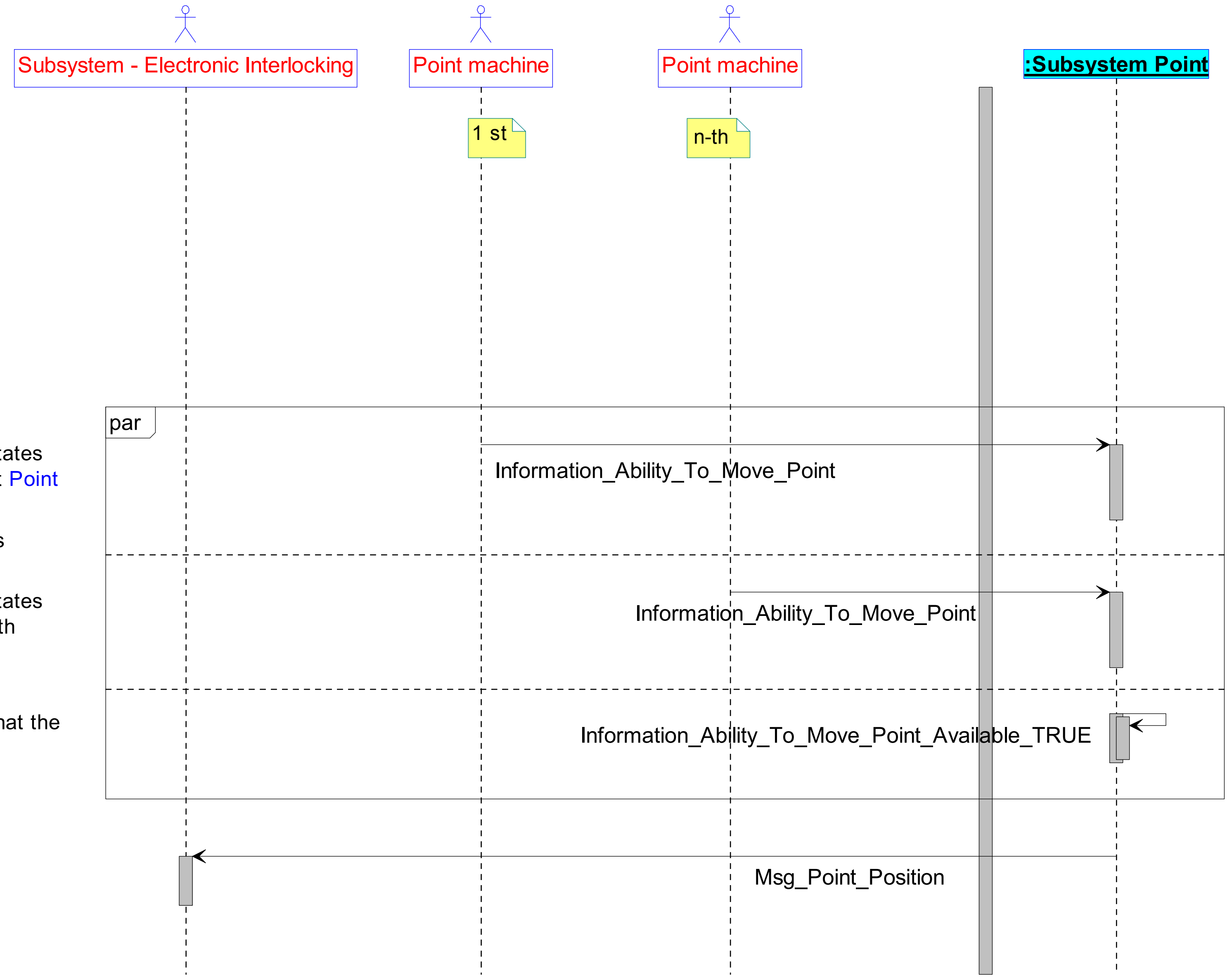
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.6730	Info	P SD 2.1.2.4.4 **P_UC2.1.2.4: Irregularities**  Subsystem - Electronic Interlocking Point machine Point machine :Subsystem Point 1st n-th Cd_Move_Point par Moving Moving {< Con_tmax_Point_Operation} Movement_Failed par Stop_Moving Stop_Moving Msg_Movement_Failed Alternative Scenario: Handle and report failed movement with n-th Point machines without position change [P SD 2.1.2.4.4] **Precondition:** The Subsystem - Point is in the state OPERATIONAL. The Subsystem - Point is configured with a non-4-wire interface to the Point machine. The Subsystem - Point is in: - an End position "Y", - No end position, or - a Unintended position. **Interaction 2.1.2.4.4.A:** **1.** - The Subsystem - Electronic Interlocking sends a Command to the Subsystem - Point to move the Point to an End position "X". **par** **2.a1** The Subsystem - Point sends the Command to the 1st Point machine to move the 1st Point machine to an End position "X". **also par** **2.b1** The Subsystem - Point sends the Command to the n-th Point machine to move the n-th Point machine to an End position "X". **also par** **2.c1** The Subsystem - Point starts the Timer for moving Con_tmax_Point_Operation. **end par** **Interaction 2.1.2.4.4.B:** **3.** - A failure occurred during the movement resulting in a Failed Movement. **par** **4.a1** The Subsystem - Point sends the Command to the 1st Point machine to stop moving the 1st Point machine. The timer Con_tmax_Point_Operation is reset. **also par** **4.b1** The Subsystem - Point sends the Command to the n-th Point machine to stop moving the n-th Point machine. **end par** **5.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that a Failed Movement has occurred. **Postcondition:** ---	Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE. The timing related to driving individual (1 to n-th) Point Machines for a single Point is supplier specific and is specified only in a general way. The timing related to individual Point Machines is not specified by EULYNX as part of the application layer, this shall be handled by the physical implementation.	Basic non-4-wire multiple P

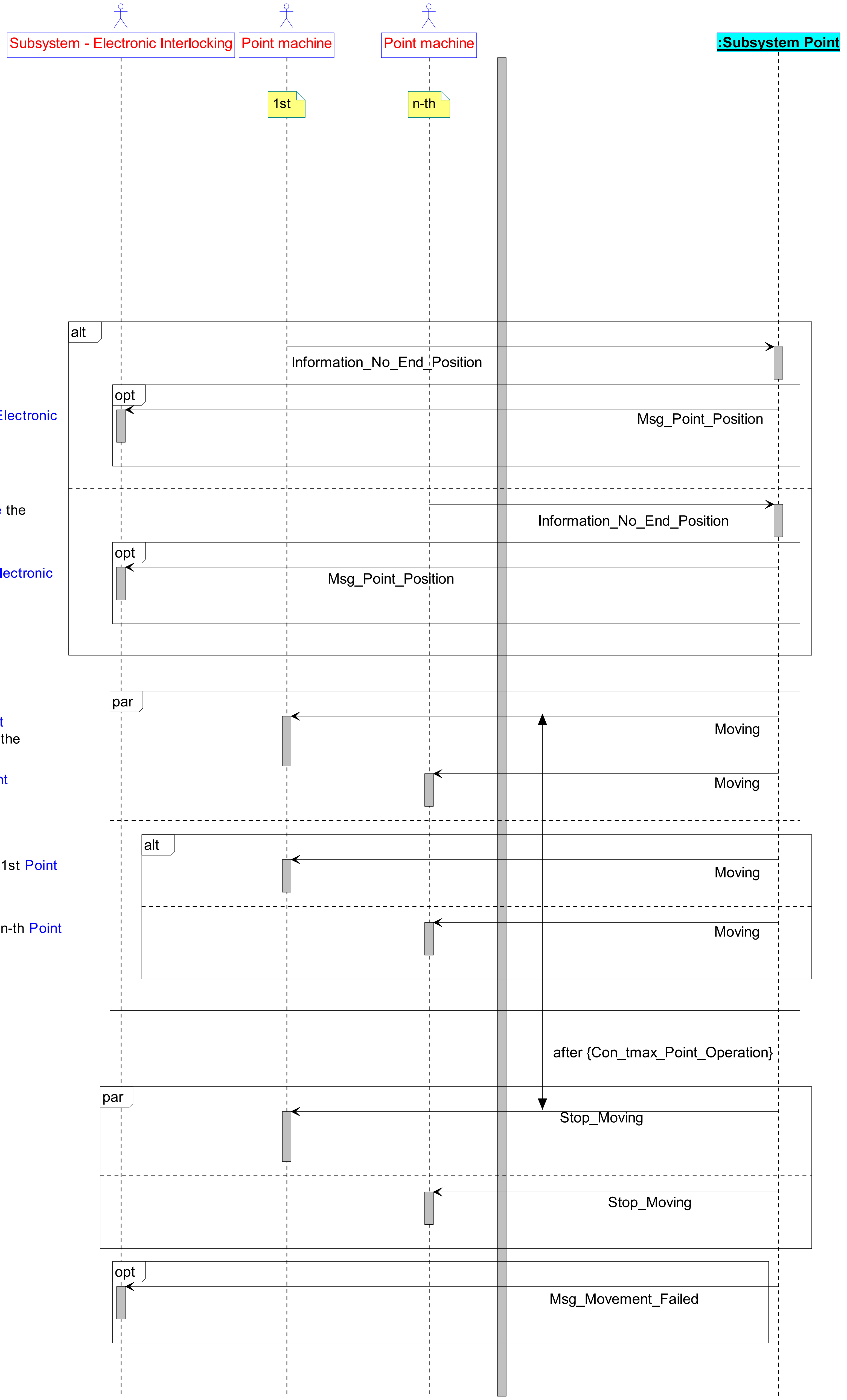
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.5805	Info	P SD 2.1.2.4.5 **P_UC2.1.2.4: Irregularities** Alternative Scenario: Handle and report No end position with n-th Point machines [P SD 2.1.2.4.5] **Precondition:** The Subsystem - Point is in the state OPERATIONAL. The Subsystem - Point is configured with a non-4-wire interface to the Point machine. The Subsystem - Point is in: - an End position "Y", or - a Unintended position **Interaction 2.1.2.4.5.A:** **par** **alt** [Any Point machine is in an End position] **alt** [The 1st Point machine is in No end position] **1.a1.a1.a1** - The Subsystem - Point receives from the 1st Point machine the Information that the 1st Point machine is in No end position. **else alt** [The n-th Point machine is in No end position] **1.a1.a1.b1** - The Subsystem - Point receives from the n-th Point machine the Information that the n-th Point machine is in No end position. **end alt** **else alt** [The Subsystem - Point is not configured with a 4-wire interface to the Point machine] **alt** [The 1st Point machine is in an End position "Y".] **opt** [The n-th Point machine is in an End position "X".] **1.a1.b1.a1.a1** - The Subsystem - Point receives from the n-th Point machine the Information that the n-th Point machine is in an End position "X". **end opt** **else alt** [The n-th Point machine is in an End position "Y"] **opt** [The 1st Point machine is in an End position "X".] **1.a1.b1.b1.a1** - The Subsystem - Point receives from the 1st Point machine the Information that the 1st Point machine is in an End position "X". **end opt** **end alt** **end alt** **also par** **1.b1** - The 1st Point machine is NOT in a Unintended position. **also par** **1.c1** - The n-th Point machine is NOT in a Unintended position. **end par** **Interaction 2.1.2.4.5.B:** **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.  Subsystem - Electronic Interlocking Point machine 1st Point machine n-th :Subsystem Point par alt alt Information_No_End_Position Information_No_End_Position alt opt Information_End_Position_Arrived opt Information_End_Position_Arrived Msg_Point_Position		Basic non-4-wire multiple P

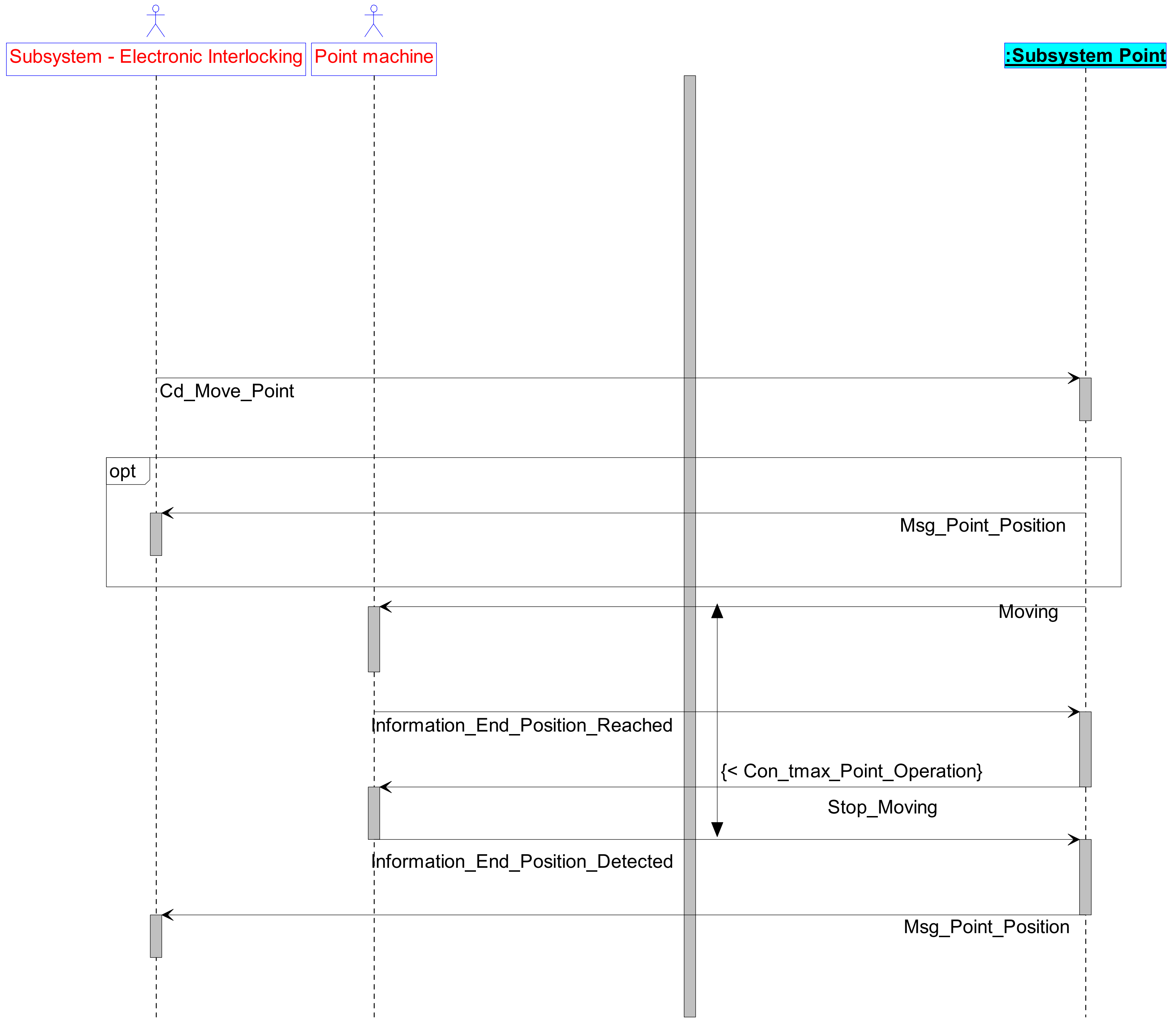
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.5808	Info	P SD 2.1.2.4.6 **P UC2.1.2.4: Irregularities**  Subsystem - Electronic Interlocking Point machine 1 st Point machine n-th :Subsystem Point alt Information_Unintended_Position Information_Unintended_Position Msg_Point_Position Alternative Scenario: Handle and report Unintended position with n-th point machine [P SD 2.1.2.4.6] **Precondition:** The Subsystem - Point is in the state OPERATIONAL. The Subsystem - Point is configured with a non-4-wire interface to the Point machine. The Subsystem - Point is in: - an End position "Y", or - No end position. **Interaction 2.1.2.4.6.A:** **alt** [The 1st Point machine is in a End position "Y" or No end position.] **1.a1** - The Subsystem - Point receives from the 1st Point machine the Information that the Point machine is in a Unintended position. **else alt** [The n-th Point machine is in a End position "Y" or No end position.] **1.b1** - The Subsystem - Point receives from the n-th Point machine the Information that the Point machine is in a Unintended position. **end alt** **Interaction 2.1.2.4.6.B:** **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.		Basic non-4-wire multiple P
Eu.P.5803	Info	P SD 2.1.2.4.7 **P UC2.1.2.4: Irregularities**  Subsystem - Electronic Interlocking Point machine 1 st Point machine n-th :Subsystem Point alt Information_End_Position_Arrived Information_End_Position_Arrived par Information_End_Position_Arrived Information_End_Position_Arrived Msg_Point_Position Alternative Scenario: Handle and report End Position with n-th point machine [P SD 2.1.2.4.7] **Precondition:** The Subsystem - Point is in the state OPERATIONAL. The Subsystem - Point is configured with a non-4-wire interface to the Point machine The Subsystem - Point is in: - a No end position, or - an Unintended position. **Interaction 2.1.2.4.7.A:** **alt** [The 1st Point machine is an No end position or a Unintended position and the n-th Point machine is in an End position "Y".] **1.a1** - The Subsystem - Point receives from the 1st Point machine the Information that the Point machine is in an End position "Y". **else alt** [The n-th Point machine is an No end position or a Unintended position and the 1st Point machine is in an End position "Y".] **1.b1** - The Subsystem - Point receives from the n-th Point machine the Information that the Point machine is in an End position "Y". **else alt** [The n-th Point machine and the 1st Point machine are in an No end position or a Unintended position.] par **1.c1.a1** - The Subsystem - Point receives from the 1st Point machine the Information that the Point machine is in an End position "Y". also par **1.c1.b1** - The Subsystem - Point receives from the n-th Point machine the Information that the Point machine is in an End position "Y". end par **end alt** **Interaction 2.1.2.4.7.B:** **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".		Basic non-4-wire multiple P

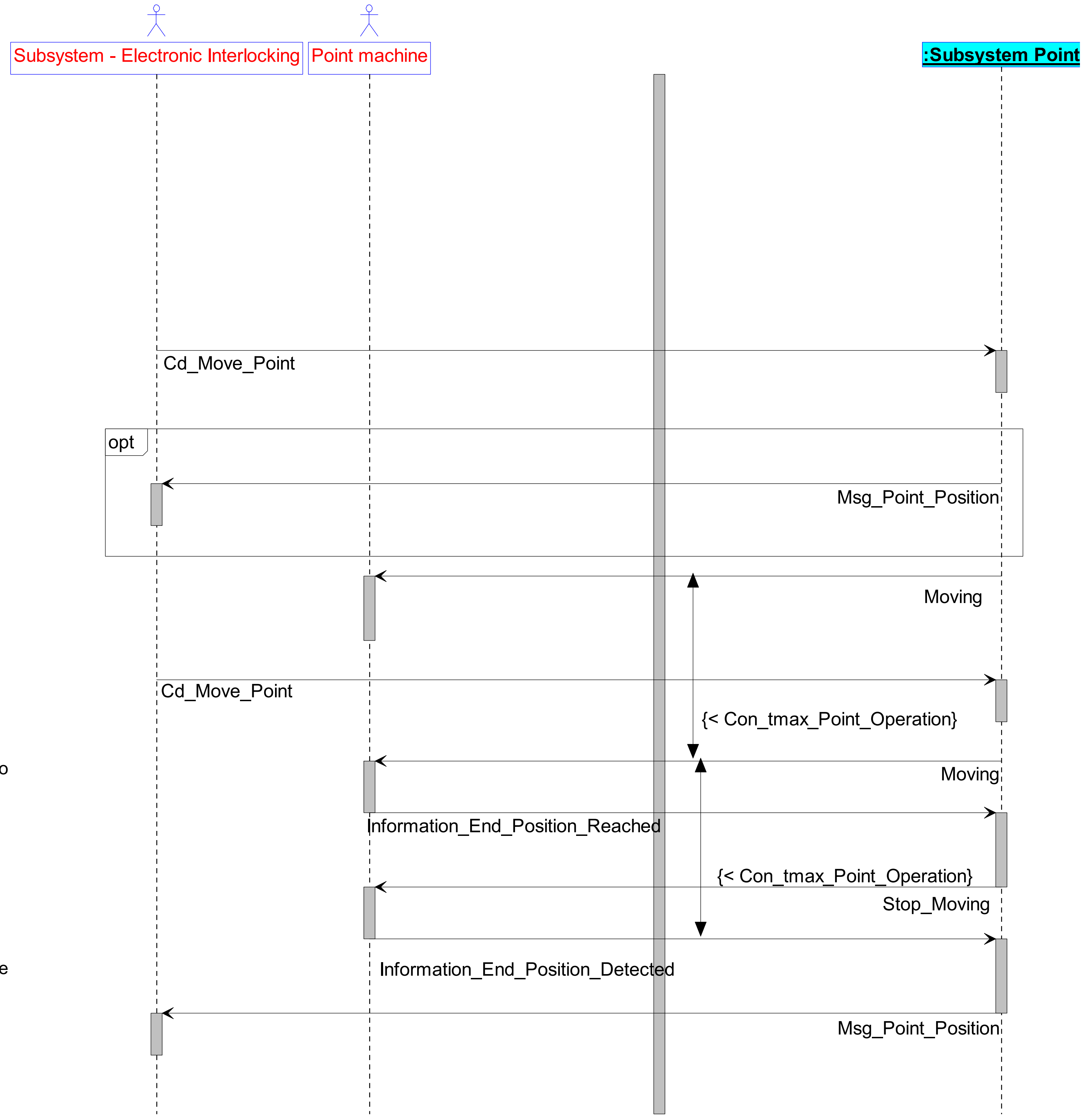
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.5802	Info	P SD 2.1.2.4.8  Subsystem - Electronic Interlocking Point machine Point machine :Subsystem Point 1 st n-th par Information_End_Position_Arrived Information_End_Position_Arrived Msg_Point_Position **P_UC2.1.2.4: Irregularities** Alternative Scenario: Handle and report End Position out of the other End Position with n-th point machine [P SD 2.1.2.4.8] **Precondition:** The Subsystem - Point is in the state OPERATIONAL. The Subsystem - Point is configured with a non-4-wire interface to the Point machine. The Subsystem - Point is in an End position "Y". **Interaction 2.1.2.4.8.A:** **par** **1.a1** - The Subsystem - Point receives from the 1st Point machine the Information that the Point machine is in an End position "X". **also par** **1.b1** - The Subsystem - Point receives from the n-th Point machine the Information that the Point machine is in an End position "X". **end par** **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".	Whether this SD can occur, depends on the national implementation of P3.	Basic non-4-wire multiple P

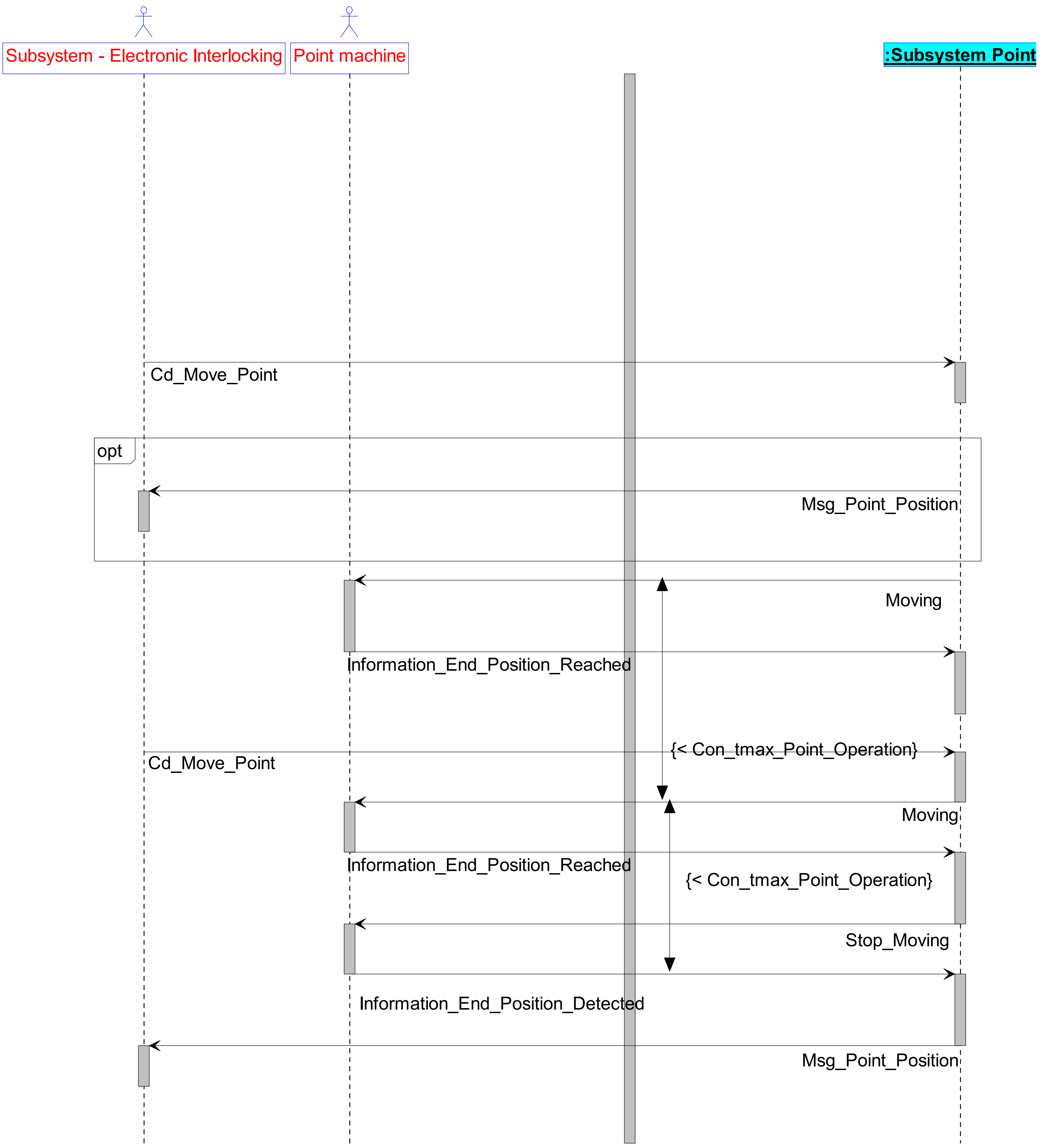
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.5804	Info	P SD 2.1.2.4.9 **P_UC2.1.2.4: Irregularities**  Subsystem - Electronic Interlocking Point machine Point machine :Subsystem Point 1st n-th alt Information_Ability_To_Move_Point opt Information_Ability_To_Move_Point Information_Ability_To_Move_Point_Available_FALSE Msg_Point_Position opt Stop_Moving opt Stop_Moving Alternative Scenario: Handle and report loss of ability to move point with n-th point machines [P SD 2.1.2.4.9] **Precondition**: The Subsystem - Point is in the state OPERATIONAL. The Subsystem - Point is configured with a non-4-wire interface to the Point machine. The Subsystem - Point is configured to observe the Ability to move point. The Subsystem - Point is Able to move point. **Interaction 2.1.2.4.9.A**: **alt** [The 1st Point machine is Unable to move point] **1.a1** - The Subsystem - Point detects from the switch states of the Interface P3 to the 1st Point machine that the 1st Point machine has no Ability to move point. **else alt** [The n-th Point machine is configured with full functionality] **opt** [The n-th Point machine is Unable to move point] **1.b1.a1** - The Subsystem - Point detects from the switch states of the Interface P3 to the n-th Point machine that the n-th Point machine has no Ability to move point. **end opt** **else alt** [internal trigger indicates non-ability to move point] **1.c1** - The Subsystem - Point internal trigger indicates that the Point is Unable to move point. **end alt** **Interaction 2.1.2.4.9.B**: **2.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that it is Unable to move point. **Interaction 2.1.2.4.9.C**: **opt** [The1st point machine is in Moving] **3.a1** - The Subsystem - Point sends the Command to the 1st Point machine to Stop moving the 1st Point machine. **end opt** **opt** [The n-th Point machine is configured with full functionality and is in Moving] **4.a1** - The Subsystem - Point sends the Command to the n-th Point machine to Stop moving the n-th Point machine. **end opt** **Postcondition:** The Subsystem - Point is Unable to move point.	Only applicable if the package [Option Able to move] is used in combination with [Basic non-4-wire multiple P].	Option Able to move

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.5374	Info	P SD 2.1.2.4.10 **P_UC2.1.2.4: Irregularities** Alternative Scenario: Handle and report restoring of Ability to move point with n-th point machine [P SD 2.1.2.4.10] **Precondition**: The Subsystem - Point is in the state OPERATIONAL. The Subsystem - Point is configured with a non-4-wire interface to the Point machine. The Subsystem - Point is configured to observe the Ability to move point. The Subsystem - Point is Unable to move point. **Interaction 2.1.2.4.10.A**: **par** [The 1st Point machine is Able to move point] **1.a1** - The Subsystem - Point detects from the switch states of the Interface P3 to the 1st Point machine that the 1st Point machine has regained Ability to move point. **also par** [The n-th Point machine is Able to move point and is configured with full functionality] **1.b1** - The Subsystem - Point detects from the switch states of the Interface P3 to the n-th Point machine that the n-th Point machine has regained Ability to move point. **also par** [internal trigger indicates ability to move point] **1.c1** - The Subsystem - Point internal trigger indicates that the Point is Able to move point. **end par** **Interaction 2.1.2.4.10.B**: **2.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that it is Able to move point. **Postcondition:** The Subsystem - Point is Able to move point.  Subsystem - Electronic Interlocking Point machine Point machine :Subsystem Point 1 st n-th par Information_Ability_To_Move_Point Information_Ability_To_Move_Point Information_Ability_To_Move_Point_Available_TRUE Msg_Point_Position	Only applicable if the package [Option Able to move] is used in combination with [Basic non-4-wire multiple P].	Option Able to move

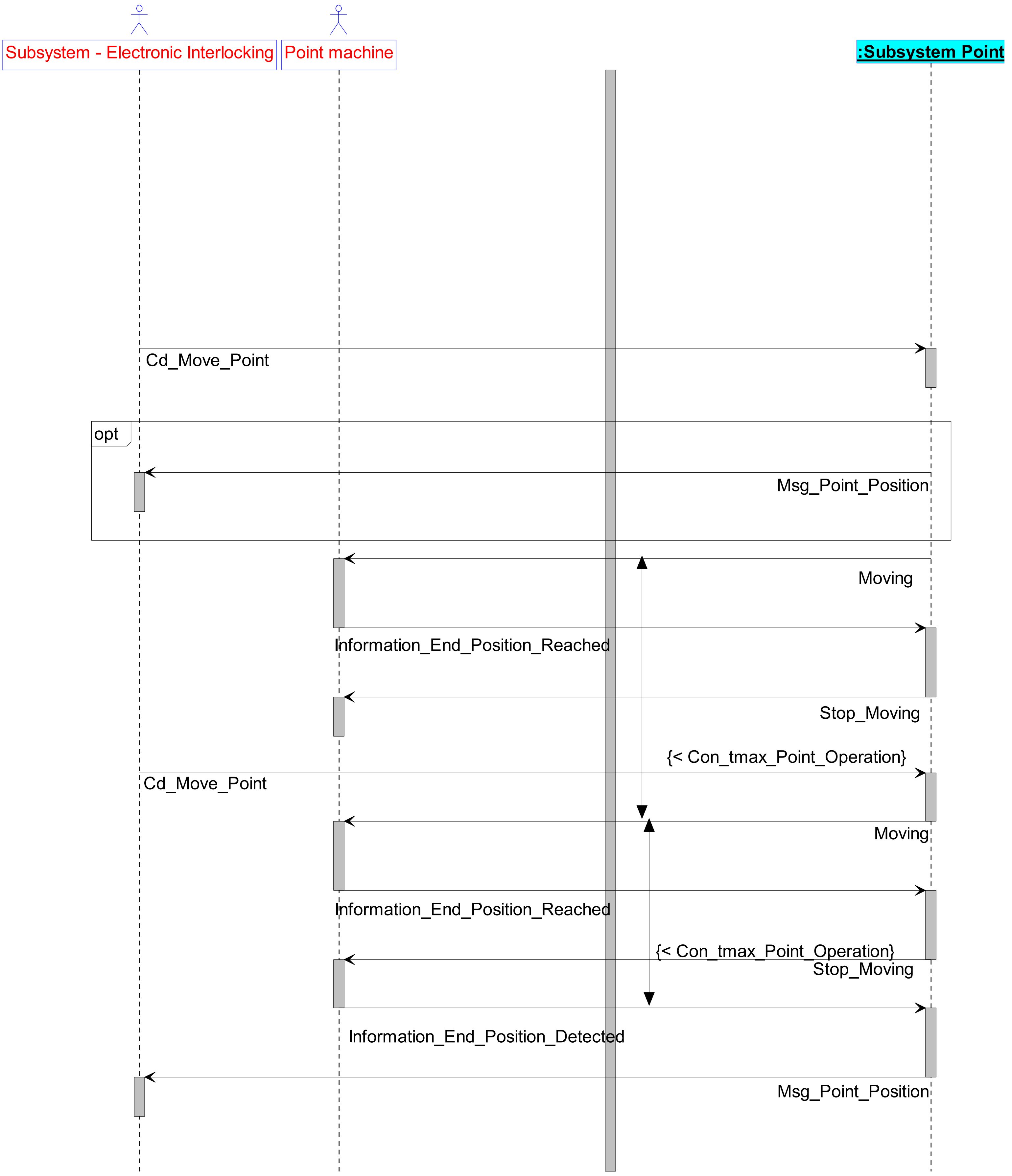
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.7099	Info	P SD 2.1.2.4.11 **P_UC2.1.2.4: Irregularities** Alternative Scenario: Handle Point operation timeout during Redrive [P SD 2.1.2.4.11] **Precondition:** The Subsystem Point is in the state INITIALISING or OPERATIONAL. The Subsystem - Point is in an End position "X" The last Cd_Move_Point Command received was also for End position "X" The last Cd_Move_Point Command received occurred since exiting the state BOOTING. The Subsystem - Point is configured with a non-4-wire interface to the Point machine. The Subsystem - Point is configured as Redrive point. **Interaction 2.1.2.4.11.A:** **alt** [The 1st Point machine was previously in an End position] **1.a1** - The Subsystem - Point receives from the Point machine the Information that the Point machine is in No end position. **opt** [The Subsystem Point is in the state OPERATIONAL.] **1.a2.a1** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in No end position. **end opt** **else alt** [The n-th Point machine was previously in an End position] **1.b1** - The Subsystem - Point receives from the n-th Point machine the Information that the Point is in No end position. **opt** [The Subsystem Point is in the state OPERATIONAL.] **1.b2.a1** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in No end position. **end opt** **end alt** **Interaction 2.1.2.4.11.B:** **par** [The Subsystem - Point is configured for common drive.] **2.a1** - The Subsystem - Point sends the Command to the 1st Point machine to move the Point to an End position "X". At this moment the Subsystem - Point starts the timer Con_tmax_Point_Operation. **2.a2** - The Subsystem - Point sends the Command to the n-th Point machine to move the Point to an End position "X". **also par** [The Subsystem - Point is configured for individual drive.] **alt** [The 1st Point machine is in No end position] **2.b1.a1** - The Subsystem - Point sends the Command to the 1st Point machine to move the Point to an End position "X". **else alt** [The n-th Point machine is in No end position] **2.b1.a2** - The Subsystem - Point sends the Command to the n-th Point machine to move the Point to an End position "X". **end alt** **end par** **Interaction 2.1.2.4.11.C:** **3.** - The timer Con_tmax_Point_Operation expires resulting in a Failed Movement. **par** **4.a1** The Subsystem - Point sends the Command to the 1st Point machine to stop moving the 1st Point machine. The timer Con_tmax_Point_Operation is reset. **also par** **4.b1** The Subsystem - Point sends the Command to the n-th Point machine to stop moving the n-th Point machine. **end par** **opt** [The Subsystem Point is in the state OPERATIONAL.] **5.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that a Failed Movement has occurred. **end opt** **Postcondition:** The Subsystem - Point is in No end position.  Subsystem - Electronic Interlocking Point machine Point machine :Subsystem Point 1st n-th alt Information_No_End_Position opt Msg_Point_Position Information_No_End_Position opt Msg_Point_Position par Moving Moving alt Moving Moving after {Con_tmax_Point_Operation} par Stop_Moving Stop_Moving opt Msg_Movement_Failed	Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE. Only applicable if the package [Option Redrive] is used in combination with [Basic non-4-wire multiple P].	Option Redrive
Eu.P.6770	Info	P_UC2.2: Point with 4-wire interface	The Subsystem-UseCase "P_UC2.2: Point with 4-wire interface" defines the behaviour of the Subsystem Point which works with a 4-wire	Basic 4-wire single P Basic 4-wire multiple P

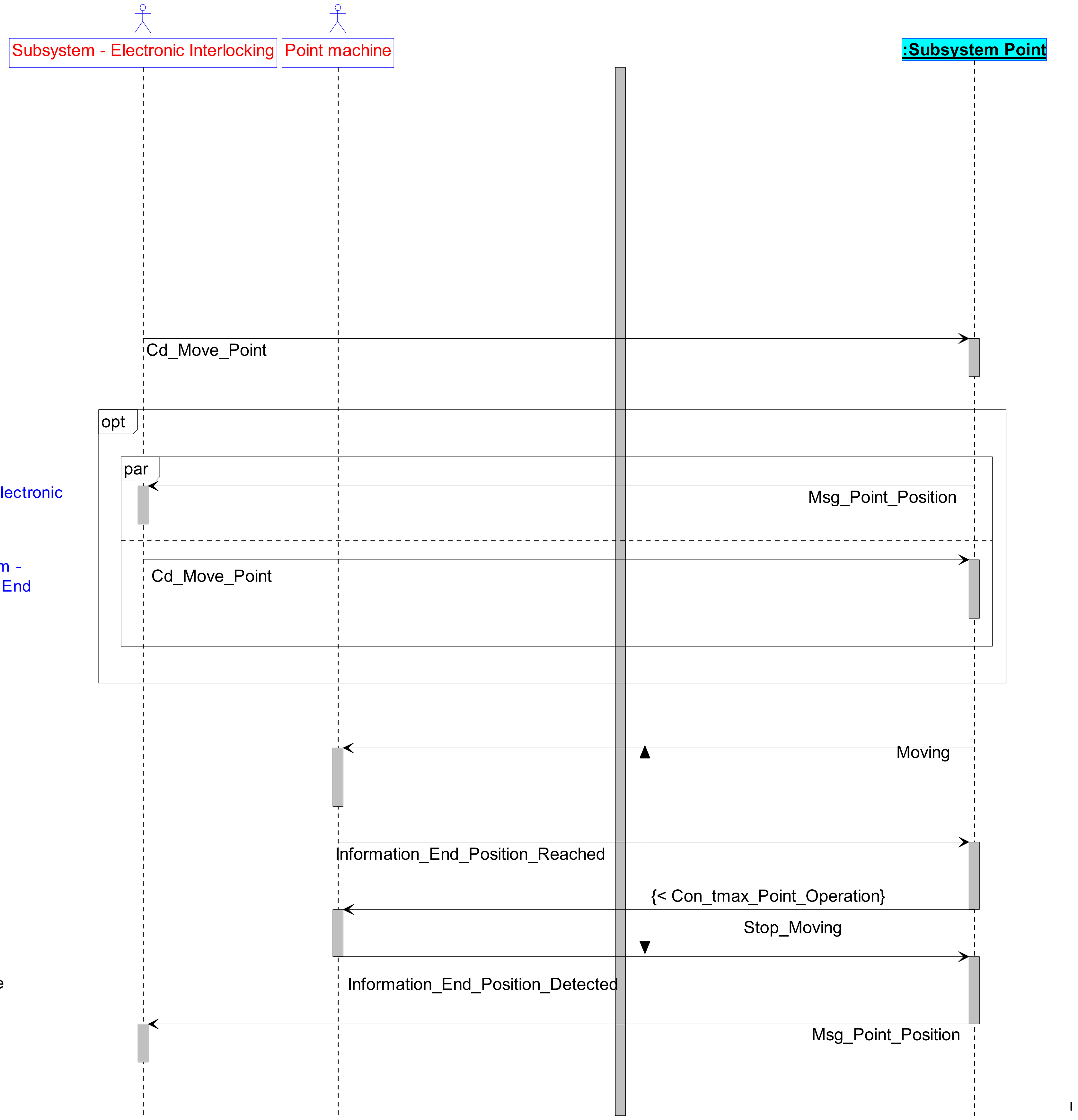
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
			interface. The behaviour will be defined in the following UseCases: P_UC2.2.1: Single point machine P_UC2.2.2: Multiple point machines	
Eu.P.6752	Info	P_UC2.2.1: Single point machine	The Subsystem-UseCase "P_UC2.2.1: Single point machine" defines the behaviour of the Subsystem Point which works with a single point machine via 4-wire interface. The behaviour will be defined in the following UseCases: P_UC2.2.1.1: Commanding and reversing P_UC2.2.1.2: Irregularities	Basic 4-wire single P
Eu.P.6733	Info	P_UC2.2.1.1: Commanding and reversing	The Subsystem-UseCase "P_UC2.2.1.1: Commanding and reversing" defines the behaviour of commanding and reversing a single point machine via 4-wire interface.	Basic 4-wire single P
Eu.P.6740	Info	P SD 2.2.1.1.1 **P_UC2.2.1.1: Commanding and reversing**  Subsystem - Electronic Interlocking Point machine :Subsystem Point Cd_Move_Point opt Msg_Point_Position Moving Information_End_Position_Reached {< Con_tmax_Point_Operation} Stop_Moving Information_End_Position_Detected Msg_Point_Position Main Success Scenario: Moving of the Point with a single point machine 4W [P SD 2.2.1.1.1] **Precondition:** The Subsystem - Point is in the state OPERATIONAL. The Subsystem - Point is configured with a 4-wire interface to the Point machine. The Subsystem - Point is in: - an End position "Y", or - No end position, or - an Unintended position. **Interaction 2.2.1.1.1.A:** **1.** - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position "X". **Interaction 2.2.1.1.1.B:** **opt** [The Subsystem Point was previously in an End position or a Unintended position] **2.a1** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in No end position. **end opt** **3.** 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 the timer Con_tmax_Point_Operation. **Interaction 2.2.1.1.1.C:** **4.** - The Subsystem - Point receives from the Point machine the Information that the Point machine has reached End position "Y" as unsafe information. **5.** The Subsystem - Point sends the Command to the Point machine to stop moving the Point machine. **6.** - The Subsystem - Point receives from the Point machine the Information that the Point machine is in a detected End position "X". The timer Con_tmax_Point_Operation is reset. **7.** 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.	Basic 4-wire single P

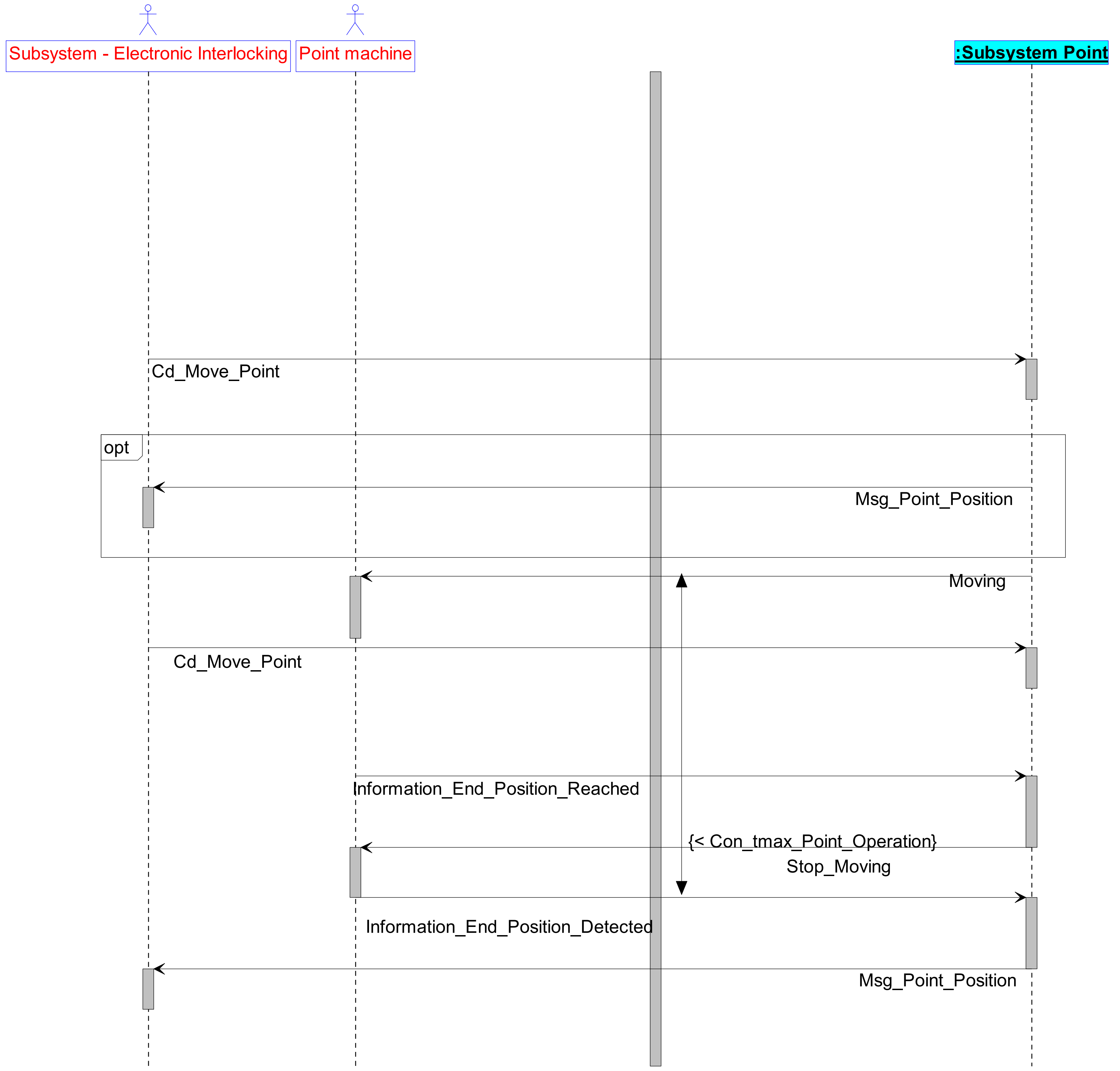
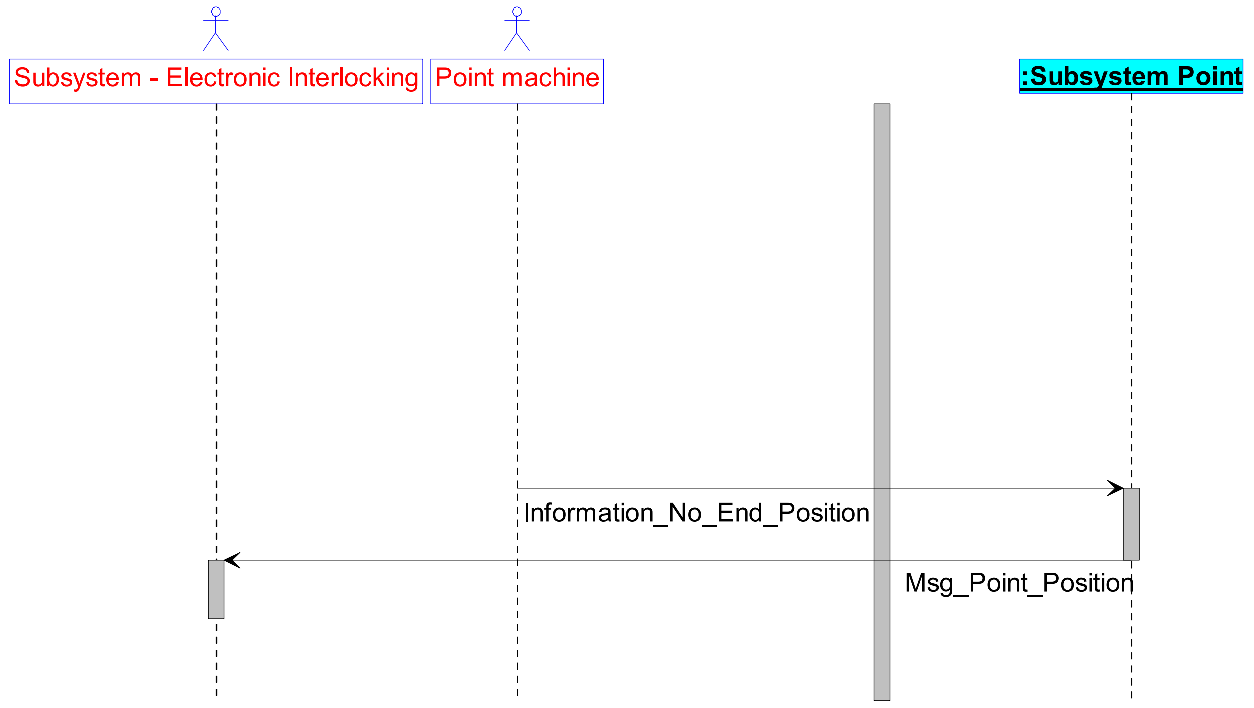
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.6736	Info	P SD 2.2.1.1.2 **P UC2.2.1.1: Commanding and reversing**  Subsystem - Electronic Interlocking Point machine :Subsystem Point Cd_Move_Point opt Msg_Point_Position Moving Cd_Move_Point {< Con_tmax_Point_Operation} Moving Information_End_Position_Reached {< Con_tmax_Point_Operation} Stop_Moving Information_End_Position_Detected Msg_Point_Position Alternative Scenario: Reversing Point 4W [P SD 2.2.1.1.2] **Precondition:** The Subsystem - Point is in the state OPERATIONAL. The Subsystem - Point is configured with a 4-wire interface to the Point machine. The Subsystem - Point is in: - an End position "Y", or - No end position, or - an Unintended position. **Interaction 2.2.1.1.2.A:** **1.** - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position "X". **Interaction 2.2.1.1.2.B:** **opt** [The Subsystem Point was previously in an End position or a Unintended position] **2.a1** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that Point is in No end position. **end opt** **3.** 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 the timer Con_tmax_Point_Operation. **Interaction 2.2.1.1.2.C:** **4.** - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position "Y". **Interaction 2.2.1.1.2.D:** **5.** - The Subsystem - Point sends the Command to the Point machine to move the Point to an End position "Y". **6.** - The Subsystem - Point receives from the Point machine the Information that the Point machine has reached End position "Y" as unsafe information. **7.** The Subsystem - Point sends the Command to the Point machine to stop moving the Point machine. **8.** - The Subsystem - Point receives from the Point machine the Information that the Point machine is in a detected End position "Y". The timer Con_tmax_Point_Operation is reset. **9.** 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".	Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE.	Basic 4-wire single P

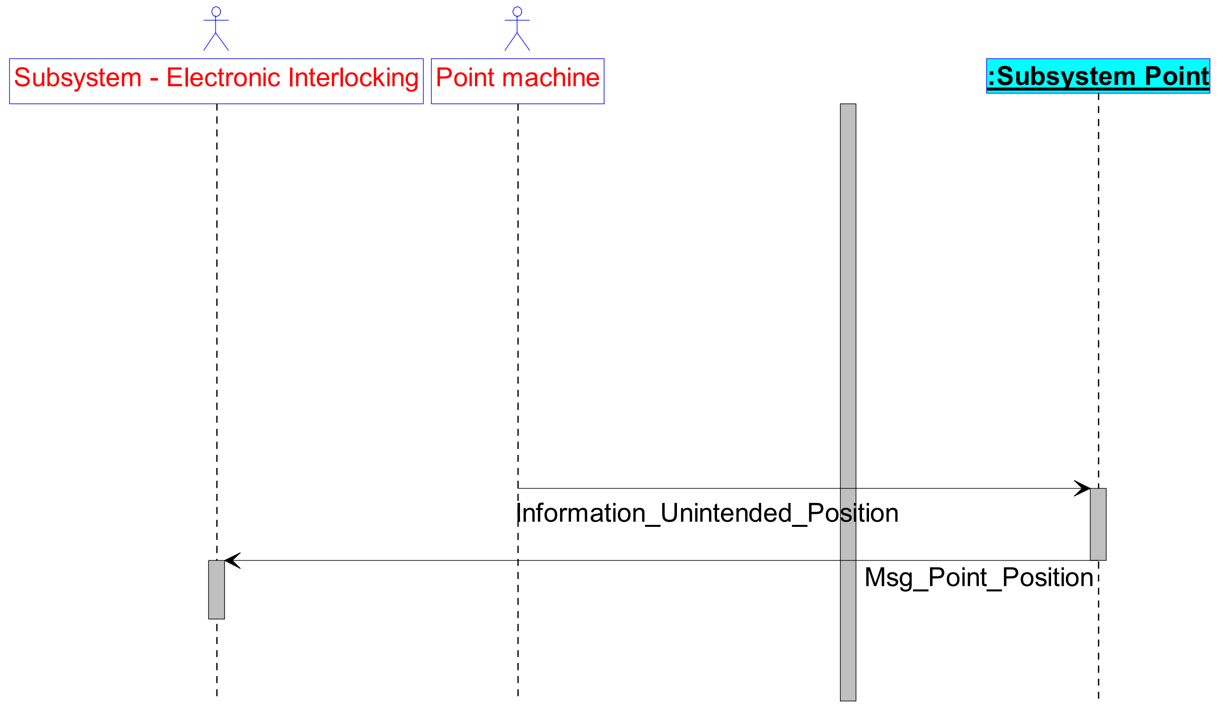
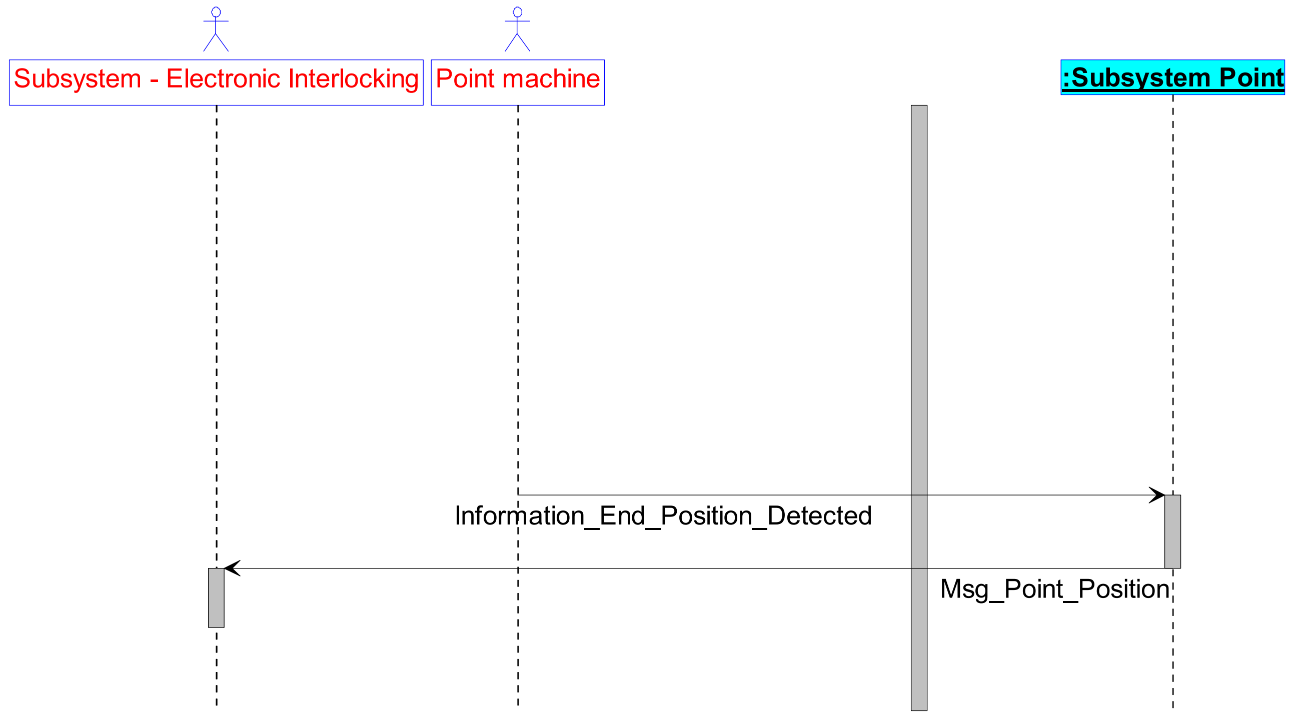
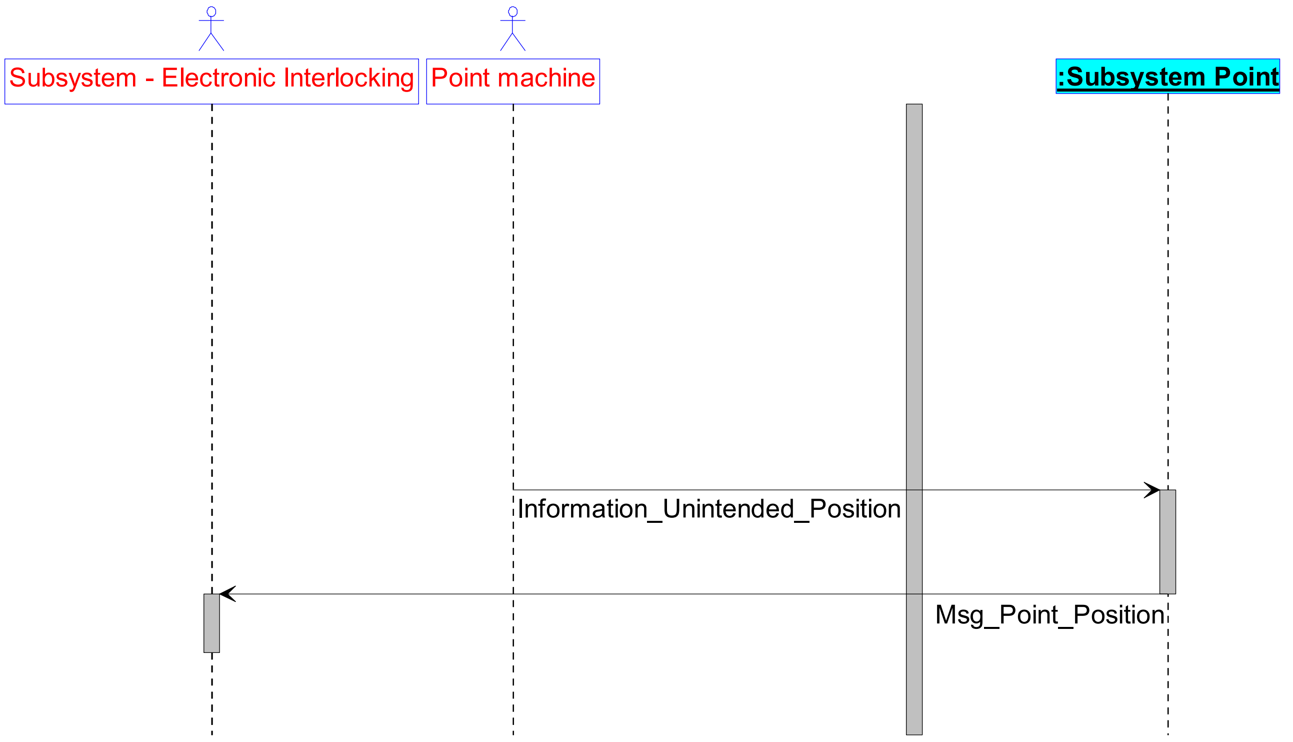
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.6737	Info	P SD 2.2.1.1.3 **P_UC2.2.1.1: Commanding and reversing** Alternative Scenario: Reversing Point 4W after End position reached [P SD 2.2.1.1.3] **Precondition:** The Subsystem - Point is in the state OPERATIONAL. The Subsystem - Point is configured with a 4-wire interface to the Point machine. The Subsystem - Point is in: - an End position "Y", or - No end position, or - an Unintended position. **Interaction 2.2.1.1.3.A:** **1.** - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position "X". **Interaction 2.2.1.1.3.B:** **opt** [The Subsystem Point was previously in an End position or a Unintended position] **2.a1** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that Point is in No end position. **end opt** **3.** 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 the timer Con_tmax_Point_Operation. **4.** - The Subsystem - Point receives from the Point machine the Information that the Point machine has reached End position "X" as unsafe information. **Interaction 2.2.1.1.3.C:** **5.** - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position "Y". **6.** - The Subsystem - Point sends the Command to the Point machine to move the Point to an End position "Y". **7.** - The Subsystem - Point receives from the Point machine the Information that the Point machine has reached End position "Y" as unsafe information. **8.** The Subsystem - Point sends the Command to the Point machine to stop moving the Point machine. **9.** - The Subsystem - Point receives from the Point machine the Information that the Point machine is in a detected End position "Y". The timer Con_tmax_Point_Operation is reset. **10.** 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". 	Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE.	Basic 4-wire single P

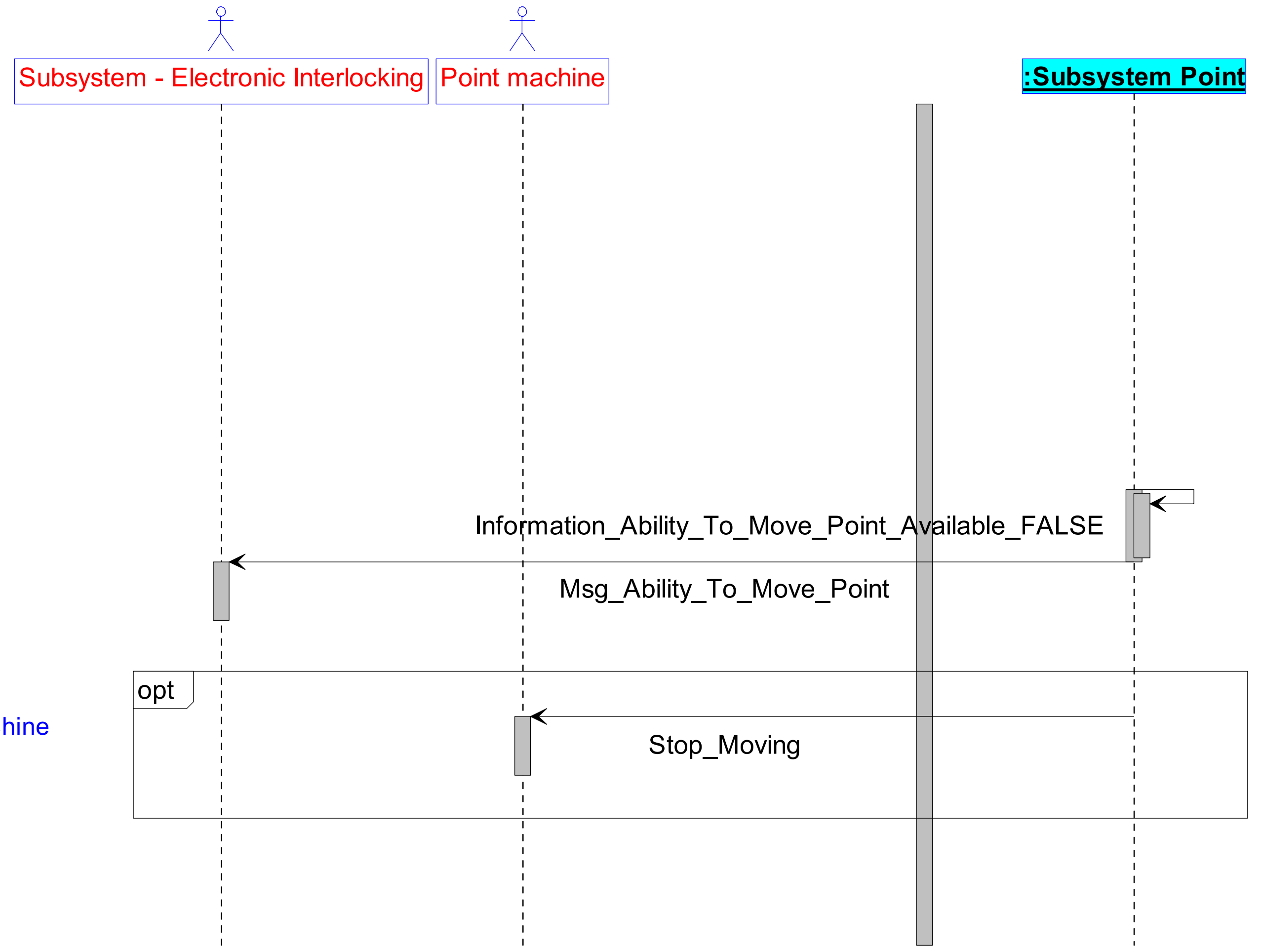
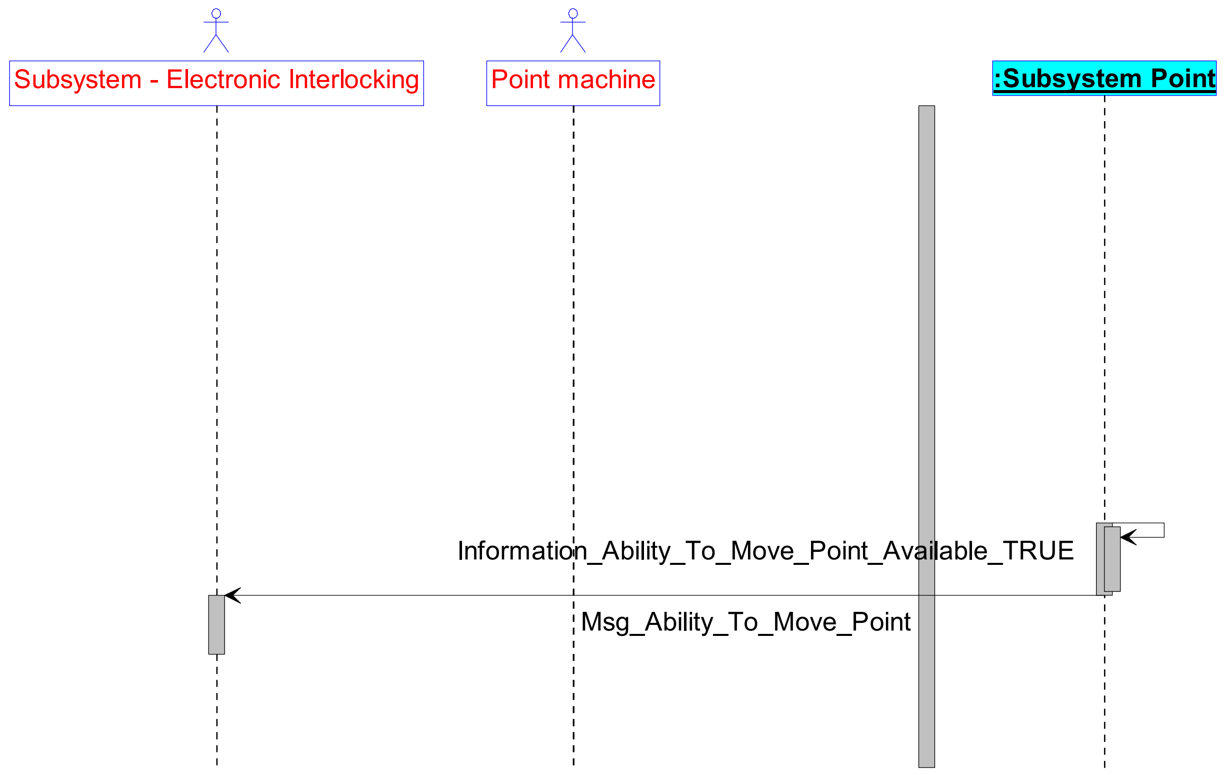
Subsystem - Electronic Interlocking
Point machine
:Subsystem Point
Cd_Move_Point
opt
Msg_Point_Position
Moving
Information_End_Position_Reached
{< Con_tmax_Point_Operation}
Cd_Move_Point
Moving
Information_End_Position_Reached
{< Con_tmax_Point_Operation}
Stop_Moving
Information_End_Position_Detected
Msg_Point_Position

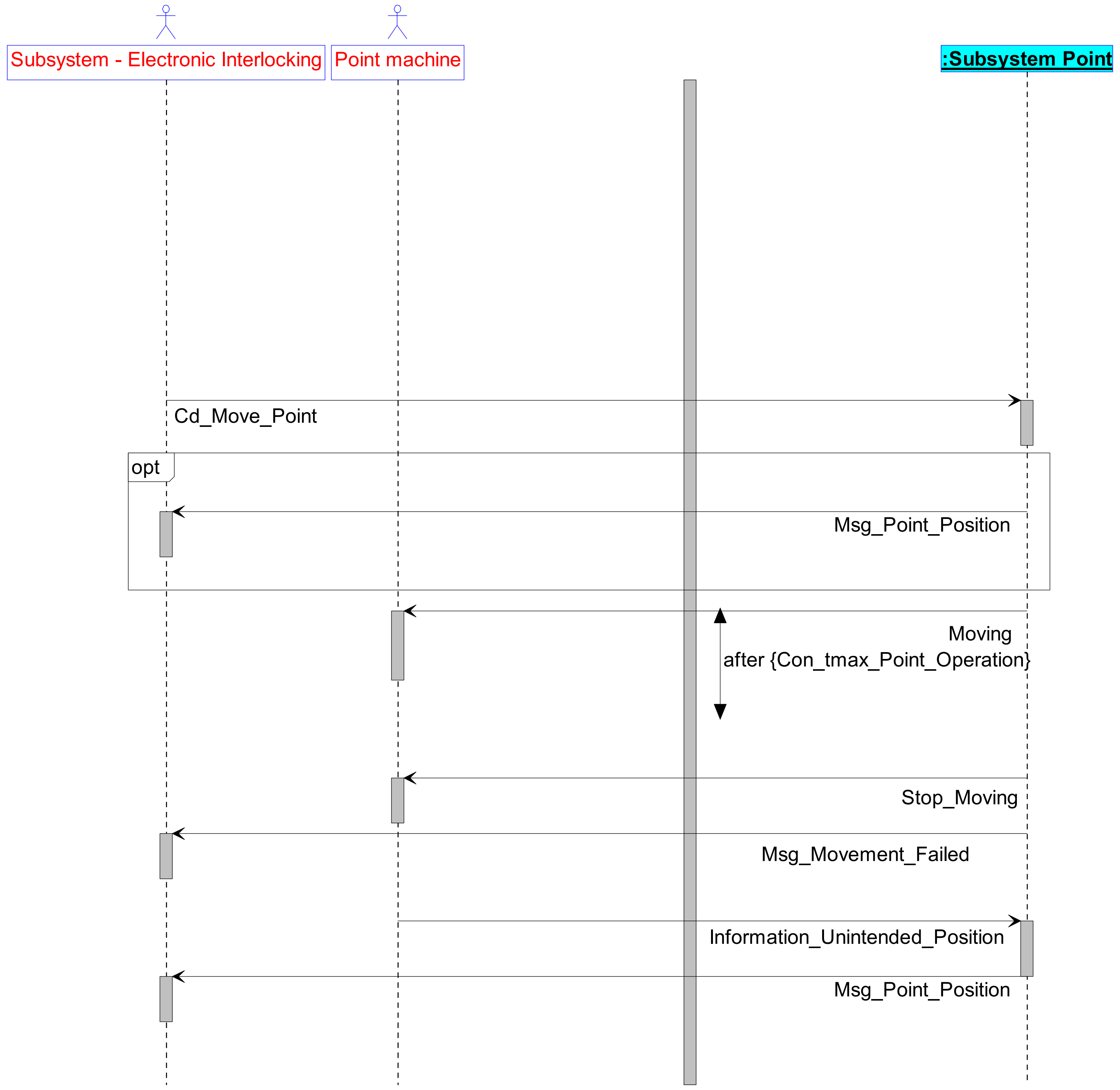
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.6738	Info	P SD 2.2.1.1.4 **P UC2.2.1.1: Commanding and reversing** Alternative Scenario: Reversing Point 4W before End position is detected[P SD 2.2.1.1.4] **Precondition:** The Subsystem - Point is in the state OPERATIONAL. The Subsystem - Point is configured with a 4-wire interface to the Point machine. The Subsystem - Point is in: - an End position "Y", or - No end position, or - an Unintended position. **Interaction 2.2.1.1.4.A:** **1.** - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position "X". **Interaction 2.2.1.1.4.B:** **opt** [The Subsystem Point was previously in an End position or a Unintended position] **2.a1** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that Point is in No end position. **end opt** **3.** 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 the timer Con_tmax_Point_Operation. **4.** - The Subsystem - Point receives from the Point machine the Information that the Point machine has reached End position "X" as unsafe information. **5.** The Subsystem - Point sends the Command to the Point machine to stop moving the Point machine. **Interaction 2.2.1.1.4.C:** **6.** - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position "Y". **7.** - 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 starts the timer Con_tmax_Point_Operation. **8.** - The Subsystem - Point receives from the Point machine the Information that the Point machine has reached End position "Y" as unsafe information. **9.** The Subsystem - Point sends the Command to the Point machine to stop moving the Point machine. **10.** - The Subsystem - Point receives from the Point machine the Information that the Point machine is in a detected End position "Y". The timer Con_tmax_Point_Operation is reset. **11.** 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".  Subsystem - Electronic Interlocking Point machine :Subsystem Point Cd_Move_Point opt Msg_Point_Position Moving Information_End_Position_Reached Stop_Moving {< Con_tmax_Point_Operation} Cd_Move_Point Moving Information_End_Position_Reached {< Con_tmax_Point_Operation} Stop_Moving Information_End_Position_Detected Msg_Point_Position	Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE.	Basic 4-wire single P

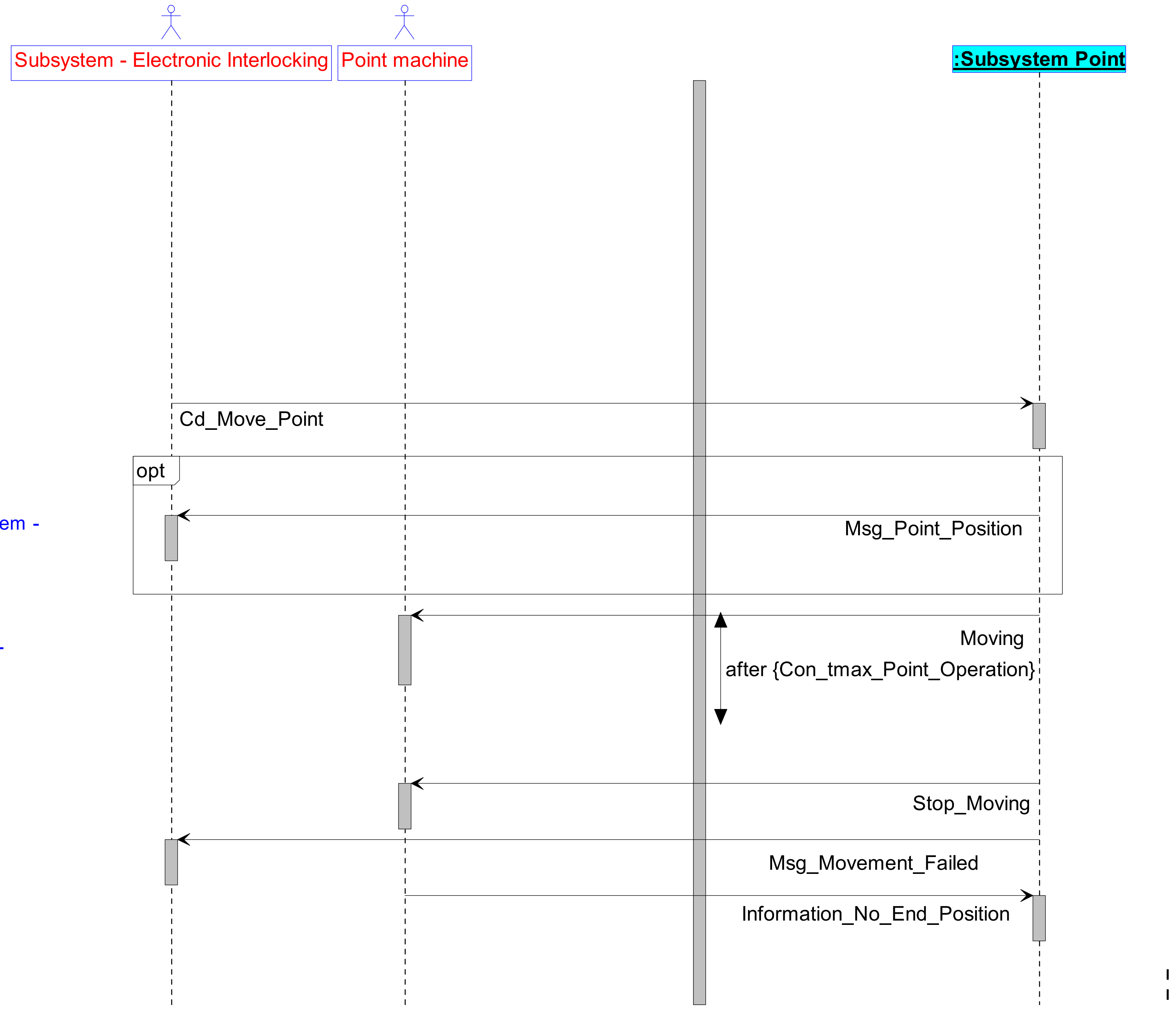
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.6734	Info	P SD 2.2.1.1.5 **P_UC2.2.1.1: Commanding and reversing** Alternative Scenario: Moving of the Point with repeated command of moving 4W case 1[P SD 2.2.1.1.5] **Precondition:** The Subsystem - Point is in the state OPERATIONAL. The Subsystem - Point is configured with a 4-wire interface to the Point machine. The Subsystem - Point is in: - an End position "Y", or - No end position, or - an Unintended position. **Interaction 2.2.1.1.5.A:** **1.** - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position "X". **Interaction 2.2.1.1.5.B:** **opt** [The Subsystem Point was previously in an End position or a Unintended position] **par** **2.a1.a1** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in No end position. **also par** **2.a1.b1** - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position "X". **end par** **end opt** **3.** The Subsystem - Point ignores the command from the Subsystem - Electronic Interlocking. **4.** 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 the timer Con_tmax_Point_Operation. **Interaction 2.2.1.1.5.C:** **5.** - The Subsystem - Point receives from the Point machine the Information that the Point machine has reached End position "Y" as unsafe information. **6.** The Subsystem - Point sends the Command to the Point machine to stop moving the Point machine. **7.** - The Subsystem - Point receives from the Point machine the Information that the Point machine is in a detected End position "X". The timer Con_tmax_Point_Operation is reset. **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".  Subsystem - Electronic Interlocking Point machine :Subsystem Point Cd_Move_Point opt par Msg_Point_Position Cd_Move_Point Moving Information_End_Position_Reached {< Con_tmax_Point_Operation} Stop_Moving Information_End_Position_Detected Msg_Point_Position	Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE.	Basic 4-wire single P

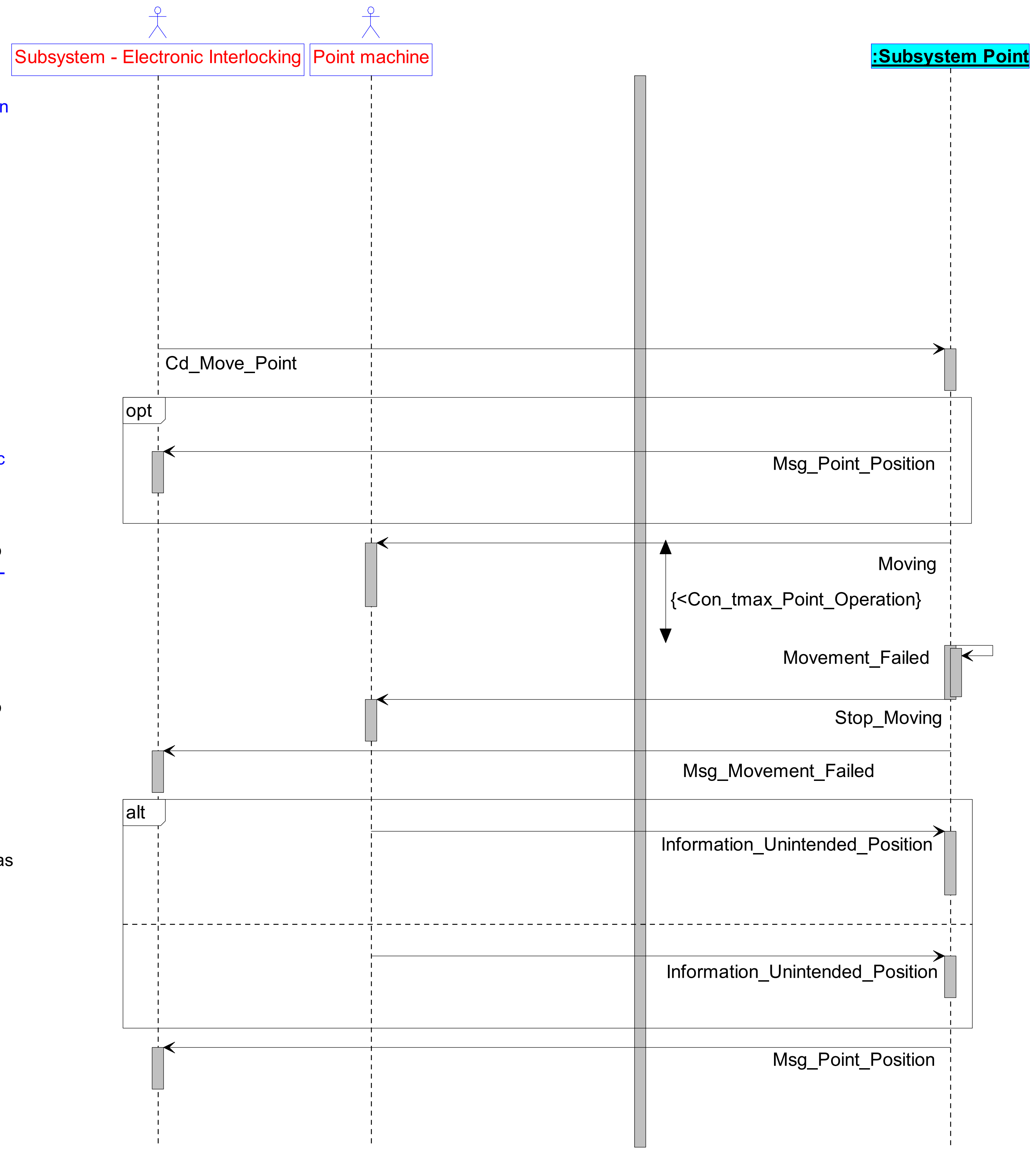
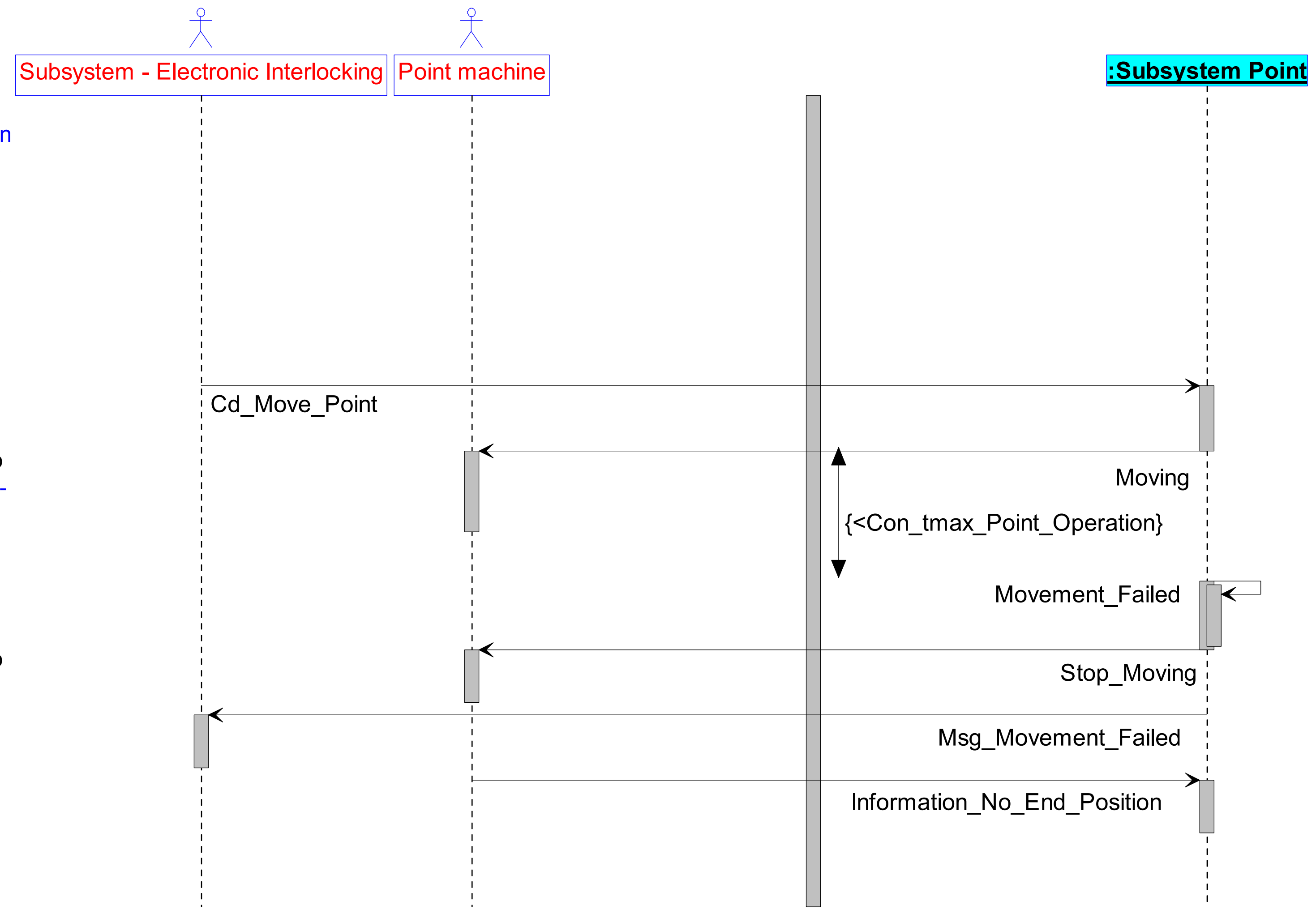
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.6735	Info	P SD 2.2.1.1.6 **P_UC2.2.1.1: Commanding and reversing** Alternative Scenario: Moving of the Point with repeated command of moving 4W case 2[P SD 2.2.1.1.6] **Precondition:** The Subsystem - Point is in the state OPERATIONAL. The Subsystem - Point is configured with a 4-wire interface to the Point machine. The Subsystem - Point is in: - an End position "Y", or - No end position, or - an Unintended position. **Interaction 2.2.1.1.6.A:** **1.** - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position "X". **Interaction 2.2.1.1.6.B:** **opt** [The Subsystem Point was previously in an End position or a Unintended position] **2.a1** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in No end position. **end opt** **3.** 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 the timer Con_tmax_Point_Operation. **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 2.2.1.1.6.C:** **6.** - The Subsystem - Point receives from the Point machine the Information that the Point machine has reached End position "Y" as unsafe information. **7.** The Subsystem - Point sends the Command to the Point machine to stop moving the Point machine. **8.** The Subsystem - Point receives from the Point machine the Information that the Point machine is in a detected End position "X". The timer Con_tmax_Point_Operation is reset. **9.** 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".  Subsystem - Electronic Interlocking Point machine :Subsystem Point Cd_Move_Point opt Msg_Point_Position Moving Cd_Move_Point Information_End_Position_Reached {< Con_tmax_Point_Operation} Stop_Moving Information_End_Position_Detected Msg_Point_Position	Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE.	Basic 4-wire single P
Eu.P.6741	Info	P_UC2.2.1.2: Irregularities	The Subsystem-UseCase "P_UC2.2.1.2: Irregularities" defines the behaviour of the Subsystem Point which works with a single point machine via 4-wire interface, when an irregularity occurs.	Basic 4-wire single P
Eu.P.6746	Info	P SD 2.2.1.2.1 **P_UC2.2.1.2: Irregularities** Alternative Scenario: Handle and report No end position with single point machine [P SD 2.2.1.2.1] **Precondition:** The Subsystem - Point is in the state OPERATIONAL. The Subsystem - Point is configured with a 4-wire interface to the Point machine. The Subsystem - Point is in: - an End position "Y", or - a Unintended position **Interaction 2.2.1.2.1.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.  Subsystem - Electronic Interlocking Point machine :Subsystem Point Information_No_End_Position Msg_Point_Position		Basic 4-wire single P

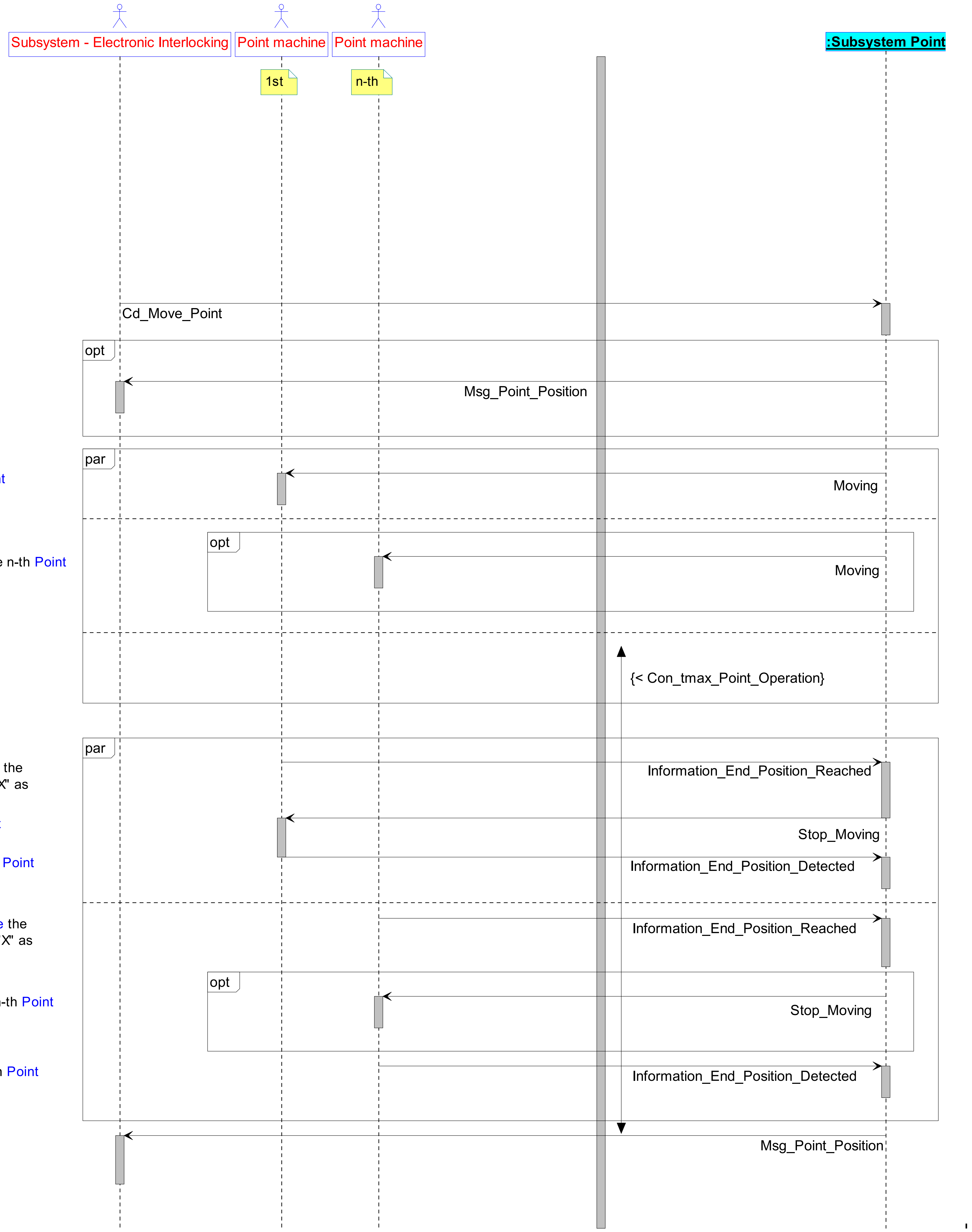
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.6751	Info	P SD 2.2.1.2.2 **P_UC2.2.1.2: Irregularities**  Subsystem - Electronic Interlocking Point machine :Subsystem Point Information_Unintended_Position Msg_Point_Position Alternative Scenario: Handle and report Unintended position with single point machine [P SD 2.2.1.2.2] **Precondition:** The Subsystem - Point is in the state OPERATIONAL. The Subsystem - Point is configured with a 4-wire interface to the Point machine. The Subsystem - Point is in: - an End position "Y", or - No end position. **Interaction 2.2.1.2.2.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.		Basic 4-wire single P
Eu.P.6742	Info	P SD 2.2.1.2.3 **P_UC2.2.1.2: Irregularities**  Subsystem - Electronic Interlocking Point machine :Subsystem Point Information_End_Position_Detected Msg_Point_Position Alternative Scenario: Handle and report commanded End Position with single point machine via 4-wire interface [P SD 2.2.1.2.3] **Precondition:** The Subsystem - Point is in the state OPERATIONAL. The Subsystem - Point is configured with a 4-wire interface to the Point machine AND the last commanded position is End position "Y". The Subsystem - Point is in: - No end position, or - a Unintended position. **Interaction 2.2.1.2.3.A:** **1.** - The Subsystem - Point receives from the Point machine the Information that the Point machine is in an End position "Y". **2.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in an End position. **Postcondition:** The Subsystem - Point is in an End position "Y".		Basic 4-wire single P
Eu.P.6747	Info	P SD 2.2.1.2.4 **P_UC2.2.1.2: Irregularities**  Subsystem - Electronic Interlocking Point machine :Subsystem Point Information_Unintended_Position Msg_Point_Position Alternative Scenario: Handle and report opposite End Position with single point machine via 4-wire interface [P SD 2.2.1.2.4] **Precondition:** The Subsystem - Point is in the state OPERATIONAL. The Subsystem - Point is configured with a 4-wire interface to the Point machine AND the last commanded position is End position "Y". The Subsystem - Point is in: - an End position "Y", or - No end position. **Interaction 2.2.1.2.4.A:** **1.** - The Subsystem - Point receives from the Point machine the pattern that represents the opposite End position "X", which is interpreted as Information that the Point machine is in an 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.		Basic 4-wire single P

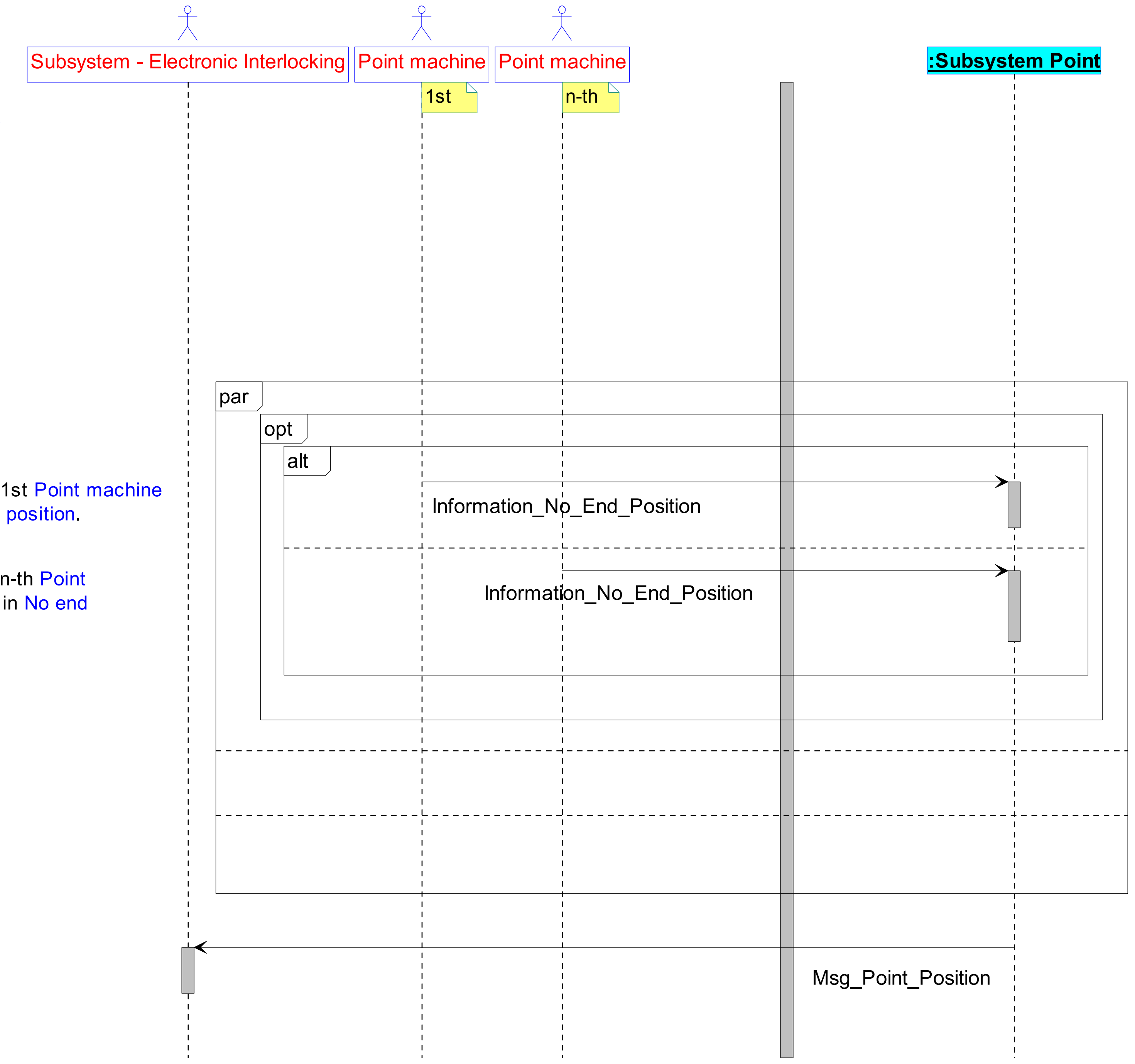
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.6745	Info	P SD 2.2.1.2.5 **P_UC2.2.1.2: Irregularities** Alternative Scenario: Handle and report loss of ability to move point with single point machine [P SD 2.2.1.2.5] **Precondition**: The Subsystem - Point is in the state OPERATIONAL. The Subsystem - Point is configured with a 4-wire interface to the Point machine. The Subsystem - Point is configured to observe the Ability to move point. The Subsystem - Point is in: - Able to move point **Interaction 2.2.1.2.5.A**: **1** - The Subsystem - Point internal trigger indicates that the Subsystem - Point is Unable to move point **2.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that it is Unable to move point. **Interaction 2.2.1.2.5.B**: **opt** [The point machine is in Moving] **3.a1** - The Subsystem - Point sends the Command to the Point machine to Stop moving the Point machine. **end opt** **Postcondition:** The Subsystem - Point is in: - Unable to move point  Subsystem - Electronic Interlocking Point machine :Subsystem Point Information_Ability_To_Move_Point_Available_FALSE Msg_Ability_To_Move_Point opt Stop_Moving	Only applicable if the package [Option Able to move] is used in combination with [Basic 4-wire single P].	Option Able to move
Eu.P.6750	Info	P SD 2.2.1.2.6 **P_UC2.2.1.2: Irregularities** Alternative Scenario: Handle and report restoring of Ability to move point with single point machine [P SD 2.2.1.2.6] **Precondition**: The Subsystem - Point is in the state OPERATIONAL. The Subsystem - Point is configured with a 4-wire interface to the Point machine. The Subsystem - Point is configured to observe the Ability to move point. The Subsystem - Point is in: - Unable to move point **Interaction 2.2.1.2.6.A**: **1.** - The Subsystem - Point internal trigger indicates that the Subsystem - Point is Able to move point. **2.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that it is Able to move point. **Postcondition:** The Subsystem - Point is in: - Able to move point  Subsystem - Electronic Interlocking Point machine :Subsystem Point Information_Ability_To_Move_Point_Available_TRUE Msg_Ability_To_Move_Point	Only applicable if the package [Option Able to move] is used in combination with [Basic 4-wire single P].	Option Able to move

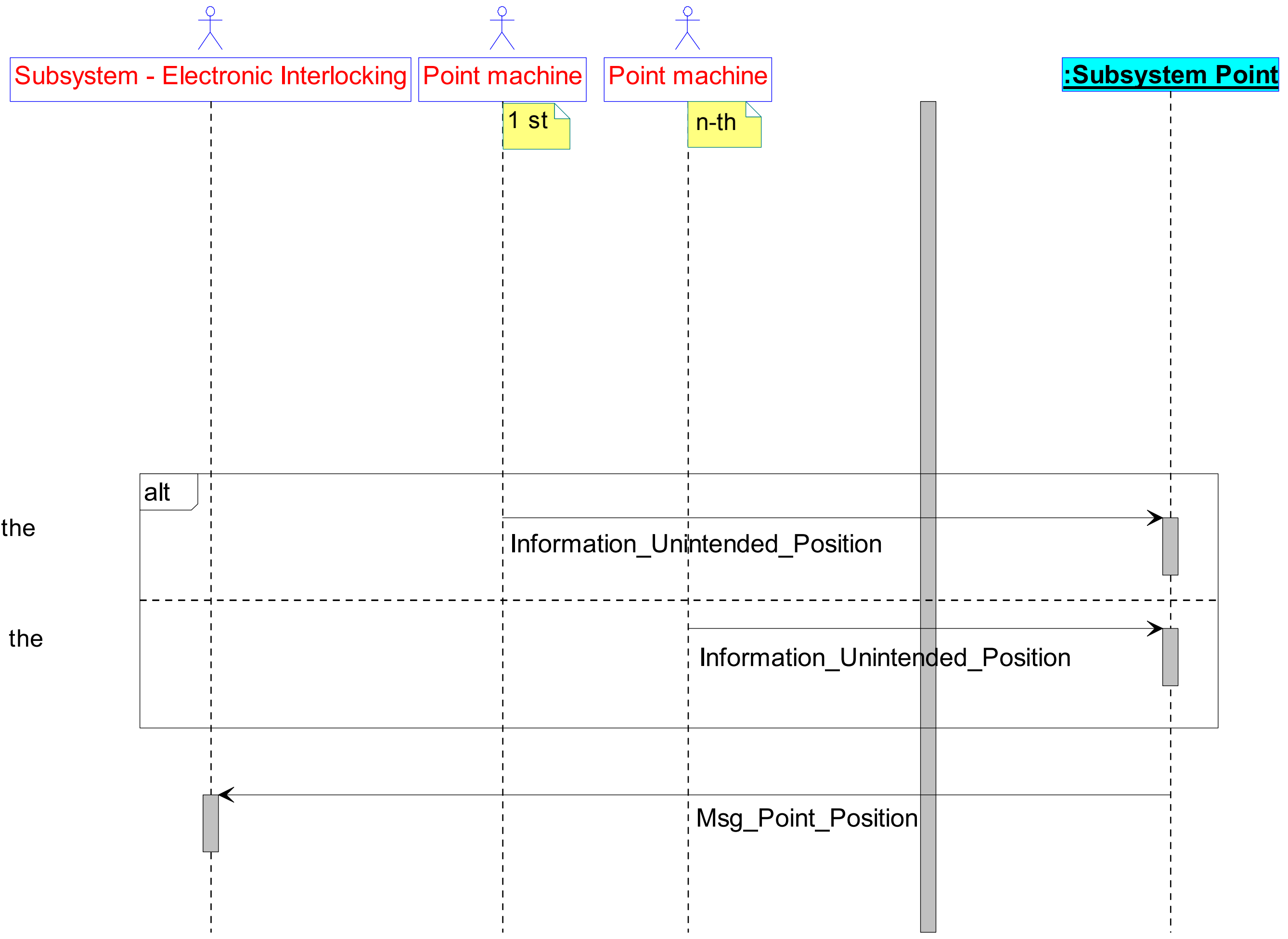
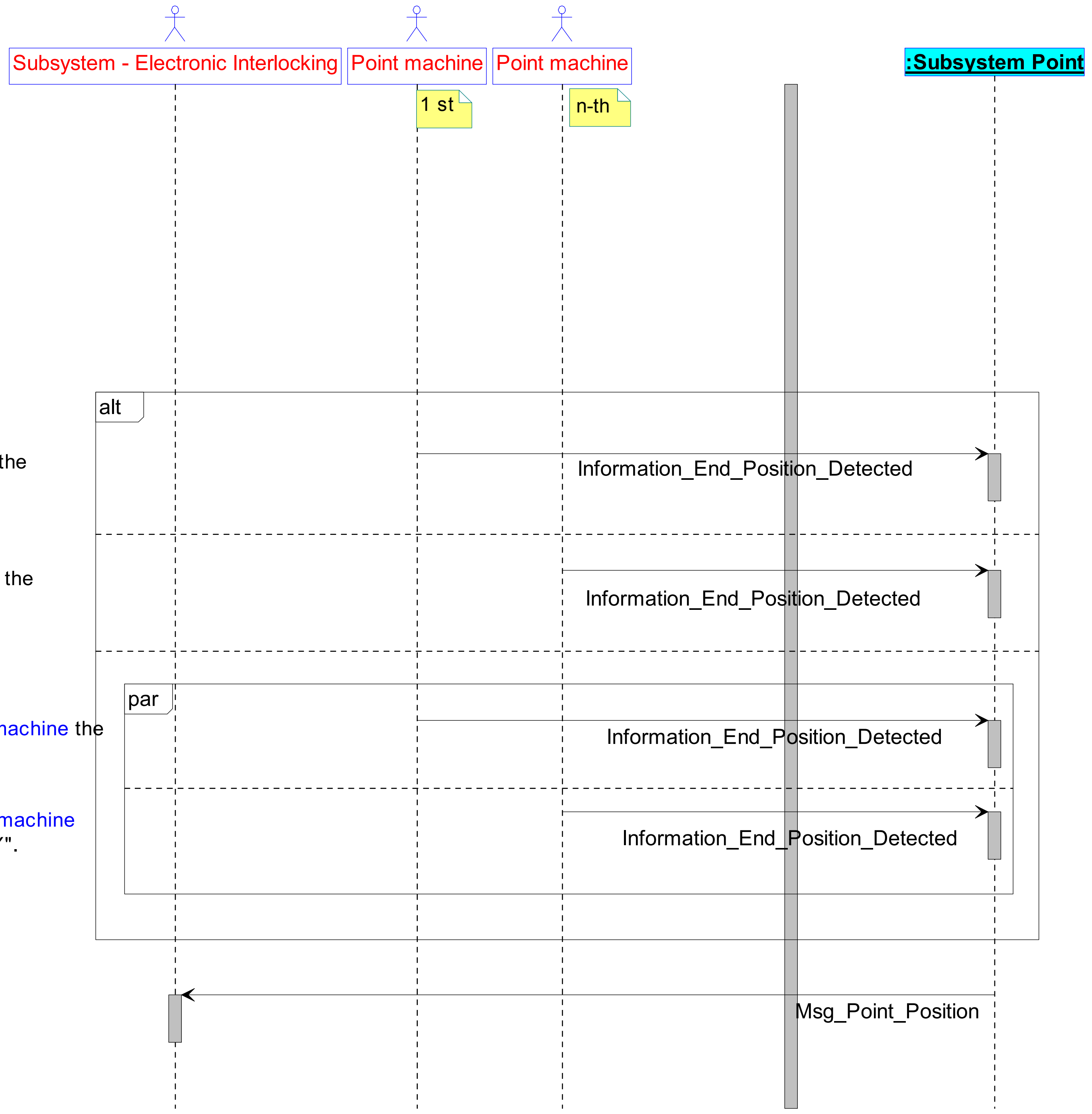
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.6748	Info	P SD 2.2.1.2.7 **P_UC2.2.1.2: Irregularities** Alternative Scenario: Handle and report Point operation timeout with position change 4W case 1 [P SD 2.2.1.2.7] **Precondition:** The Subsystem - Point is in the state OPERATIONAL. The Subsystem - Point is configured with a 4-wire interface to the Point machine. The Subsystem - Point is in: - an End position "Y", or - No end position, or - an Unintended position. **Interaction 2.2.1.2.7.A:** **1.** - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position "X". **opt** [The Subsystem Point was previously in an End position or a Unintended position] **2.a1** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in No end position. end opt **3.** 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 the timer Con_tmax_Point_Operation. **Interaction 2.2.1.2.7.B:** **4.** - The timer Con_tmax_Point_Operation expires resulting in a Failed Movement. **5.** The Subsystem - Point sends the Command to the Point machine to stop moving the Point. **6.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that a Failed Movement has occurred. **Interaction 2.2.1.2.7.C:** **7.** - The Subsystem - Point receives from the Point machine the Information that the Point machine is in a Unintended position. **8.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in Unintended position. **Postcondition:** The Subsystem - Point is in a Unintended position.  Subsystem - Electronic Interlocking Point machine :Subsystem Point Cd_Move_Point opt Msg_Point_Position Moving after {Con_tmax_Point_Operation} Stop_Moving Msg_Movement_Failed Information_Unintended_Position Msg_Point_Position	Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE.	Basic 4-wire single P

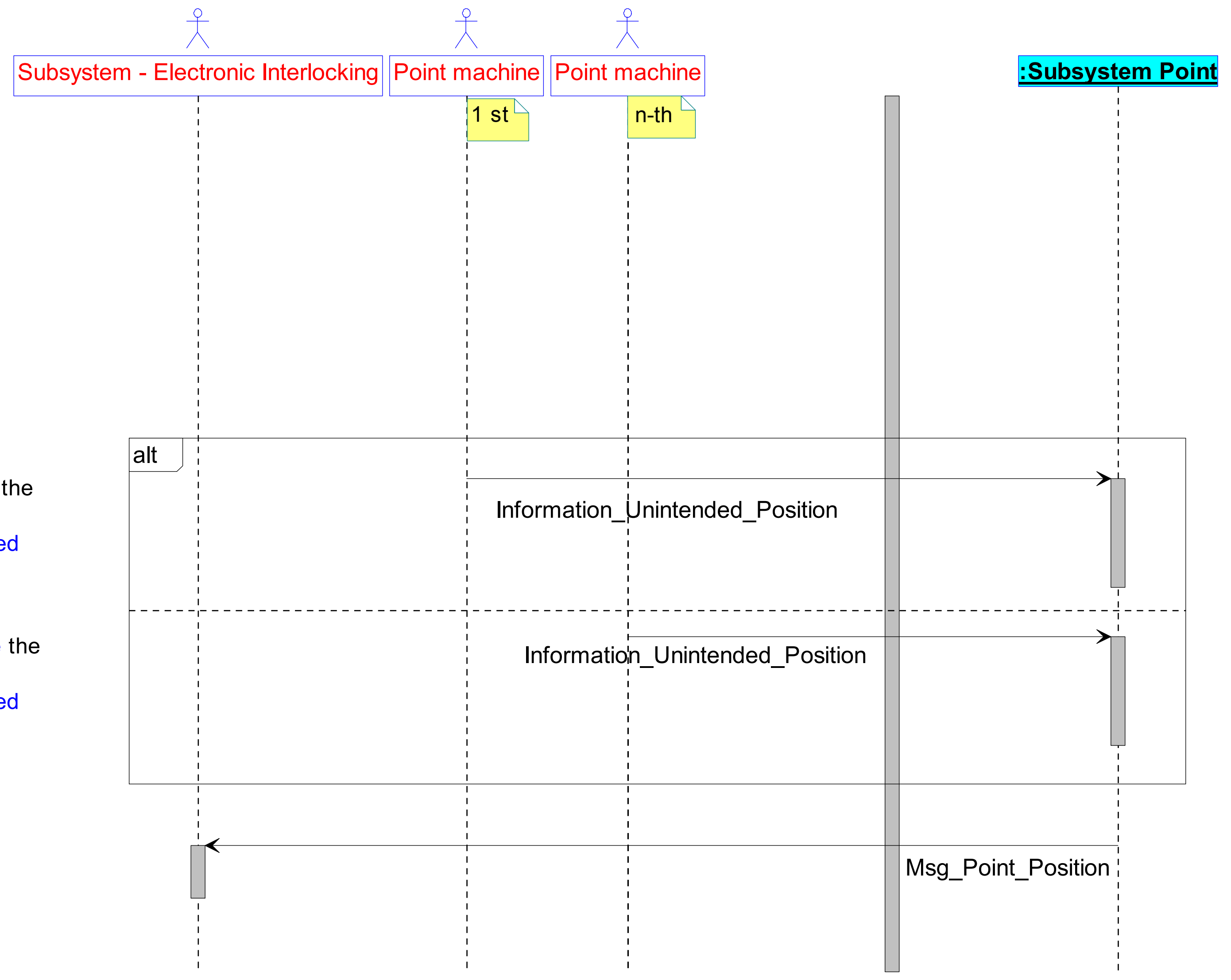
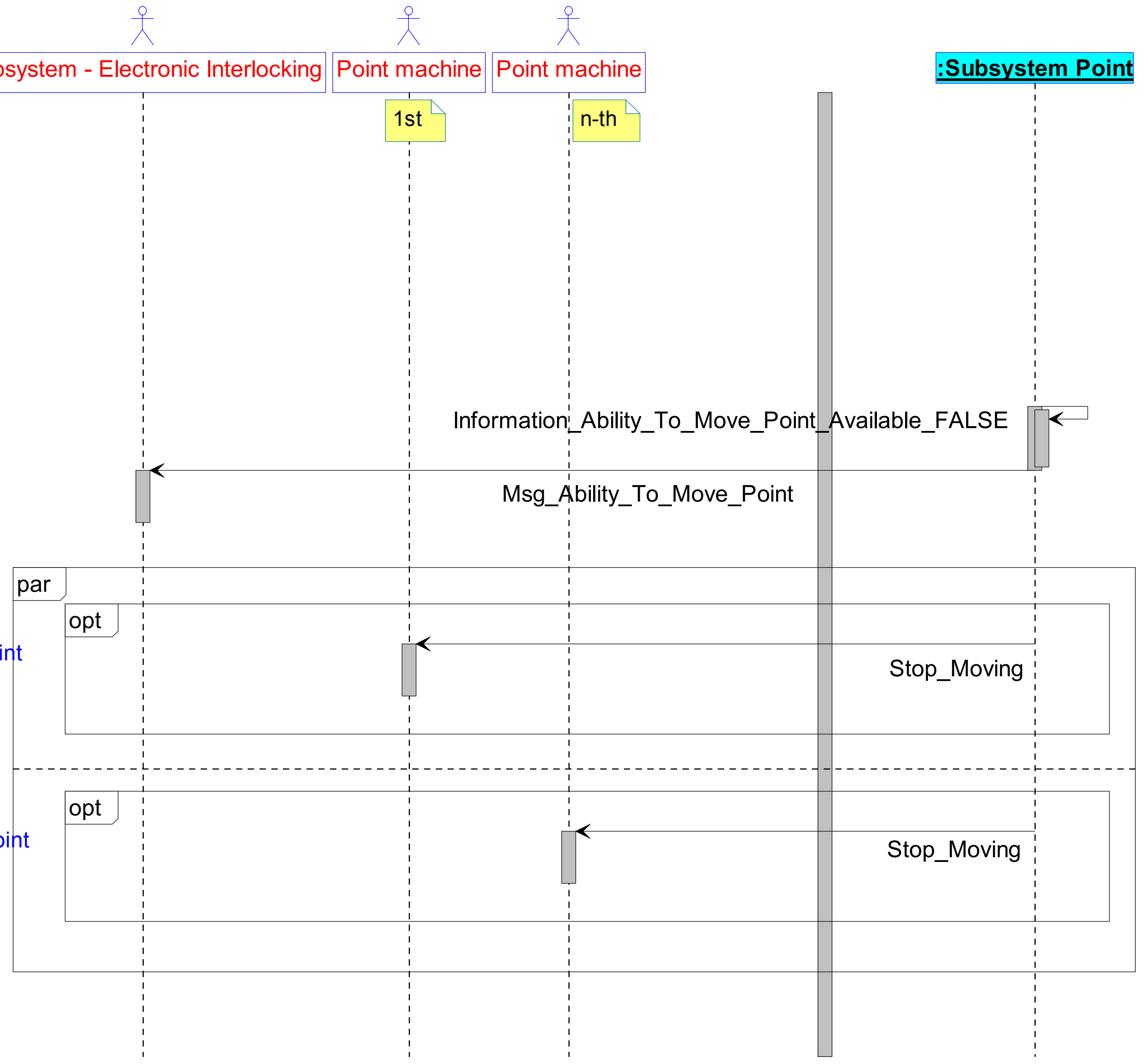
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.6749	Info	P SD 2.2.1.2.8 **P_UC2.2.1.2: Irregularities** Alternative Scenario: Handle and report Point operation timeout with position change 4W case 2 [P SD 2.2.1.2.8] **Precondition:** The Subsystem - Point is in the state OPERATIONAL. The Subsystem - Point is configured with a 4-wire interface to the Point machine. The Subsystem - Point is in: - an End position "Y", or - No end position, or - an Unintended position. **Interaction 2.2.1.2.8.A:** **1.** - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position "X". **opt** [The Subsystem Point was previously in an End position or a Unintended position] **2.a1** The Subsystem - Point sends the Information to the Subsystem - Electronic Interlocking that the Point is in No end position. **end opt** **3.** 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 the timer Con_tmax_Point_Operation. **Interaction 2.2.1.2.8.B:** **4.** - The timer Con_tmax_Point_Operation expires resulting in a Failed Movement. **5.** The Subsystem - Point sends the Command to the Point machine to stop moving the Point. **6.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that a Failed Movement has occurred. **7.** - The Subsystem - Point receives from the Point machine the Information that the Point machine is in No end position. **Postcondition:** The Subsystem - Point is in a No end position.  Subsystem - Electronic Interlocking Point machine :Subsystem Point Cd_Move_Point opt Msg_Point_Position Moving after {Con_tmax_Point_Operation} Stop_Moving Msg_Movement_Failed Information_No_End_Position	Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE.	Basic 4-wire single P

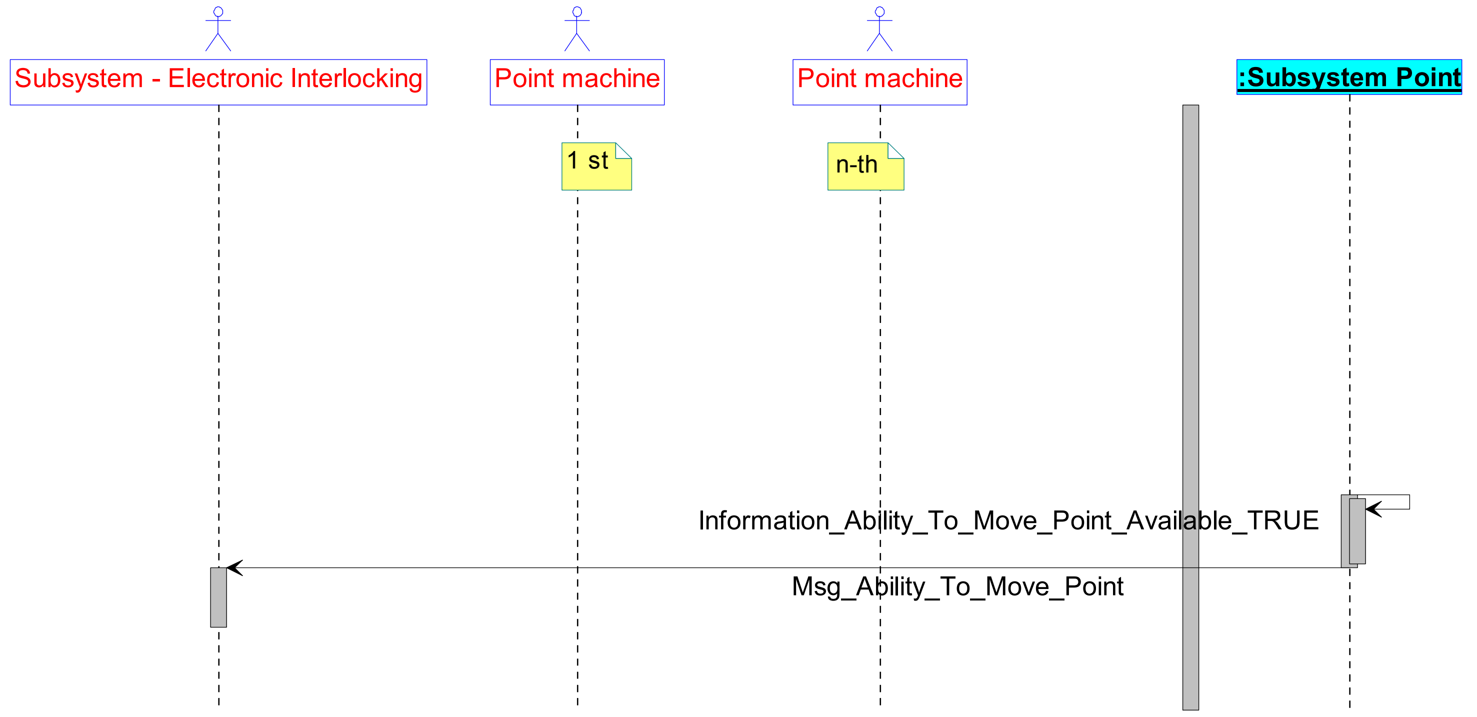
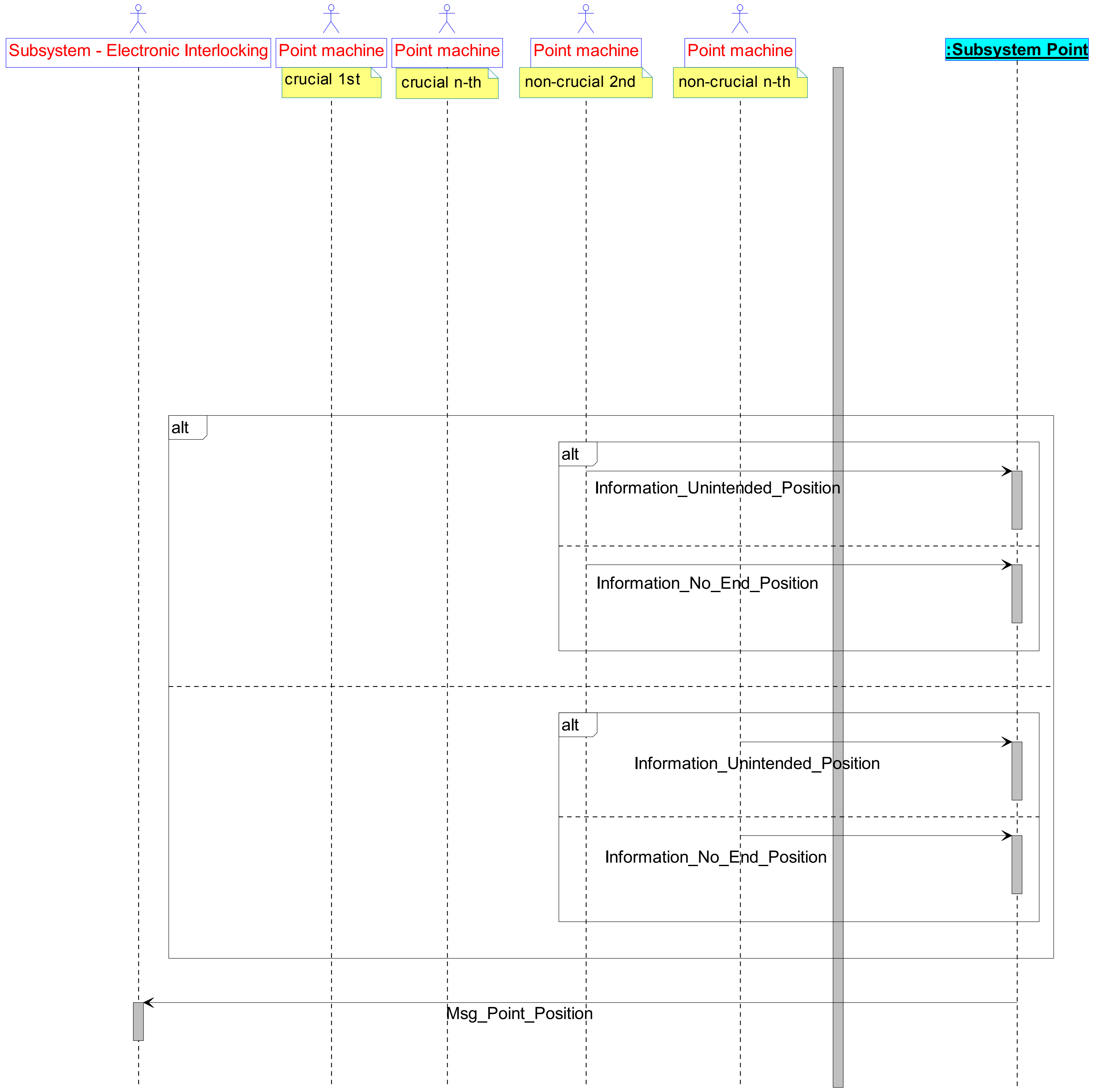
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.6743	Info	P SD 2.2.1.2.9 **P_UC2.2.1.2: Irregularities**  Subsystem - Electronic Interlocking Point machine :Subsystem Point Cd_Move_Point opt Msg_Point_Position Moving {<Con_tmax_Point_Operation} Movement_Failed Stop_Moving Msg_Movement_Failed alt Information_Unintended_Position Information_Unintended_Position Msg_Point_Position Alternative Scenario: Handle and report failed movement without position change 4W case 1 [P SD 2.2.1.2.9] **Precondition:** The Subsystem - Point is in the state OPERATIONAL. The Subsystem - Point is configured with a 4-wire interface to the Point machine. The Subsystem - Point is in: - an End position "Y"or - a Unintended position. **Interaction 2.2.1.2.9.A:** **1.** - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position "X". **opt** [The Subsystem Point was previously in an End position or a Unintended position] **2.a1** The Subsystem - Point reports to the Subsystem - Electronic Interlocking indicating that the Point is in No end position. **end opt** **3.** 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 the timer Con_tmax_Point_Operation. **Interaction 2.2.1.2.9.B:** **4.** - A failure occurred during the movement resulting in a Failed Movement. **5.** The Subsystem - Point sends the Command to the Point machine to stop moving the Point. **6.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that a Failed Movement has occurred. **alt** [The Subsystem Point was previously in an End position] **7.a1** The Subsystem - Point receives from the Point machine the pattern that represents the End position "Y", which is interpreted as Information that the Point machine is in an Unintended position. **else alt** [The Subsystem Point was previously in an Unintended position] **7.b1** The Subsystem - Point receives from the Point machine the Information that the Point machine is in a Unintended position. **end alt** **8.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in Unintended position. **Postcondition:** The Subsystem - Point is in a Unintended position.	Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE.	Basic 4-wire single P
Eu.P.6744	Info	P SD 2.2.1.2.10 **P_UC2.2.1.2: Irregularities**  Subsystem - Electronic Interlocking Point machine :Subsystem Point Cd_Move_Point Moving {<Con_tmax_Point_Operation} Movement_Failed Stop_Moving Msg_Movement_Failed Information_No_End_Position Alternative Scenario: Handle and report failed movement without position change 4W case 2 [P SD 2.2.1.2.10] **Precondition:** The Subsystem - Point is in the state OPERATIONAL. The Subsystem - Point is configured with a 4-wire interface to the Point machine. The Subsystem - Point is in No end position. **Interaction 2.2.1.2.10.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 the timer Con_tmax_Point_Operation. **Interaction 2.2.1.2.10.B:** **3.** - A failure occurred during the movement resulting in a Failed Movement. **4.** The Subsystem - Point sends the Command to the Point machine to stop moving the Point. **5.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that a Failed Movement has occurred. **6.** The Subsystem - Point receives from the Point machine the Information that the Point machine is in No end position. **Postcondition:** The Subsystem - Point is in a No end position.	Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE.	Basic 4-wire single P

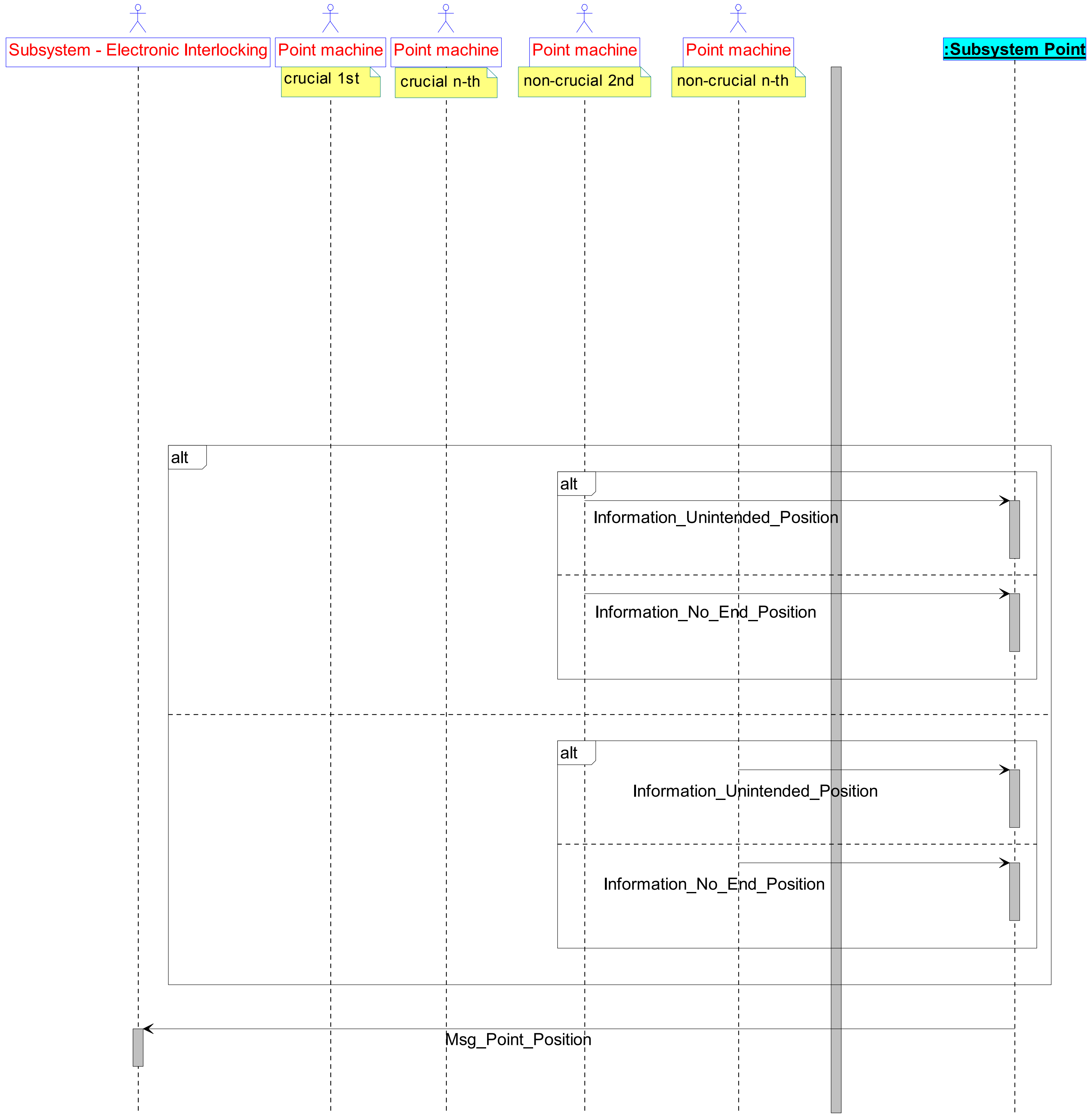
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.6769	Info	P_UC2.2.2: Multiple point machines	The Subsystem-UseCase "P_UC2.2.2: Multiple point machines" defines the behaviour of the Subsystem Point which works with multiple point machines via 4-wire interface. The behaviour will be defined in the following UseCases: P_UC2.2.2.1: Commanding and reversing P_UC2.2.2.2: Irregularities	Basic 4-wire multiple P
Eu.P.6753	Info	P_UC2.2.2.1: Commanding and reversing	The Subsystem-UseCase "P_UC2.2.2.1: Commanding and reversing" defines the behaviour of commanding and reversing a multiple point machine via 4-wire interface.	Basic 4-wire multiple P
Eu.P.6754	Info	P SD 2.2.2.1.1 **P_UC2.2.2.1: Commanding and reversing** Main Success Scenario: Moving of the Point with multiple Point machines 4W [P SD 2.2.2.1.1] **Precondition:** The Subsystem - Point is in the state OPERATIONAL. The Subsystem - Point is configured with a 4-wire interface to the Point machine. The Subsystem - Point is in: - an End position "Y", - No end position, or a Unintended position. The Subsystem - Point is configured with a 4-wire interface to the Point machine **Interaction 2.2.2.1.1.A:** **1.** - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position "X". **opt** [The Subsystem - Point was previously in an End position or a Unintended position] **2.a1** The Subsystem - Point reports the Subsystem - Electronic Interlocking that the Point is in No end position. **end opt** **par** **3.a1** - The Subsystem - Point sends the Command to the 1st Point machine to move the Point to an End position "X". **also par** **opt** [The n-th Point machine has drive capability] **3.b1.a1** - The Subsystem - Point sends the Command to the n-th Point machine to move the Point to an End position "X". **end opt** **also par** **3.c1** - The Subsystem - Point starts the timer Con_tmax_Point_Operation. **end par** **Interaction 2.2.2.1.1.B:** **par** **4.a1** - The Subsystem - Point receives from the 1st Point machine the Information that the 1st Point machine has reached End position "X" as unsafe information. **4.a2** The Subsystem - Point sends the Command to the 1st Point machine to stop moving the Point. **4.a3** - The Subsystem - Point receives the Information that the 1st Point machine is in an detected End position "X". **also par** **4.b1** - The Subsystem - Point receives from the n-th Point machine the Information that the n-th Point machine has reached End position "X" as unsafe information. **opt** [The n-th Point machine has drive capability] **4.b2.a1** The Subsystem - Point sends the Command to the n-th Point machine to stop moving the Point. **end opt** **4.b3** - The Subsystem - Point receives the Information that the n-th Point machine is in an detected End position "X". **end par** **5.** - The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in an End position "X". The timer Con_tmax_Point_Operation is reset. **Postcondition:** The Subsystem - Point is in an End position "X".  Subsystem - Electronic Interlocking Point machine Point machine :Subsystem Point 1st n-th Cd_Move_Point opt Msg_Point_Position par Moving opt Moving {< Con_tmax_Point_Operation} par Information_End_Position_Reached Stop_Moving Information_End_Position_Detected Information_End_Position_Reached opt Stop_Moving Information_End_Position_Detected Msg_Point_Position	Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE. The timing related to driving individual (1 to n-th) Point Machines for a single Point is supplier specific and is specified only in a general way. The timing related to individual Point Machines is not specified by EULYNX as part of the application layer, this shall be handled by the physical implementation.	Basic 4-wire multiple P

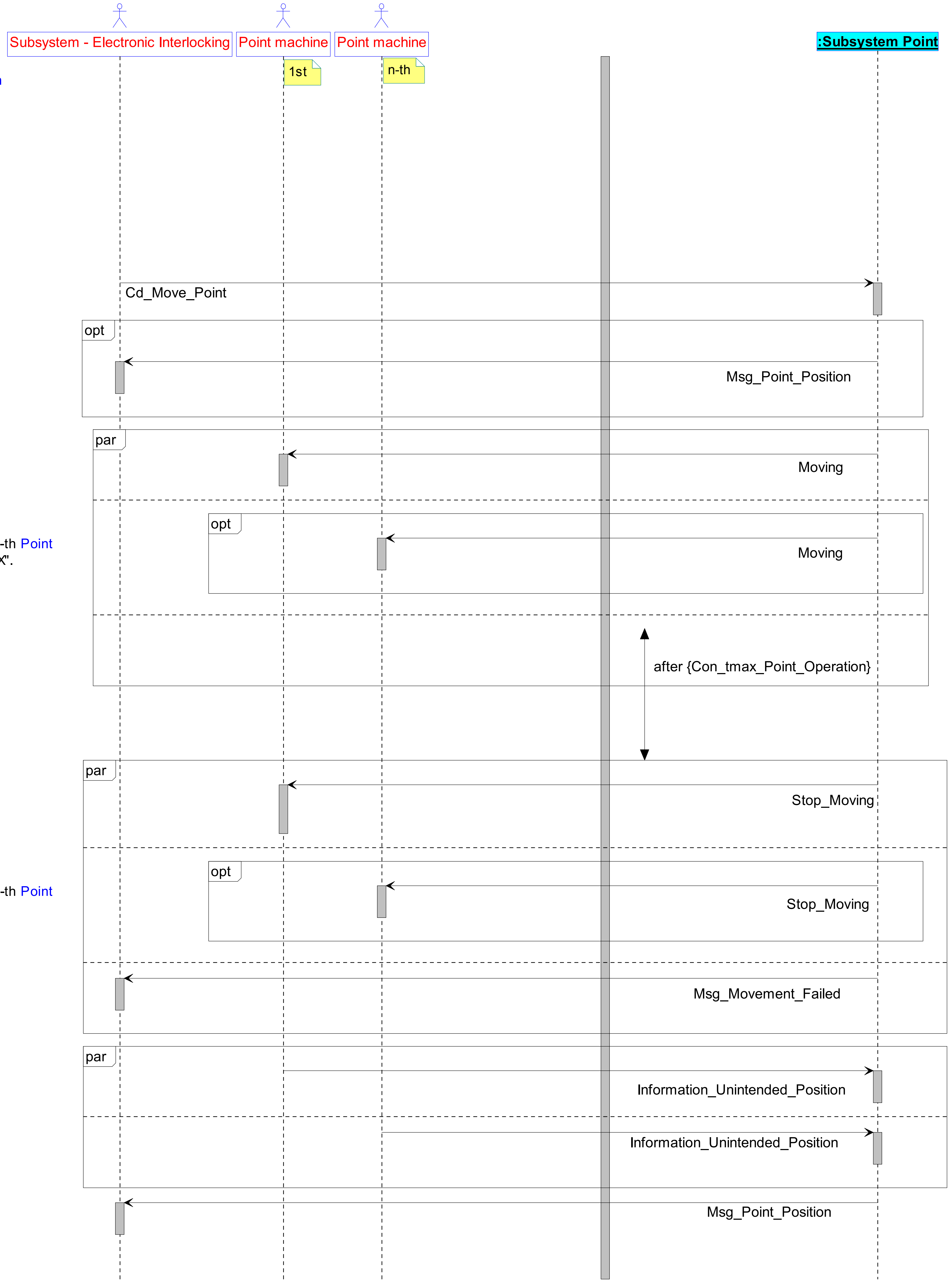
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.6755	Info	P_UC2.2.2.2: Irregularities	The Subsystem-UseCase "P_UC2.2.2.2: Irregularities" defines the behaviour of the Subsystem Point which works with a multiple point machine via 4-wire interface, when an irregularity occurs	Basic 4-wire multiple P
Eu.P.6762	Info	P SD 2.2.2.2.1 **P_UC2.2.2.2: Irregularities** Alternative Scenario: Handle and report No end position with n-th Point machines [P SD 2.2.2.2.1] **Precondition:** The Subsystem - Point is in the state OPERATIONAL. The Subsystem - Point is configured with a 4-wire interface to the Point machine AND the last commanded position is End position "Y". The Subsystem - Point is in: - an End position "Y", or- a Unintended position. **Interaction 2.2.2.2.1.A:** **par** **opt** [Any Point machine is in an End position] **alt** [The 1st Point machine is in No end position] **1.a1.a1.a1** - The Subsystem - Point receives from the 1st Point machine the Information that the 1st Point machine is in No end position. **else alt** [The n-th Point machine is in No end position] **1.a1.a1.b1** - The Subsystem - Point receives from the n-th Point machine the Information that the n-th Point machine is in No end position. **end alt** **end opt** **also par** **1.b1** - The 1st Point machine is NOT in a Unintended position. **also par** **1.c1** - The n-th Point machine is NOT in a Unintended position. **end par** **Interaction 2.2.2.2.1.B:** **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.  Subsystem - Electronic Interlocking Point machine Point machine :Subsystem Point 1st n-th par opt alt Information_No_End_Position Information_No_End_Position Msg_Point_Position		Basic 4-wire multiple P

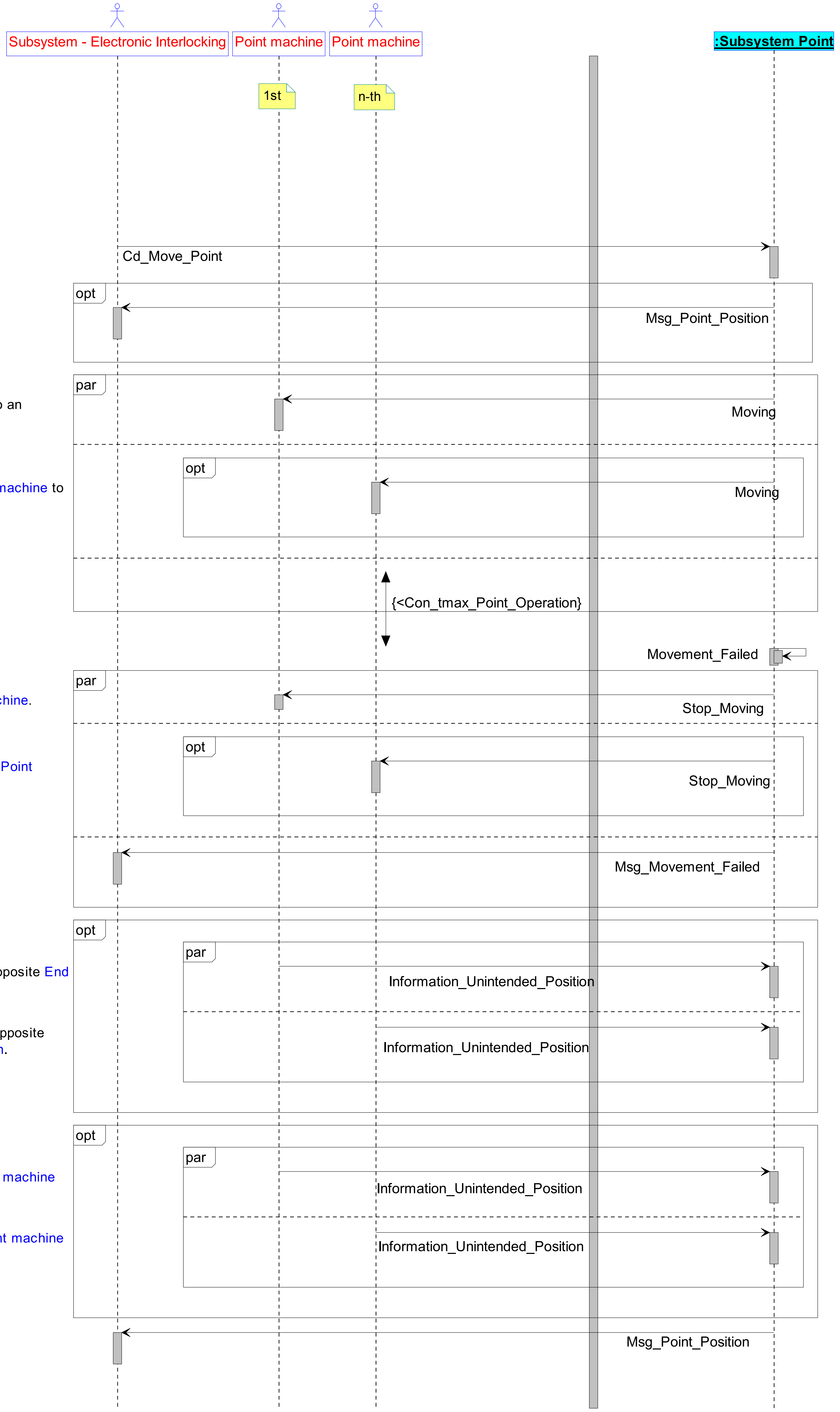
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.6768	Info	P SD 2.2.2.2.2 **P_UC2.2.2.2: Irregularities**  Subsystem - Electronic Interlocking Point machine Point machine :Subsystem Point 1 st n-th alt Information_Unintended_Position Information_Unintended_Position Msg_Point_Position Alternative Scenario: Handle and report Unintended position with n-th point machine [P SD 2.2.2.2.2] **Precondition:** The Subsystem - Point is in the state OPERATIONAL. The Subsystem - Point is configured with a 4-wire interface to the Point machine. The Subsystem - Point is in: - an End position "Y", or - No end position. **Interaction 2.2.2.2.2.A:** **alt** [The 1st Point machine is in an Unintended position.] **1.a1** - The Subsystem - Point receives from the 1st Point machine the Information that the Point machine is in a Unintended position. **else alt** [The n-th Point machine is in a Unintended position.] **1.b1** - The Subsystem - Point receives from the n-th Point machine the Information that the Point machine is in a Unintended position. **end alt** **Interaction 2.2.2.2.2.B:** **2.** - The Subsystem - Point reports to the Subsystem - Electronic Interlocking indicating that the Point is in a Unintended position. **Postcondition:** The Subsystem - Point is in a Unintended position.		Basic 4-wire multiple P
Eu.P.6758	Info	P SD 2.2.2.2.3 **P_UC2.2.2.2: Irregularities**  Subsystem - Electronic Interlocking Point machine Point machine :Subsystem Point 1 st n-th alt Information_End_Position_Detected Information_End_Position_Detected par Information_End_Position_Detected Information_End_Position_Detected Msg_Point_Position Alternative Scenario: Handle and report End Position with n-th point machine [P SD 2.2.2.2.3] **Precondition:** he Subsystem - Point is in the state OPERATIONAL. The Subsystem - Point is configured with a 4-wire interface to the Point machine AND the last commanded position is End position "Y". The Subsystem - Point is in: - an No end position, or - a Unintended position. **Interaction 2.2.2.2.3.A:** **alt** [The 1st Point machine is not in an End position "Y" and the n-th Point machine is in an End position "Y".] **1.a1** - The Subsystem - Point receives from the 1st Point machine the Information that the Point machine is in an End position "Y". **else alt** [The n-th Point machine is not in an End position "Y" and the 1st Point machine is in an End position "Y".] **1.b1** - The Subsystem - Point receives from the n-th Point machine the Information that the Point machine is in an End position "Y". **else alt** [The n-th Point machine is not in an End position "Y" and the 1st Point machine is not in an End position "Y".] **par** **1.c1.a1** - The Subsystem - Point receives from the 1st Point machine the Information that the Point machine is in an End position "Y". **also par** **1.c1.b1** - The Subsystem - Point receives from the n-th Point machine the Information that the Point machine is in an End position "Y". **end par** **end alt** **Interaction 2.2.2.2.3.B:** **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".		Basic 4-wire multiple P

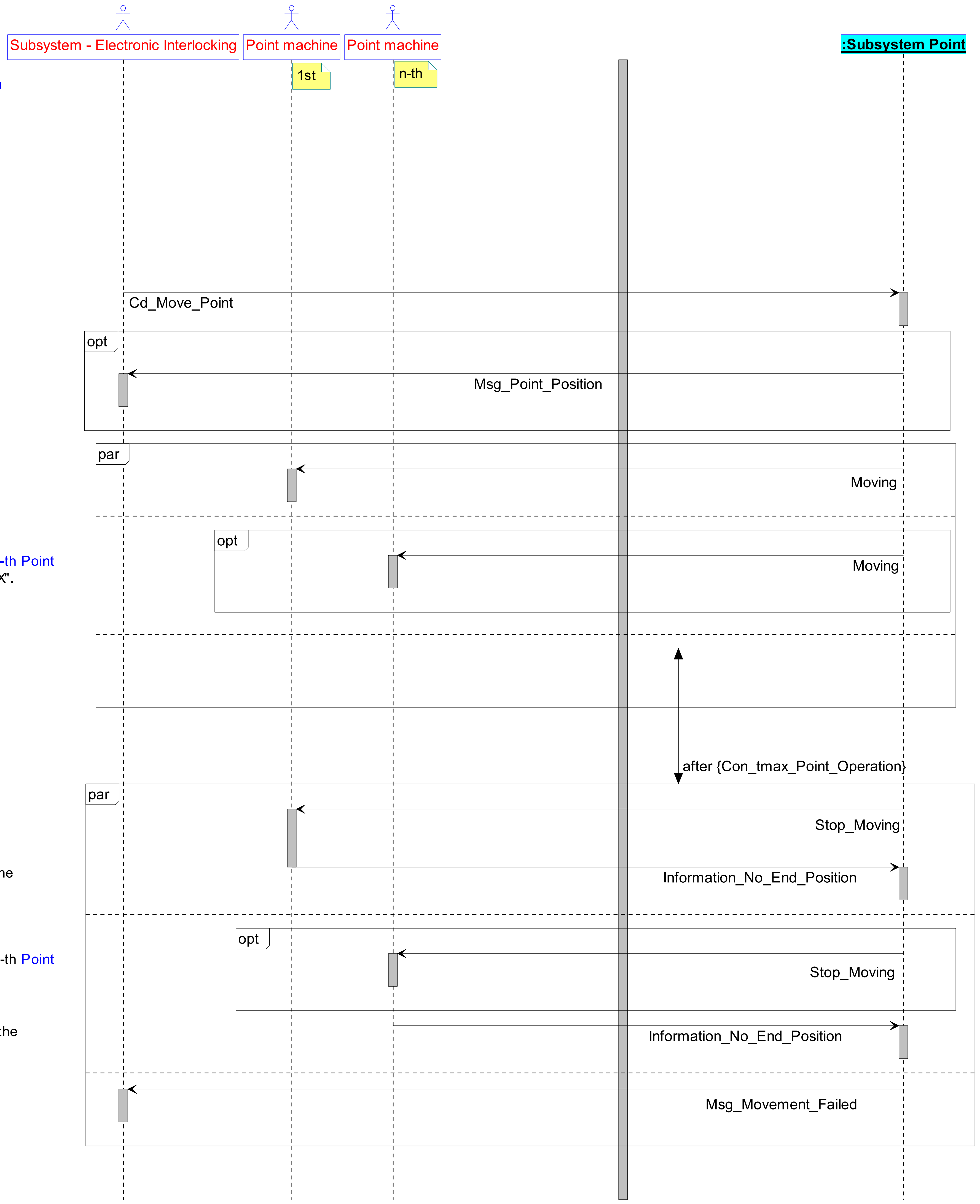
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.6764	Info	P SD 2.2.2.2.4 **P_UC2.2.2.2: Irregularities** Alternative Scenario: Handle and report opposite End Position with n-th point machine via 4-wire interface [P SD 2.2.2.2.4] **Precondition:** The Subsystem - Point is in the state OPERATIONAL. The Subsystem - Point is configured with a 4-wire interface to the Point machine AND the last commanded position is End position "Y". The Subsystem - Point is in: - an End position "Y", or - No end position. **Interaction 2.2.2.2.4.A:** **alt** [The 1st Point machine is in End position "X".] **1.a1** - The Subsystem - Point receives from the 1st Point machine the pattern that represents the opposite End position "X", which is interpreted as Information that the Point machine is in an Unintended position. **else alt** [The n-th Point machine is in End position "X".] **1.b1** - The Subsystem - Point receives from the n-th Point machine the pattern that represents the opposite End position "X", which is interpreted as Information that the Point machine is in an Unintended position. **end alt** **Interaction 2.2.2.2.4.B:** **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.  Subsystem - Electronic Interlocking Point machine Point machine :Subsystem Point 1 st n-th alt Information_Unintended_Position Information_Unintended_Position Msg_Point_Position	Note: The Subsystem - Point receives from the Point machine the pattern that represents the opposite end position "X", which is interpreted as Information that the Point machine is in an Unintended position.	Basic 4-wire multiple P
Eu.P.6761	Info	P SD 2.2.2.2.5 **P_UC2.2.2.2: Irregularities** Alternative Scenario: Handle and report loss of ability to move point with n-th point machines [P SD 2.2.2.2.5] **Precondition**: The Subsystem - Point is in the state OPERATIONAL. The Subsystem - Point is configured with a 4-wire interface to the Point machine. The Subsystem - Point is configured to observe the Ability to move point. The Subsystem - Point is Able to move point. **Interaction 2.2.2.2.6.A**: **1.** - The Subsystem - Point internal trigger indicates that the Point is Unable to move point. **2.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that it is Unable to move point. **Interaction 2.2.2.2.6.B**: **par** **opt** [The1st point machine is in Moving] **3.a1.a1** - The Subsystem - Point sends the Command to the 1st Point machine to Stop moving the 1st Point machine. **end opt** **also par** **opt** [The n-th Point machine is in Moving and has drive capability] **3.b1.a1** - The Subsystem - Point sends the Command to the n-th Point machine to Stop moving the n-th Point machine. **end opt** **end par** **Postcondition:** The Subsystem - Point is Unable to move point.  Subsystem - Electronic Interlocking Point machine Point machine :Subsystem Point 1st n-th Information_Ability_To_Move_Point_Available_FALSE Msg_Ability_To_Move_Point par opt Stop_Moving opt Stop_Moving	Only applicable if the package [Option Able to move] is used in combination with [Basic 4-wire multiple P].	Option Able to move

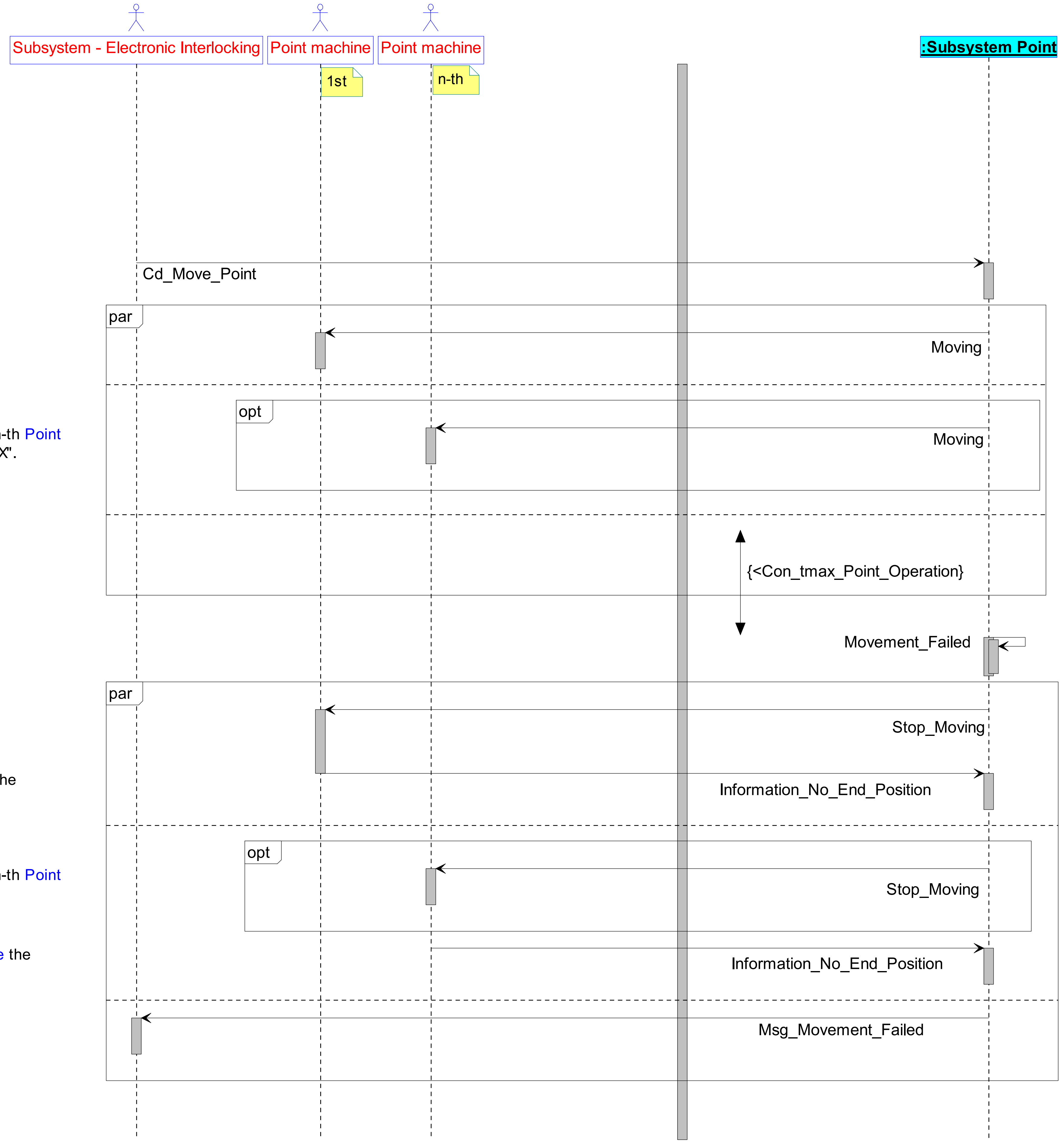
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.6767	Info	P SD 2.2.2.2.6 **P_UC2.2.2.2: Irregularities** Alternative Scenario: Handle and report restoring of Ability to move point with n-th point machine [P SD 2.2.2.2.6] **Precondition**: The Subsystem - Point is in the state OPERATIONAL. The Subsystem - Point is configured with a 4-wire interface to the Point machine. The Subsystem - Point is configured to observe the Ability to move point. The Subsystem - Point is Unable to move point. **Interaction 2.2.2.2.7.A**: **1.** - The Subsystem - Point internal trigger indicates that the Point is Able to move point. **2.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that it is Able to move point. **Postcondition:** The Subsystem - Point is Able to move point.  Subsystem - Electronic Interlocking Point machine Point machine :Subsystem Point 1 st n-th Information_Ability_To_Move_Point_Available_TRUE Msg_Ability_To_Move_Point	Only applicable if the package [Option Able to move] is used in combination with [Basic 4-wire multiple P].	Option Able to move
Eu.P.6756	Info	P SD 2.2.2.2.7 **P_UC2.2.2.2: Irregularities** Alternative Scenario: Handle and report Degraded Point Position with n-th Point machines [P SD 2.2.2.2.7] **Precondition:** The Subsystem - Point is in the state OPERATIONAL. The Subsystem - Point is in a Non degraded position. The 1st crucial Point machine is in an End position "X". The Subsystem - Point is configured with only one crucial Point machine OR the Subsystem - Point is configured with more than one crucial Point machine AND the n-th crucial Point machine is in an End position "X". The non-crucial 2nd Point machine is NOT in End position "Y". If the Subsystem - Point is configured with a non-crucial n-th Point machine, it is NOT in End position "Y". The Subsystem - Point is configured with a 4-wire interface to the Point machine AND the last commanded position is End position "X". **Interaction 2.2.2.2.8.A** **alt** [The non-crucial 2nd Point machine is not in an End position] **alt** [The non-crucial 2nd Point machine is in a Unintended position.] **1.a1.a1** - The Subsystem - Point receives from the non-crucial 2nd Point machine the Information to the that the non-crucial 2nd Point machine is in a Unintended position. **else alt** [The non-crucial 2nd Point machine is in No end position.] **1.a1.b1** - The Subsystem - Point receives from the non-crucial 2nd Point machine the Information that the non-crucial 2nd Point machine is in No end position. **end alt** **else alt** [The Subsystem - Point is configured with more than one non-crucial Point machine] **alt** [The non-crucial n-th Point machine is in a Unintended position.] **1.b1.a1** - The Subsystem - Point receives from the non-crucial n-th Point machine the Information that the non-crucial n-th Point machine is in a Unintended position. **else alt** [The non-crucial n-th Point machine is in No end position.] **1.b1.b1** - The Subsystem - Point receives from the non-crucial n-th Point machine the Information that the non-crucial n-th Point machine is in No end position. **end alt** **end alt** **Interaction 2.2.2.2.8.B** **2.** - The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in a Degraded point position "X" **Postcondition**: The Subsystem - Point is in a Degraded point position "X"  Subsystem - Electronic Interlocking Point machine Point machine Point machine Point machine :Subsystem Point crucial 1st crucial n-th non-crucial 2nd non-crucial n-th alt alt Information_Unintended_Position Information_No_End_Position alt Information_Unintended_Position Information_No_End_Position Msg_Point_Position		Basic 4-wire multiple P

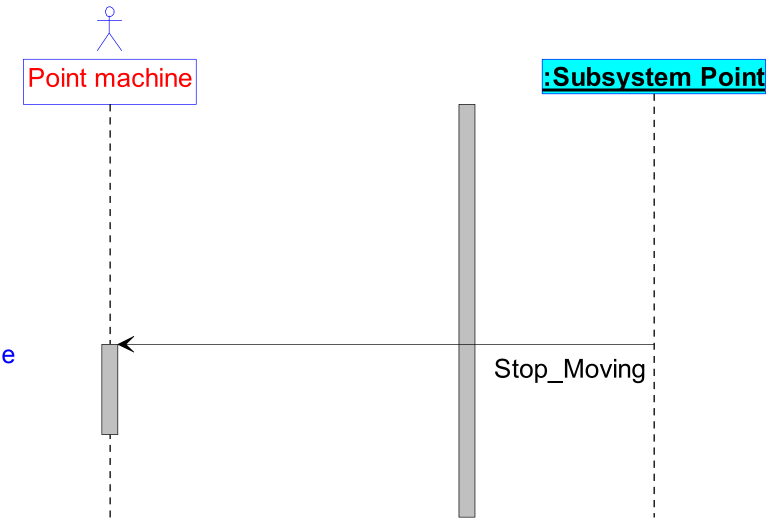
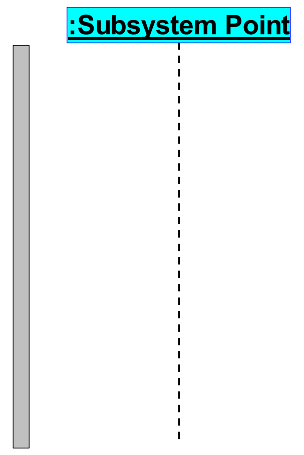
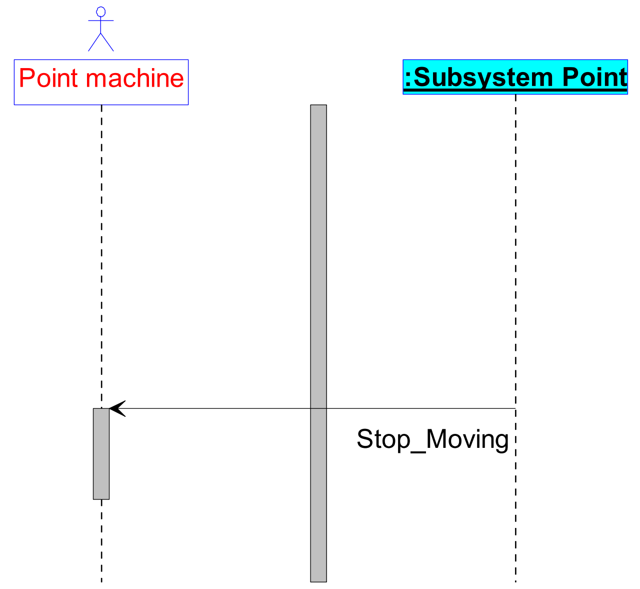
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.6763	Info	P SD 2.2.2.2.8 **P_UC2.2.2.2: Irregularities** Alternative Scenario: Handle and report non-degraded Point Position with n-th Point machines [P SD 2.2.2.2.8] **Precondition:** The Subsystem - Point is in the state OPERATIONAL. The Subsystem - Point is in a Degraded point position "X" The Subsystem - Point is configured with a 4-wire interface to the Point machine AND the last commanded position is End position "X". **Interaction 2.2.2.2.9.A** The following conditions are no longer fulfilled for either "X" or "Y". The 1st crucial Point machine is in an is End position "X". The non-crucial 2nd Point machine is NOT in End position "Y". If the Subsystem - Point is configured with a non-crucial n-th Point machine, it is NOT in End position "Y". The Subsystem - Point is configured with only one crucial Point machine OR the Subsystem - Point is configured with more than one crucial Point machine AND the n-th crucial Point machine is in an is End position "X". **alt** [The non-crucial 2nd Point machine is not in an End position] **alt** [The non-crucial 2nd Point machine is in a Unintended position.] **1.a1.a1** - The Subsystem - Point receives from the non-crucial 2nd Point machine the Information that the non-crucial 2nd Point machine is in a Unintended position. **else alt** [The non-crucial 2nd Point machine is in No end position.] **1.a1.b1** - The Subsystem - Point receives from the non-crucial 2nd Point machine the Information that the non-crucial 2nd Point machine is in No end position. **end alt** **else alt** [The Subsystem - Point is configured with more than one non-crucial Point machine] **alt** [The non-crucial n-th Point machine is in a Unintended position.] **1.b1.a1** - The Subsystem - Point receives from the non-crucial n-th Point machine the Information that the non-crucial n-th Point machine is in a Unintended position. **else alt** [The non-crucial n-th Point machine is in No end position.] **1.b1.b1** - The Subsystem - Point receives from the non-crucial n-th Point machine the Information that the non-crucial n-th Point machine is in No end position. **end alt** **end alt** **Interaction 2.2.2.2.9.B** **2.** - The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is not in a Degraded point position "X" **Postcondition**: The Subsystem - Point is in a Non degraded position.  Subsystem - Electronic Interlocking Point machine crucial 1st Point machine crucial n-th Point machine non-crucial 2nd Point machine non-crucial n-th :Subsystem Point alt alt Information_Unintended_Position Information_No_End_Position alt Information_Unintended_Position Information_No_End_Position Msg_Point_Position		Basic 4-wire multiple P

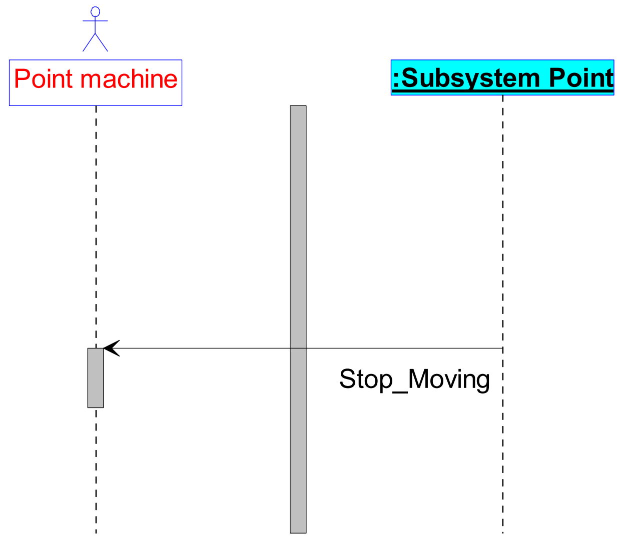
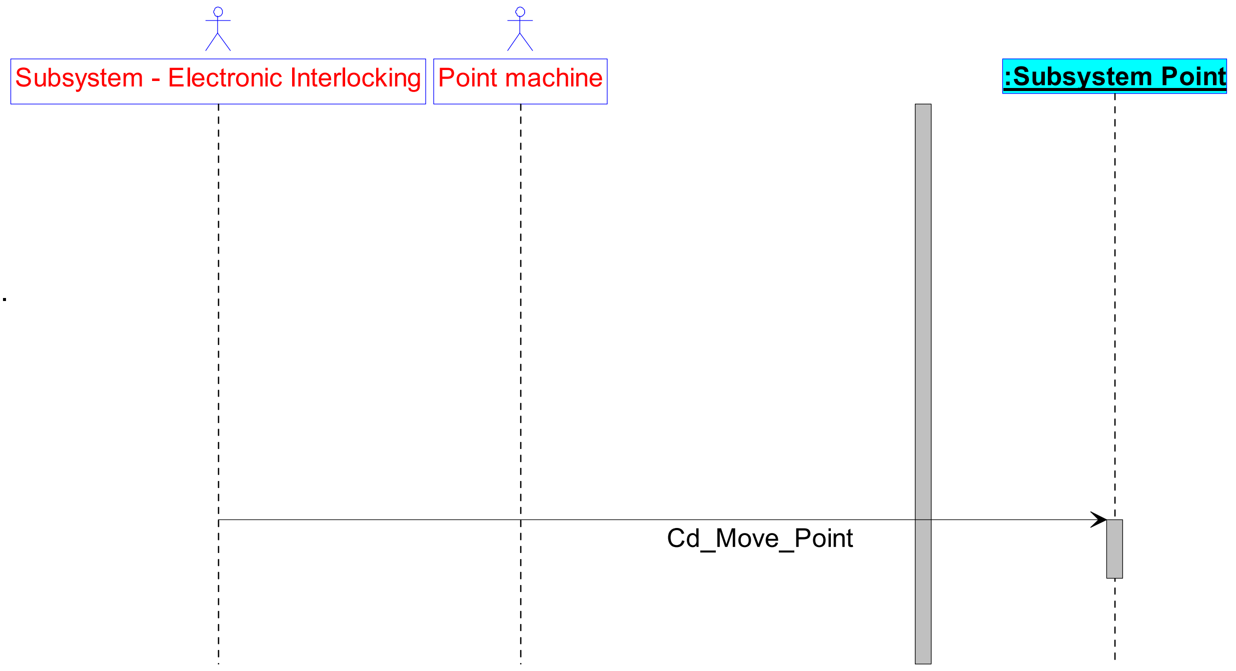
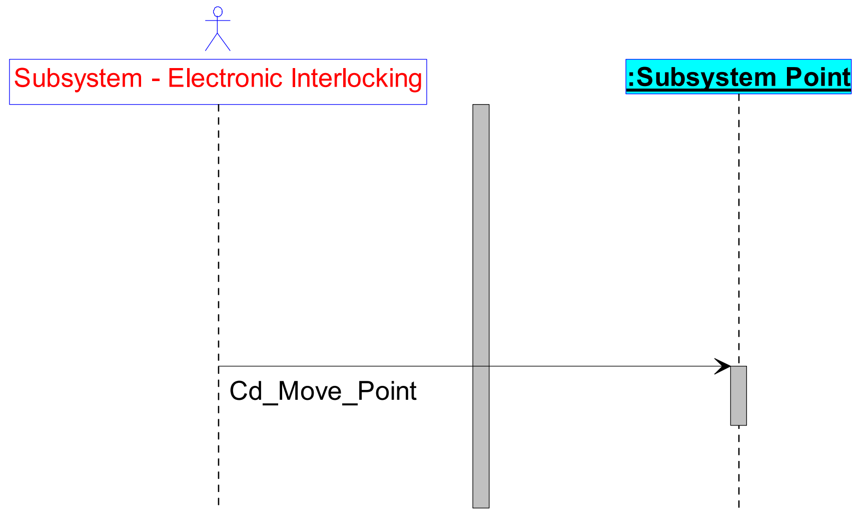
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.6765	Info	P SD 2.2.2.2.9 **P_UC2.2.2.2: Irregularities**  Subsystem - Electronic Interlocking Point machine Point machine :Subsystem Point 1st n-th Cd_Move_Point opt Msg_Point_Position par Moving opt Moving after {Con_tmax_Point_Operation} par Stop_Moving opt Stop_Moving Msg_Movement_Failed par Information_Unintended_Position Information_Unintended_Position Msg_Point_Position Alternative Scenario: Handle and report Point operation timeout with n-th Point machines with position change case 1 [P SD 2.2.2.2.9] **Precondition:** The Subsystem - Point is in the state OPERATIONAL. The Subsystem - Point is configured with a 4-wire interface to the Point machine. The Subsystem - Point is in: - an End position "Y", - No end position, or - a Unintended position. **Interaction 2.2.2.2.10.A:** **1.** - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position "X". **opt** [The Subsystem - Point was previously in an End position or a Unintended position] **2.a1** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in No end position. **end opt** **par** **3.a1** The Subsystem - Point sends the Command to the 1st Point machine to move the 1st Point machine to an End position "X". **also par** **opt** [The n-th Point machine has drive capability] **3.b1.a1** The Subsystem - Point sends the Command to the n-th Point machine to move the n-th Point machine to an End position "X". **end opt** **also par** **3.c1** The Subsystem - Point starts the Timer for moving Con_tmax_Point_Operation. **end par** **Interaction 2.2.2.2.10.B:** **4.** - The timer Con_tmax_Point_Operation expires resulting in a Failed Movement. **par** **5.a1** The Subsystem - Point sends the Command to the 1st Point machine to stop moving the 1st Point machine. The timer Con_tmax_Point_Operation is reset. **also par** **opt** [The n-th Point machine has drive capability] **5.b1.a1** The Subsystem - Point sends the Command to the n-th Point machine to stop moving the n-th Point machine. **end opt** **also par** **5.c1** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that a Failed Movement has occurred. **end par** **par** **6.a1** - The Subsystem - Point receives from the Point machine the Information that the Point machine is in a Unintended position. **also par** **6.b1** - The Subsystem - Point receives from the Point machine the Information that the Point machine is in a Unintended position. **end par** **7.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in Unintended position. **Postcondition:** The Subsystem - Point is in Unintended position.	Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE. The timing related to driving individual (1 to n-th) Point Machines for a single Point is supplier specific and is specified only in a general way. The timing related to individual Point Machines is not specified by EULYNX as part of the application layer, this shall be handled by the physical implementation.	Basic 4-wire multiple P

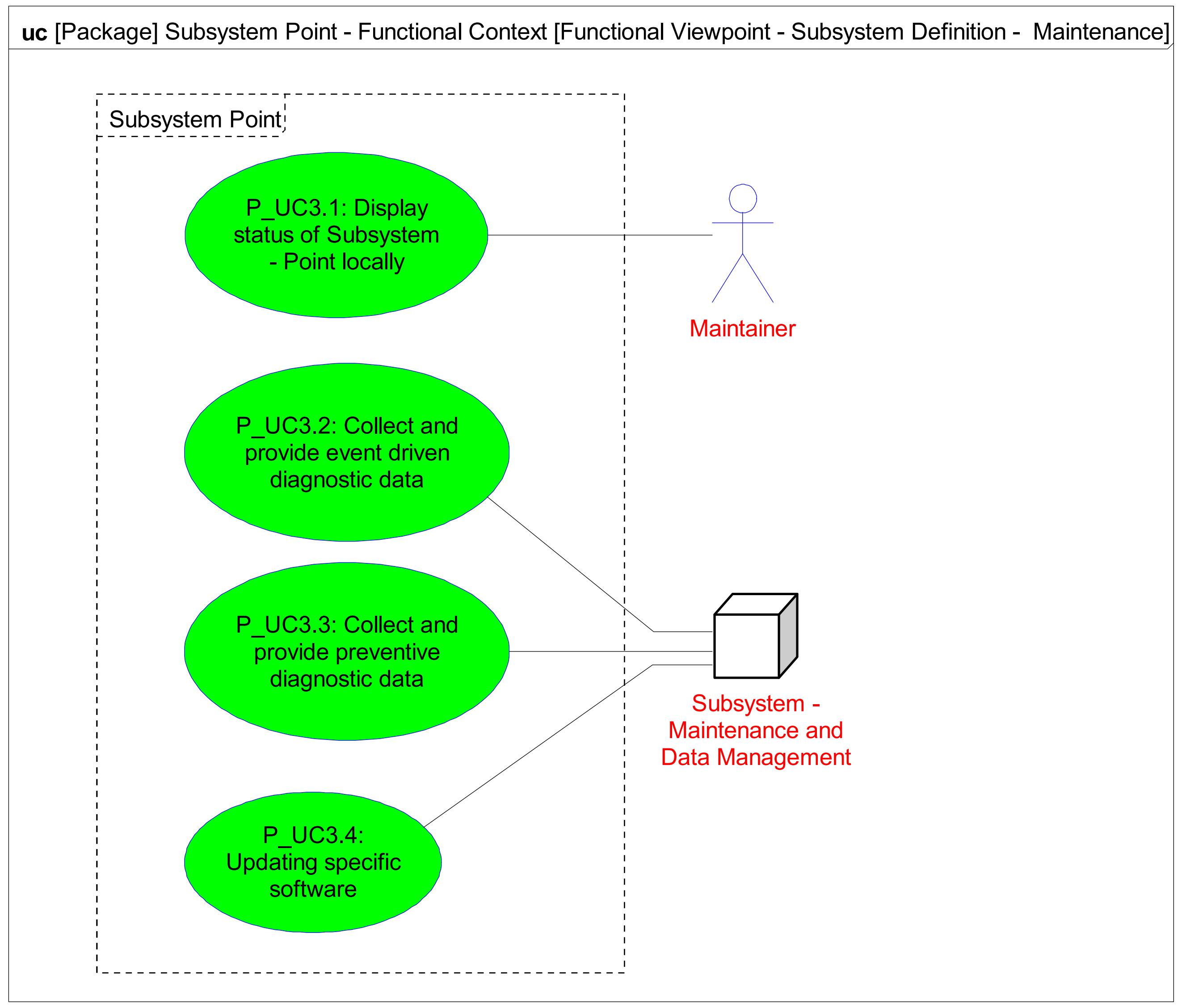
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.6759	Info	P SD 2.2.2.2.10 **P_UC2.2.2.2: Irregularities** Alternative Scenario: Handle and report failed movement with n-th Point machines without position change case 1 [P SD 2.2.2.2.10] **Precondition:** The Subsystem - Point is in the state OPERATIONAL. The Subsystem - Point is configured with a 4-wire interface to the Point machine. The Subsystem - Point is in: - an End position "Y", or - an Unintended position. **Interaction 2.2.2.2.11.A:** **1.** - The Subsystem - Electronic Interlocking sends a Command to the Subsystem - Point to move the Point to an End position "X". **opt** [The Subsystem - Point was previously in an End position or a Unintended position] **2.a1** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in No end position. **end opt** **par** **3.a1** The Subsystem - Point sends the Command to the 1st Point machine to move the 1st Point machine to an End position "X". **also par** **opt** [The n-th Point machine has drive capability] **3.b1.a1** The Subsystem - Point sends the Command to the n-th Point machine to move the n-th Point machine to an End position "X". **end opt** **also par** **3.c1** The Subsystem - Point starts the Timer for moving Con_tmax_Point_Operation. **end par** **Interaction 2.2.2.2.11.B:** **4.** - A failure occurred during the movement resulting in a Failed Movement. **par** **5.a1** The Subsystem - Point sends the Command to the 1st Point machine to stop moving the 1st Point machine. **also par** **opt** [The n-th Point machine has drive capability] **5.b1.a1** The Subsystem - Point sends the Command to the n-th Point machine to stop moving the n-th Point machine. **end opt** **also par** **5.c1** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that a Failed Movement has occurred. **end par** **opt** [The Subsystem - Point was in an End position "Y".] **par** **6.a1.a1** - The Subsystem - Point receives from the 1st Point machine the pattern that represents the opposite End position "Y", which is interpreted as Information that the Point machine is in an Unintended position. **also par** **6.b1.a1** - The Subsystem - Point receives from the n-th Point machine the pattern that represents the opposite End position "Y", which is interpreted as Information that the Point machine is in an Unintended position. **end par** **end opt** **opt** [The Subsystem - Point was in a Unintended position.] **par** **7.a1.a1** - The Subsystem - Point receives from the 1st Point machine the Information that the 1st Point machine is in a Unintended position. **also par** **7.b1.a1** - The Subsystem - Point receives from the n-th Point machine the Information that the n-th Point machine is in a Unintended position. **end par** **end opt** **8.** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in Unintended position. **Postcondition:** The Subsystem - Point is in Unintended position.  Subsystem - Electronic Interlocking Point machine Point machine :Subsystem Point 1st n-th Cd_Move_Point opt Msg_Point_Position par Moving opt Moving {<Con_tmax_Point_Operation} Movement_Failed par Stop_Moving opt Stop_Moving Msg_Movement_Failed opt par Information_Unintended_Position Information_Unintended_Position opt par Information_Unintended_Position Information_Unintended_Position Msg_Point_Position	Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE. The timing related to driving individual (1 to n-th) Point Machines for a single Point is supplier specific and is specified only in a general way. The timing related to individual Point Machines is not specified by EULYNX as part of the application layer, this shall be handled by the physical implementation.	Basic 4-wire multiple P

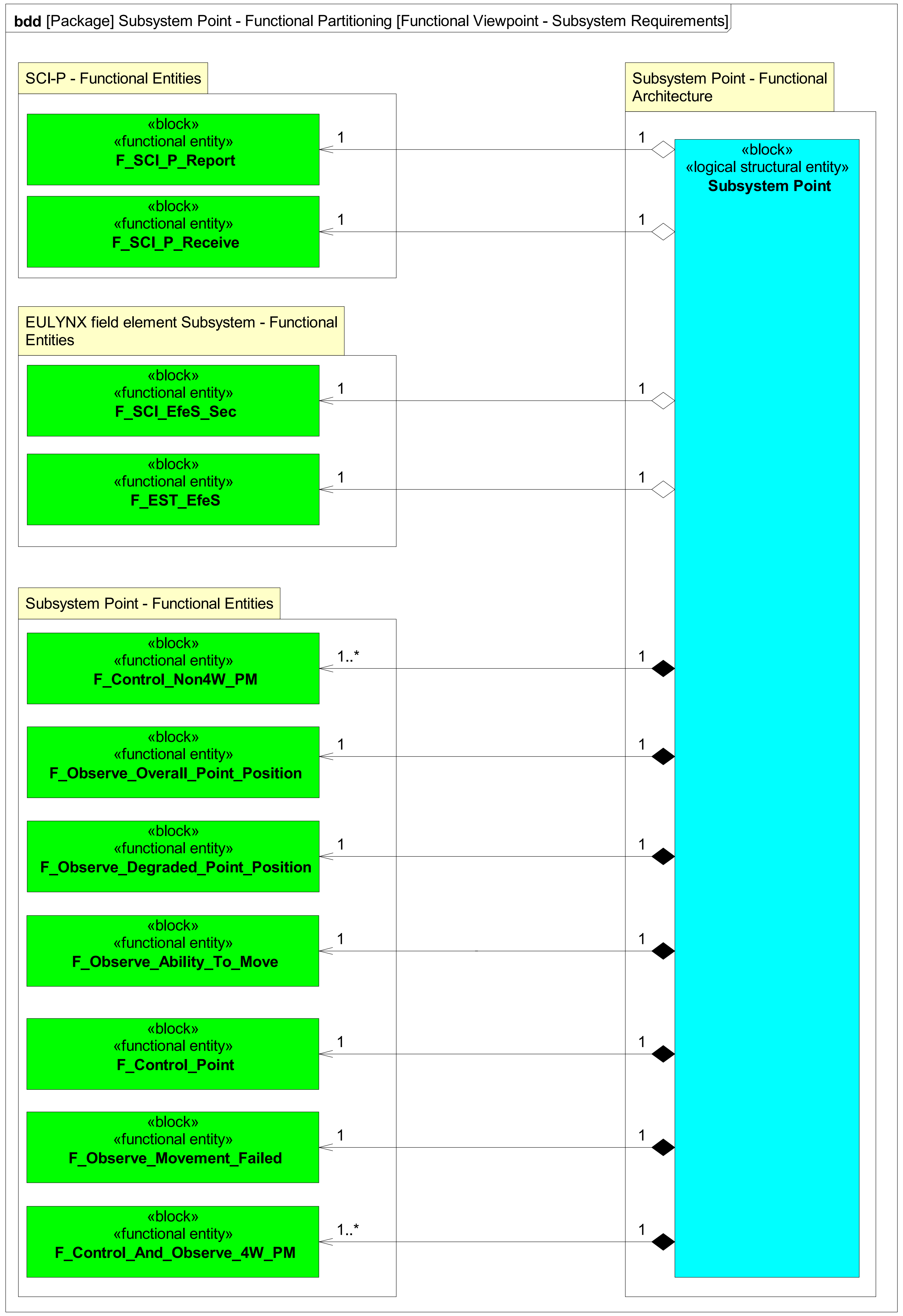
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.6766	Info	P SD 2.2.2.2.11 **P_UC2.2.2.2: Irregularities**  Subsystem - Electronic Interlocking Point machine Point machine :Subsystem Point 1st n-th Cd_Move_Point opt Msg_Point_Position par Moving opt Moving after {Con_tmax_Point_Operation} par Stop_Moving Information_No_End_Position opt Stop_Moving Information_No_End_Position Msg_Movement_Failed Alternative Scenario: Handle and report Point operation timeout with n-th Point machines without position change case 2 [P SD 2.2.2.2.11] **Precondition:** The Subsystem - Point is in the state OPERATIONAL. The Subsystem - Point is configured with a 4-wire interface to the Point machine. The Subsystem - Point is in: - an End position "Y", or - No end position, or - an Unintended position. **Interaction 2.2.2.2.12.A:** **1.** - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position "X". **opt** [The Subsystem - Point was previously in an End position or a Unintended position] **2.a1** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that the Point is in No end position. **end opt** **par** **3.a1** The Subsystem - Point sends the Command to the 1st Point machine to move the 1st Point machine to an End position "X". **also par** **opt** [The n-th Point machine has drive capability] **3.b1.a1** The Subsystem - Point sends the Command to the n-th Point machine to move the n-th Point machine to an End position "X". **end opt** **also par** **3.c1** The Subsystem - Point starts the Timer for moving Con_tmax_Point_Operation. **end par** **Interaction 2.2.2.2.12.B:** **4.** - The timer Con_tmax_Point_Operation expires resulting in a Failed Movement. **par** **5.a1** The Subsystem - Point sends the Command to the 1st Point machine to stop moving the 1st Point machine. The timer Con_tmax_Point_Operation is reset. **5.a2** The Subsystem - Point receives from the 1st Point machine the Information that the 1st Point machine is in No end position. **also par** **opt** [The n-th Point machine has drive capability] **5.b1.a1** The Subsystem - Point sends the Command to the n-th Point machine to stop moving the n-th Point machine. **end opt** **5.b2** The Subsystem - Point receives from the n-th Point machine the Information that the n-th Point machine is in No end position. **also par** **5.c1** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that a Failed Movement has occurred. **end par** **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. The timing related to driving individual (1 to n-th) Point Machines for a single Point is supplier specific and is specified only in a general way. The timing related to individual Point Machines is not specified by EULYNX as part of the application layer, this shall be handled by the physical implementation.	Basic 4-wire multiple P

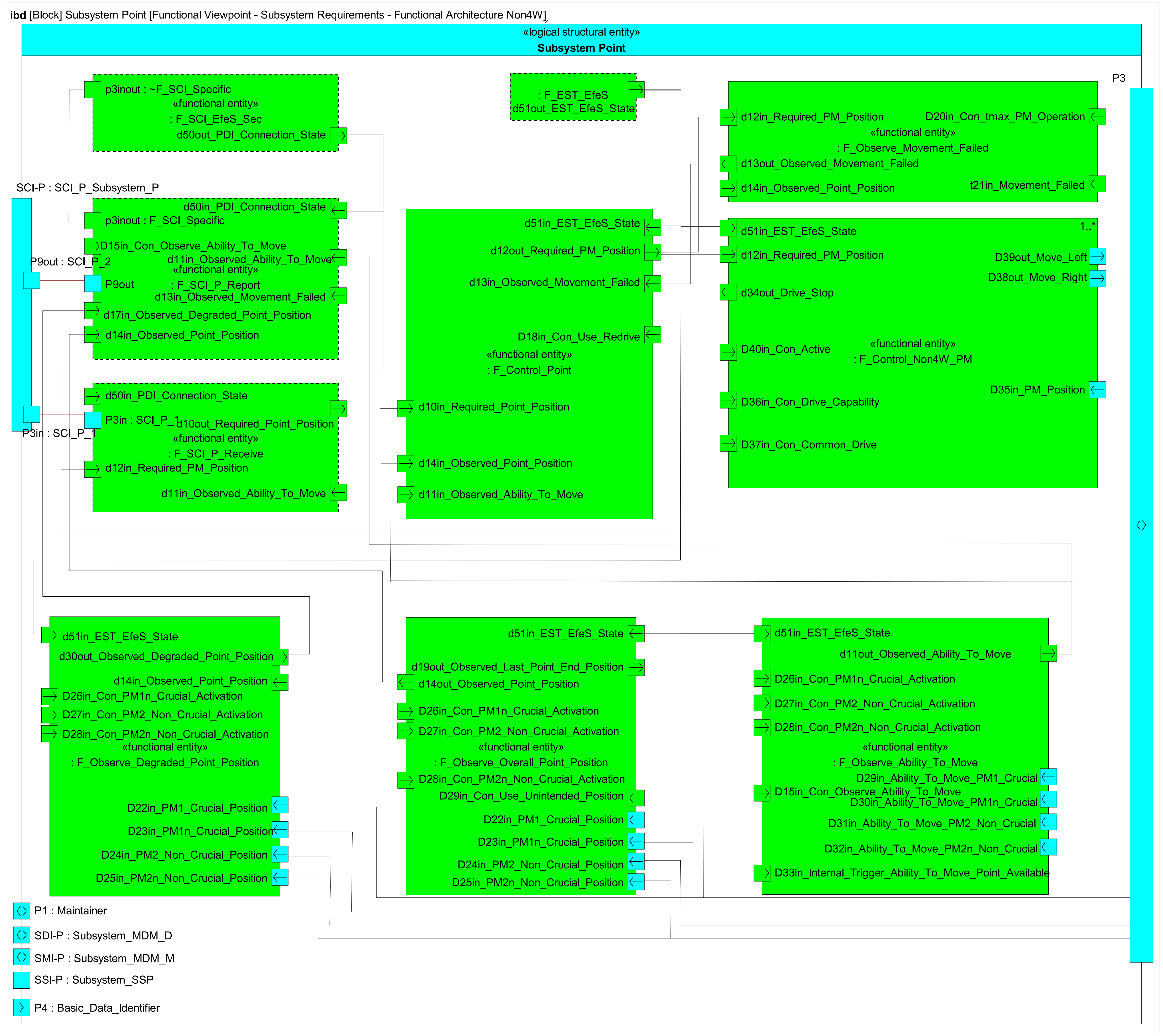
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.6760	Info	P SD 2.2.2.2.12 **P_UC2.2.2.2: Irregularities** Alternative Scenario: Handle and report failed movement with n-th Point machines without position change case 2 [P SD 2.2.2.2.12] **Precondition:** The Subsystem - Point is in the state OPERATIONAL. The Subsystem - Point is configured with a 4-wire interface to the Point machine. The Subsystem - Point is in No end position. **Interaction 2.2.2.2.13.A:** **1.** - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position "X". **par** **2.a1** The Subsystem - Point sends the Command to the 1st Point machine to move the 1st Point machine to an End position "X". **also par** **opt** [The n-th Point machine has drive capability] **2.b1.a1** The Subsystem - Point sends the Command to the n-th Point machine to move the n-th Point machine to an End position "X". **end opt** **also par** **2.c1** The Subsystem - Point starts the Timer for moving Con_tmax_Point_Operation. **end par** **Interaction 2.2.2.2.13.B:** **3.** - A failure occurred during the movement resulting in a Failed Movement. **par** **4.a1** The Subsystem - Point sends the Command to the 1st Point machine to stop moving the 1st Point machine. The timer Con_tmax_Point_Operation is reset. **4.a2** The Subsystem - Point receives from the 1st Point machine the Information that the 1st Point machine is in No end position. **also par** **opt** [The n-th Point machine has drive capability] **4.b1.a1** The Subsystem - Point sends the Command to the n-th Point machine to stop moving the n-th Point machine. **end opt** **4.b2** - The Subsystem - Point receives from the n-th Point machine the Information that the n-th Point machine is in No end position. **also par** **4.c1** The Subsystem - Point reports to the Subsystem - Electronic Interlocking that a Failed Movement has occurred. **end par** **Postcondition:** The Subsystem - Point is in No end position.  Subsystem - Electronic Interlocking Point machine Point machine :Subsystem Point 1st n-th Cd_Move_Point par Moving opt Moving {<Con_tmax_Point_Operation} Movement_Failed par Stop_Moving Information_No_End_Position opt Stop_Moving Information_No_End_Position Msg_Movement_Failed	Stop_Moving is functionally realised by setting the Moving commands for left and right to FALSE. The timing related to driving individual (1 to n-th) Point Machines for a single Point is supplier specific and is specified only in a general way. The timing related to individual Point Machines is not specified by EULYNX as part of the application layer, this shall be handled by the physical implementation.	Basic 4-wire multiple P
Eu.P.1251	Info	P_UC2.3: Handle general Irregularities	The Subsystem-UseCase P_UC2.3: Handle general Irregularities defines the behaviour of the Subsystem - Point when an irregularity occurs.	Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.1252	Info	P SD 2.3.1 **P_UC2.3: Handle general Irregularities** Alternative Scenario: Perform fallback operation [P SD 2.3.1] **Precondition:** --- **Interaction 2.3.1.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 timer Con_tmax_Point_Operation is reset. **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.	Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.1299	Info	P SD 2.3.2 **P_UC2.3: Handle general Irregularities** Alternative Scenario: Handling of interrupted PDI connection [P SD 2.3.2] **Precondition:** The Subsystem - Point is in the state OPERATIONAL. **Interaction 2.3.2.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.	Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.3093	Info	P SD 2.3.3 **P_UC2.3: Handle general Irregularities** Alternative Scenario: Supply voltage of the Subsystem has gone outside of the required range for operation [P SD 2.3.3] **Precondition:** --- **Interaction 2.3.3.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 timer Con_tmax_Point_Operation is reset. **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.	Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P

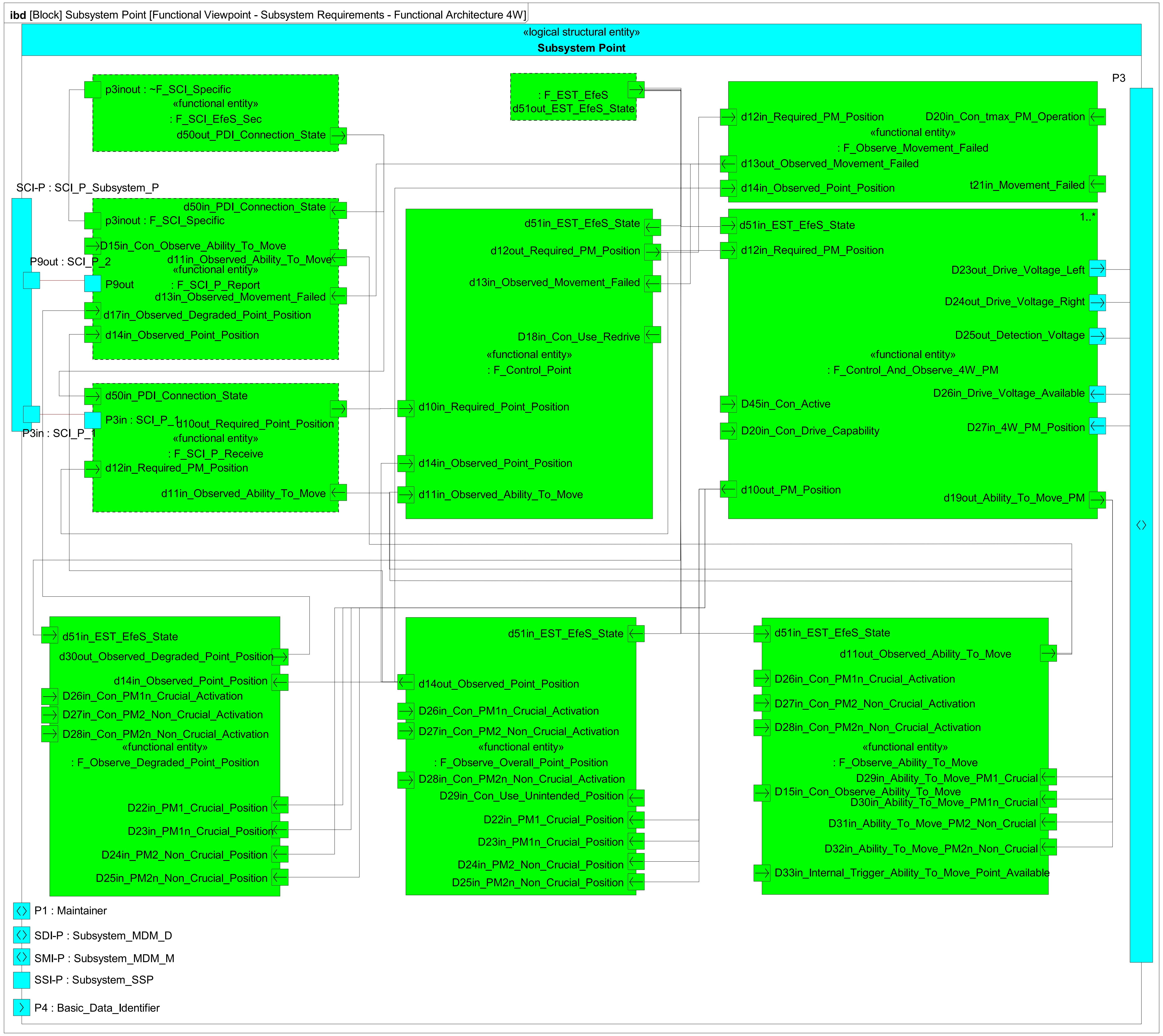
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.3104	Info	P SD 2.3.4 **P_UC2.3: Handle general Irregularities**  Alternative Scenario: Reset occurs [P SD 2.3.4] **Precondition:** The Subsystem Point is in the state INITIALISING or OPERATIONAL. **Interaction 2.3.4.A:** **1.** - A reset has occurred. **2.** The Subsystem - Point sends the Command to the Point machine to stop moving the Point. The timer Con_tmax_Point_Operation is reset. **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.	Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.5371	Info	P SD 2.3.5 **P_UC2.3: Handle general Irregularities**  Subsystem - Electronic Interlocking Point machine :Subsystem Point Cd_Move_Point Alternative Scenario: Handle received move command if ability to move point is lost [P SD 2.3.5] **Precondition:** The Subsystem - Point is in the state OPERATIONAL. The Subsystem - Point is configured to observe the Ability to move point. The Subsystem - Point is Unable to move point. The Subsystem - Point is in: - an End position, or - No end position, or - an Unintended position. **Interaction 2.3.5.A:** **1.** - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position. **Postcondition:** ---		Option Able to move
Eu.P.1469	Info	P SD 2.3.6 **P_UC2.3: Handle general Irregularities**  Subsystem - Electronic Interlocking :Subsystem Point Cd_Move_Point Alternative Scenario: Commanding of the Point to the current End position [P SD 2.3.6] **Precondition:** The Subsystem - Point is in the state OPERATIONAL. The Subsystem - Point is in an End position "Y". **Interaction 2.3.6.A:** **1.** - The Subsystem - Point receives from the Subsystem - Electronic Interlocking the Command to move the Point to an End position "Y". **Postcondition:** ---		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.1127	Info	[Package] Subsystem Point - Functional Context [Functional Viewpoint - Subsystem Definition - Maintenance]  uc [Package] Subsystem Point - Functional Context [Functional Viewpoint - Subsystem Definition - Maintenance] Subsystem Point P_UC3.1: Display status of Subsystem - Point locally P_UC3.2: Collect and provide event driven diagnostic data P_UC3.3: Collect and provide preventive diagnostic data P_UC3.4: Updating specific software Maintainer Subsystem - Maintenance and Data Management		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.1124	Req	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.	Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.1125	Req	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.	Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.1126	Req	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.	Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.1468	Req	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.	Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6272	Head	**3.3.3 Subsystem Point - Functional Partitioning**		

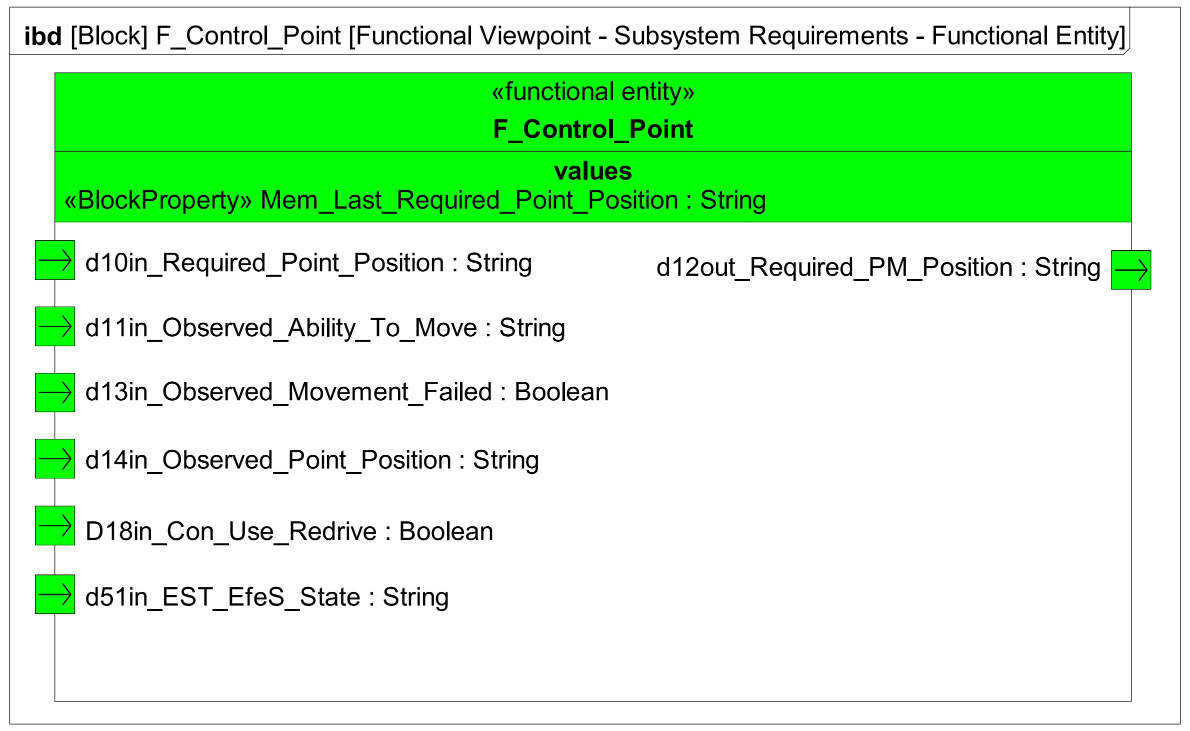
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.3297	Info	[Package] Subsystem Point - Functional Partitioning [Functional Viewpoint - Subsystem Requirements]  bdd [Package] Subsystem Point - Functional Partitioning [Functional Viewpoint - Subsystem Requirements] SCI-P - Functional Entities «block» «functional entity» F_SCI_P_Report «block» «functional entity» F_SCI_P_Receive EULYNX field element Subsystem - Functional Entities «block» «functional entity» F_SCI_EfeS_Sec «block» «functional entity» F_EST_EfeS Subsystem Point - Functional Entities «block» «functional entity» F_Control_Non4W_PM «block» «functional entity» F_Observe_Overall_Point_Position «block» «functional entity» F_Observe_Degraded_Point_Position «block» «functional entity» F_Observe_Ability_To_Move «block» «functional entity» F_Control_Point «block» «functional entity» F_Observe_Movement_Failed «block» «functional entity» F_Control_And_Observe_4W_PM Subsystem Point - Functional Architecture «block» «logical structural entity» Subsystem Point 1 1..*		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.5781	Head	**3.3.4 Subsystem Point - Functional Architecture**		
Eu.P.939	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.	Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P

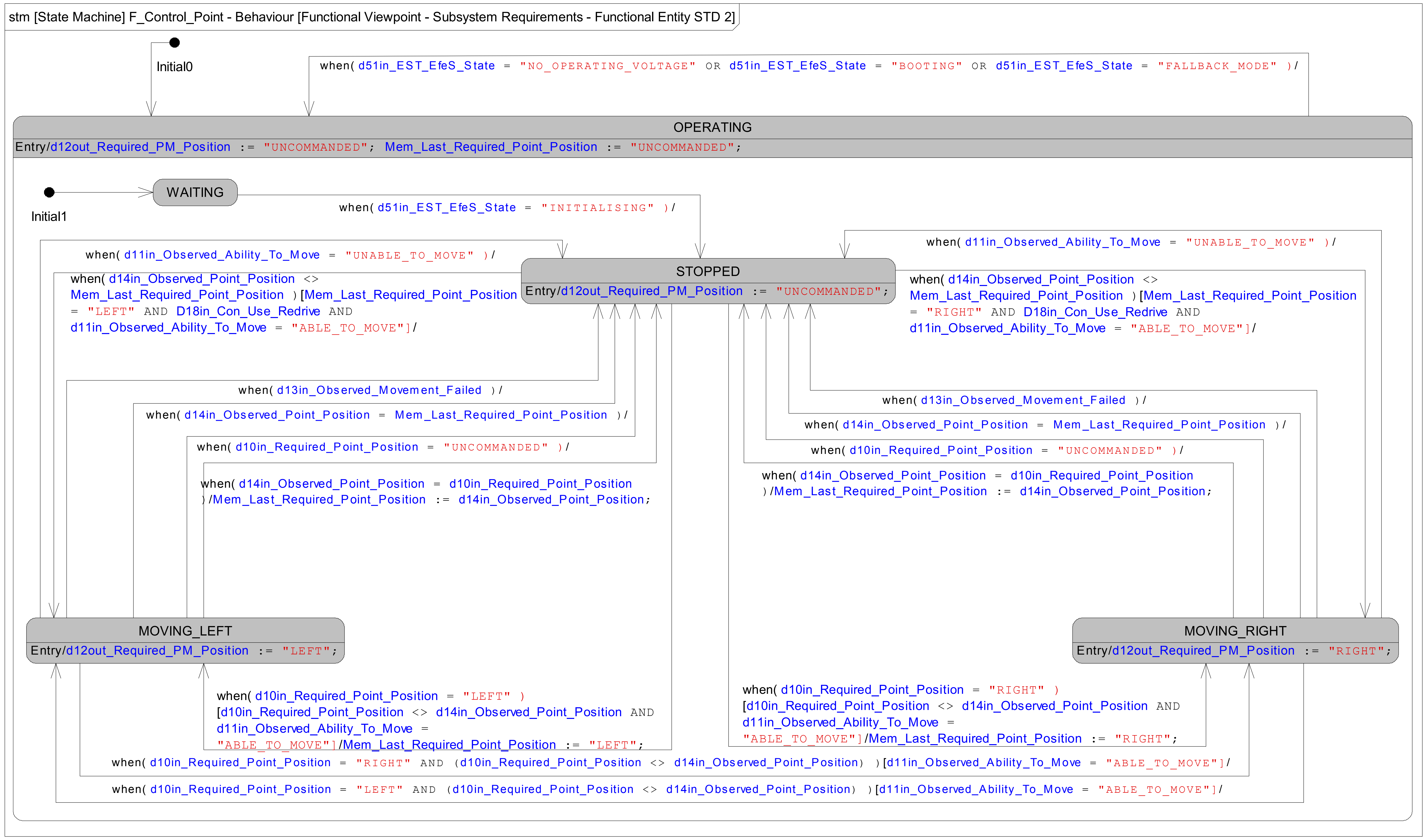
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.6208	Info	[Block] Subsystem Point [Functional Viewpoint - Subsystem Requirements - Functional Architecture Non4W] 	The Subsystem - Point shall provide the functional interfaces shown in "[Block] Subsystem Point [Functional Viewpoint - Subsystem Requirements - Functional Architecture Non4W]", typed by FlowSpecifications. Each FlowSpecification is defined by a set of FlowProperties that specify the possible exchange of information through the particular interface.	Basic non-4-wire single P Basic non-4-wire multiple P

ibd [Block] Subsystem Point [Functional Viewpoint - Subsystem Requirements - Functional Architecture Non4W]
«logical structural entity»
Subsystem Point
p3inout : ~F_SCI_Specific
«functional entity»
: F_SCI_EfeS_Sec
d50out_PDI_Connection_State
: F_EST_EfeS
d51out_EST_EfeS_State
d12in_Required_PM_Position
D20in_Con_tmax_PM_Operation
«functional entity»
: F_Observe_Movement_Failed
d13out_Observed_Movement_Failed
d14in_Observed_Point_Position
t21in_Movement_Failed
P3
SCI-P : SCI_P_Subsystem_P
d50in_PDI_Connection_State
p3inout : F_SCI_Specific
D15in_Con_Observe_Ability_To_Move
d11in_Observed_Ability_To_Move
«functional entity»
: F_SCI_P_Report
P9out : SCI_P_2
P9out
d13in_Observed_Movement_Failed
d17in_Observed_Degraded_Point_Position
d14in_Observed_Point_Position
d51in_EST_EfeS_State
d12out_Required_PM_Position
d13in_Observed_Movement_Failed
D18in_Con_Use_Redrive
«functional entity»
: F_Control_Point
d10in_Required_Point_Position
d14in_Observed_Point_Position
d11in_Observed_Ability_To_Move
d51in_EST_EfeS_State
1..*
d12in_Required_PM_Position
D39out_Move_Left
D38out_Move_Right
d34out_Drive_Stop
D40in_Con_Active
«functional entity»
: F_Control_Non4W_PM
D35in_PM_Position
D36in_Con_Drive_Capability
D37in_Con_Common_Drive
d50in_PDI_Connection_State
P3in : SCI_P_1
d10out_Required_Point_Position
«functional entity»
: F_SCI_P_Receive
P3in : SCI_P_1
d12in_Required_PM_Position
d11in_Observed_Ability_To_Move
d51in_EST_EfeS_State
d30out_Observed_Degraded_Point_Position
d14in_Observed_Point_Position
D26in_Con_PM1n_Crucial_Activation
D27in_Con_PM2_Non_Crucial_Activation
D28in_Con_PM2n_Non_Crucial_Activation
«functional entity»
: F_Observe_Degraded_Point_Position
D22in_PM1_Crucial_Position
D23in_PM1n_Crucial_Position
D24in_PM2_Non_Crucial_Position
D25in_PM2n_Non_Crucial_Position
d51in_EST_EfeS_State
d19out_Observed_Last_Point_End_Position
d14out_Observed_Point_Position
D26in_Con_PM1n_Crucial_Activation
D27in_Con_PM2_Non_Crucial_Activation
«functional entity»
: F_Observe_Overall_Point_Position
D28in_Con_PM2n_Non_Crucial_Activation
D29in_Con_Use_Unintended_Position
D22in_PM1_Crucial_Position
D23in_PM1n_Crucial_Position
D24in_PM2_Non_Crucial_Position
D25in_PM2n_Non_Crucial_Position
d51in_EST_EfeS_State
d11out_Observed_Ability_To_Move
D26in_Con_PM1n_Crucial_Activation
D27in_Con_PM2_Non_Crucial_Activation
D28in_Con_PM2n_Non_Crucial_Activation
«functional entity»
: F_Observe_Ability_To_Move
D29in_Ability_To_Move_PM1_Crucial
D15in_Con_Observe_Ability_To_Move
D30in_Ability_To_Move_PM1n_Crucial
D31in_Ability_To_Move_PM2_Non_Crucial
D32in_Ability_To_Move_PM2n_Non_Crucial
D33in_Internal_Trigger_Ability_To_Move_Point_Available
P1 : Maintainer
SDI-P : Subsystem_MDM_D
SMI-P : Subsystem_MDM_M
SSI-P : Subsystem_SSP
P4 : Basic_Data_Identifier

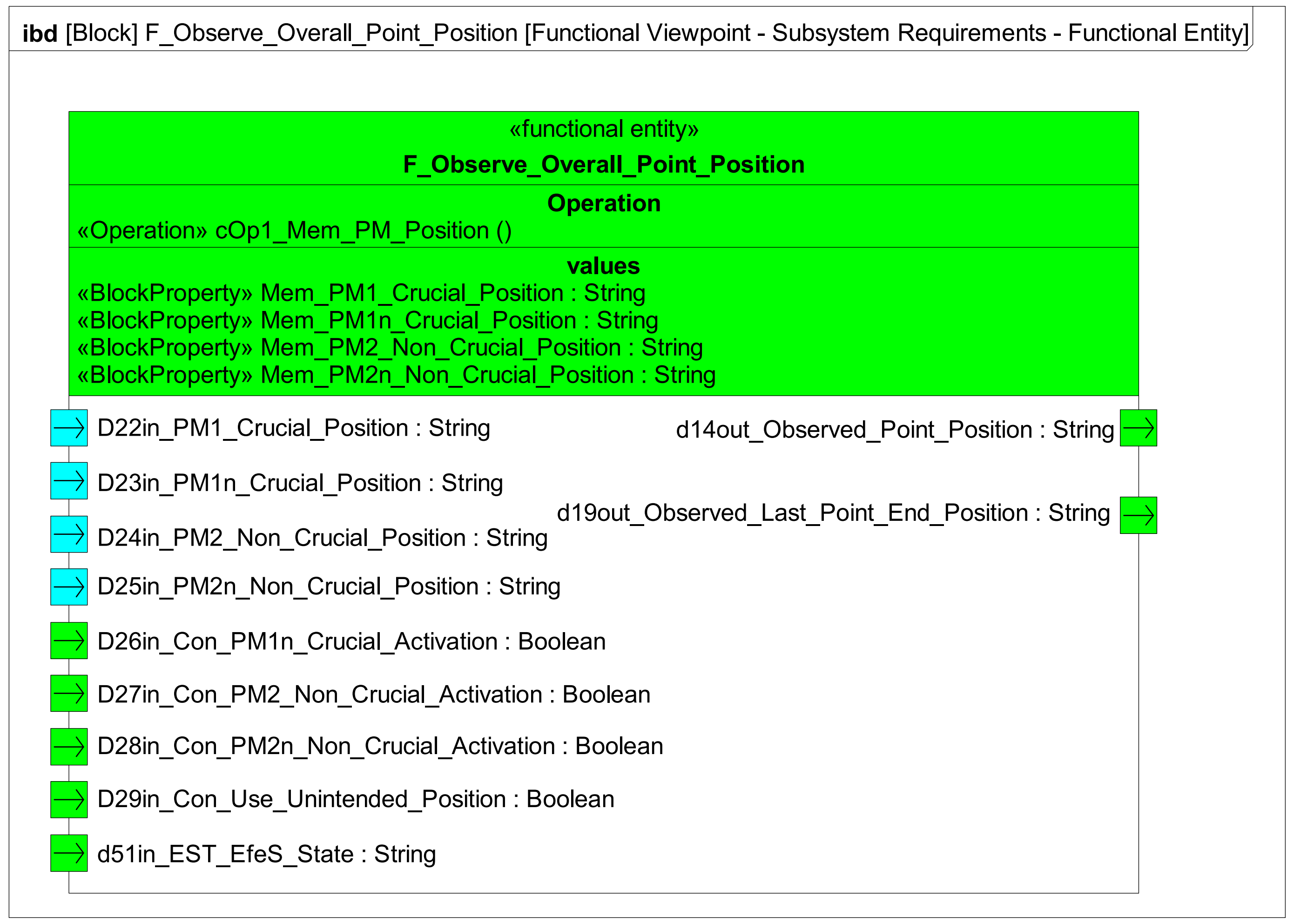
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.6934	Info	[Block] Subsystem Point [Functional Viewpoint - Subsystem Requirements - Functional Architecture 4W] 	The Subsystem - Point shall provide the functional interfaces shown in "[Block] Subsystem Point [Functional Viewpoint - Subsystem Requirements - Functional Architecture 4W]", typed by FlowSpecifications. Each FlowSpecification is defined by a set of FlowProperties that specify the possible exchange of information through the particular interface.	Basic 4-wire single P Basic 4-wire multiple P

ibd [Block] Subsystem Point [Functional Viewpoint - Subsystem Requirements - Functional Architecture 4W]
«logical structural entity»
Subsystem Point
P3
p3inout : ~F_SCI_Specific
«functional entity»
: F_SCI_EfeS_Sec
d50out_PDI_Connection_State
: F_EST_EfeS
d51out_EST_EfeS_State
d12in_Required_PM_Position
D20in_Con_tmax_PM_Operation
«functional entity»
: F_Observe_Movement_Failed
d13out_Observed_Movement_Failed
d14in_Observed_Point_Position
t21in_Movement_Failed
SCI-P : SCI_P_Subsystem_P
d50in_PDI_Connection_State
p3inout : F_SCI_Specific
D15in_Con_Observe_Ability_To_Move
d11in_Observed_Ability_To_Move
«functional entity»
P9out : SCI_P_2
P9out
: F_SCI_P_Report
d13in_Observed_Movement_Failed
d17in_Observed_Degraded_Point_Position
d14in_Observed_Point_Position
d51in_EST_EfeS_State
d12out_Required_PM_Position
d13in_Observed_Movement_Failed
D18in_Con_Use_Redrive
«functional entity»
: F_Control_Point
d10in_Required_Point_Position
d14in_Observed_Point_Position
d11in_Observed_Ability_To_Move
1..*
d51in_EST_EfeS_State
d12in_Required_PM_Position
D23out_Drive_Voltage_Left
D24out_Drive_Voltage_Right
D25out_Detection_Voltage
«functional entity»
: F_Control_And_Observe_4W_PM
D26in_Drive_Voltage_Available
D45in_Con_Active
D20in_Con_Drive_Capability
D27in_4W_PM_Position
d10out_PM_Position
d19out_Ability_To_Move_PM
d50in_PDI_Connection_State
P3in : SCI_P_1
d10out_Required_Point_Position
«functional entity»
: F_SCI_P_Receive
d12in_Required_PM_Position
d11in_Observed_Ability_To_Move
P3in : SCI_P_1
d51in_EST_EfeS_State
d30out_Observed_Degraded_Point_Position
d14in_Observed_Point_Position
D26in_Con_PM1n_Crucial_Activation
D27in_Con_PM2_Non_Crucial_Activation
D28in_Con_PM2n_Non_Crucial_Activation
«functional entity»
: F_Observe_Degraded_Point_Position
D22in_PM1_Crucial_Position
D23in_PM1n_Crucial_Position
D24in_PM2_Non_Crucial_Position
D25in_PM2n_Non_Crucial_Position
d51in_EST_EfeS_State
d14out_Observed_Point_Position
D26in_Con_PM1n_Crucial_Activation
D27in_Con_PM2_Non_Crucial_Activation
«functional entity»
: F_Observe_Overall_Point_Position
D28in_Con_PM2n_Non_Crucial_Activation
D29in_Con_Use_Unintended_Point_Position
D22in_PM1_Crucial_Position
D23in_PM1n_Crucial_Position
D24in_PM2_Non_Crucial_Position
D25in_PM2n_Non_Crucial_Position
d51in_EST_EfeS_State
d11out_Observed_Ability_To_Move
D26in_Con_PM1n_Crucial_Activation
D27in_Con_PM2_Non_Crucial_Activation
D28in_Con_PM2n_Non_Crucial_Activation
«functional entity»
: F_Observe_Ability_To_Move
D29in_Ability_To_Move_PM1_Crucial
D15in_Con_Observe_Ability_To_Move
D30in_Ability_To_Move_PM1n_Crucial
D31in_Ability_To_Move_PM2_Non_Crucial
D32in_Ability_To_Move_PM2n_Non_Crucial
D33in_Internal_Trigger_Ability_To_Move_Point_Available
P1 : Maintainer
SDI-P : Subsystem_MDM_D
SMI-P : Subsystem_MDM_M
SSI-P : Subsystem_SSP
P4 : Basic_Data_Identifier

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.942	Info	P3	The functional control interface to Point machines for the information flow through the interface, which is defined by the FlowSpecification "Point_machine".	Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6378	Info	P1	The functional Maintenance/Operation/Display interface to the Maintainer. The InformationFlow through the interface is defined by "Maintainer".	Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6379	Info	P4	The functional System Data interface to the Basic Data identifier. The InformationFlow through the interface is defined by "Basic_Data_Identifier".	Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6380	Info	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.	Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6381	Info	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".	Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6382	Info	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".	Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6383	Info	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).	Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.5809	Head	**3.3.5 Subsystem Point - Functional Entities**		
Eu.P.6948	Info	F_Control_Point		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6949	Info	[Block] F_Control_Point [Functional Viewpoint - Subsystem Requirements - Functional Entity]  ibd [Block] F_Control_Point [Functional Viewpoint - Subsystem Requirements - Functional Entity] «functional entity» F_Control_Point values «BlockProperty» Mem_Last_Required_Point_Position : String d10in_Required_Point_Position : String d12out_Required_PM_Position : String d11in_Observed_Ability_To_Move : String d13in_Observed_Movement_Failed : Boolean d14in_Observed_Point_Position : String D18in_Con_Use_Redrive : Boolean d51in_EST_EfeS_State : String		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6950	Info	d10in_Required_Point_Position		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6951	Info	d11in_Observed_Ability_To_Move		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6952	Info	d13in_Observed_Movement_Failed		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6953	Info	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	Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
			Information_No_End_Position, Information_End_Position_Arrived, Information_End_Position_Reached, Information_End_Position_Detected and Information_Unintended_Position.	
Eu.P.6954	Info	D18in_Con_Use_Redrive	The port D18in_Con_Use_Redrive provides a configuration value to the Point for Redrive point functionality. true: Point has Redrive point functionality false: Point has no Redrive point functionality	Option Redrive
Eu.P.6956	Info	d12out_Required_PM_Position		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6957	Info	d51in_EST_EfeS_State		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6958	Info	F_Control_Point - Behaviour		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6959	Info	Functional Viewpoint - Subsystem Requirements - Functional Entity STD 2  stm [State Machine] F_Control_Point - Behaviour [Functional Viewpoint - Subsystem Requirements - Functional Entity STD 2] Initial0 when(d51in_EST_EfeS_State = "NO_OPERATING_VOLTAGE" OR d51in_EST_EfeS_State = "BOOTING" OR d51in_EST_EfeS_State = "FALLBACK_MODE")/ OPERATING Entry/d12out_Required_PM_Position := "UNCOMMANDED"; Mem_Last_Required_Point_Position := "UNCOMMANDED"; Initial1 WAITING when(d51in_EST_EfeS_State = "INITIALISING")/ when(d11in_Observed_Ability_To_Move = "UNABLE_TO_MOVE")/ when(d11in_Observed_Ability_To_Move = "UNABLE_TO_MOVE")/ STOPPED Entry/d12out_Required_PM_Position := "UNCOMMANDED"; when(d14in_Observed_Point_Position <> Mem_Last_Required_Point_Position)[Mem_Last_Required_Point_Position = "LEFT" AND D18in_Con_Use_Redrive AND d11in_Observed_Ability_To_Move = "ABLE_TO_MOVE"]/ when(d14in_Observed_Point_Position <> Mem_Last_Required_Point_Position)[Mem_Last_Required_Point_Position = "RIGHT" AND D18in_Con_Use_Redrive AND d11in_Observed_Ability_To_Move = "ABLE_TO_MOVE"]/ when(d13in_Observed_Movement_Failed)/ when(d14in_Observed_Point_Position = Mem_Last_Required_Point_Position)/ when(d10in_Required_Point_Position = "UNCOMMANDED")/ when(d14in_Observed_Point_Position = d10in_Required_Point_Position)/Mem_Last_Required_Point_Position := d14in_Observed_Point_Position; when(d13in_Observed_Movement_Failed)/ when(d14in_Observed_Point_Position = Mem_Last_Required_Point_Position)/ when(d10in_Required_Point_Position = "UNCOMMANDED")/ when(d14in_Observed_Point_Position = d10in_Required_Point_Position)/Mem_Last_Required_Point_Position := d14in_Observed_Point_Position; MOVING_LEFT Entry/d12out_Required_PM_Position := "LEFT"; MOVING_RIGHT Entry/d12out_Required_PM_Position := "RIGHT"; when(d10in_Required_Point_Position = "LEFT")[d10in_Required_Point_Position <> d14in_Observed_Point_Position AND d11in_Observed_Ability_To_Move = "ABLE_TO_MOVE"]/Mem_Last_Required_Point_Position := "LEFT"; when(d10in_Required_Point_Position = "RIGHT")[d10in_Required_Point_Position <> d14in_Observed_Point_Position AND d11in_Observed_Ability_To_Move = "ABLE_TO_MOVE"]/Mem_Last_Required_Point_Position := "RIGHT"; when(d10in_Required_Point_Position = "RIGHT" AND (d10in_Required_Point_Position <> d14in_Observed_Point_Position))[d11in_Observed_Ability_To_Move = "ABLE_TO_MOVE"]/ when(d10in_Required_Point_Position = "LEFT" AND (d10in_Required_Point_Position <> d14in_Observed_Point_Position))[d11in_Observed_Ability_To_Move = "ABLE_TO_MOVE"]/		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6960	Info	Initial0		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P

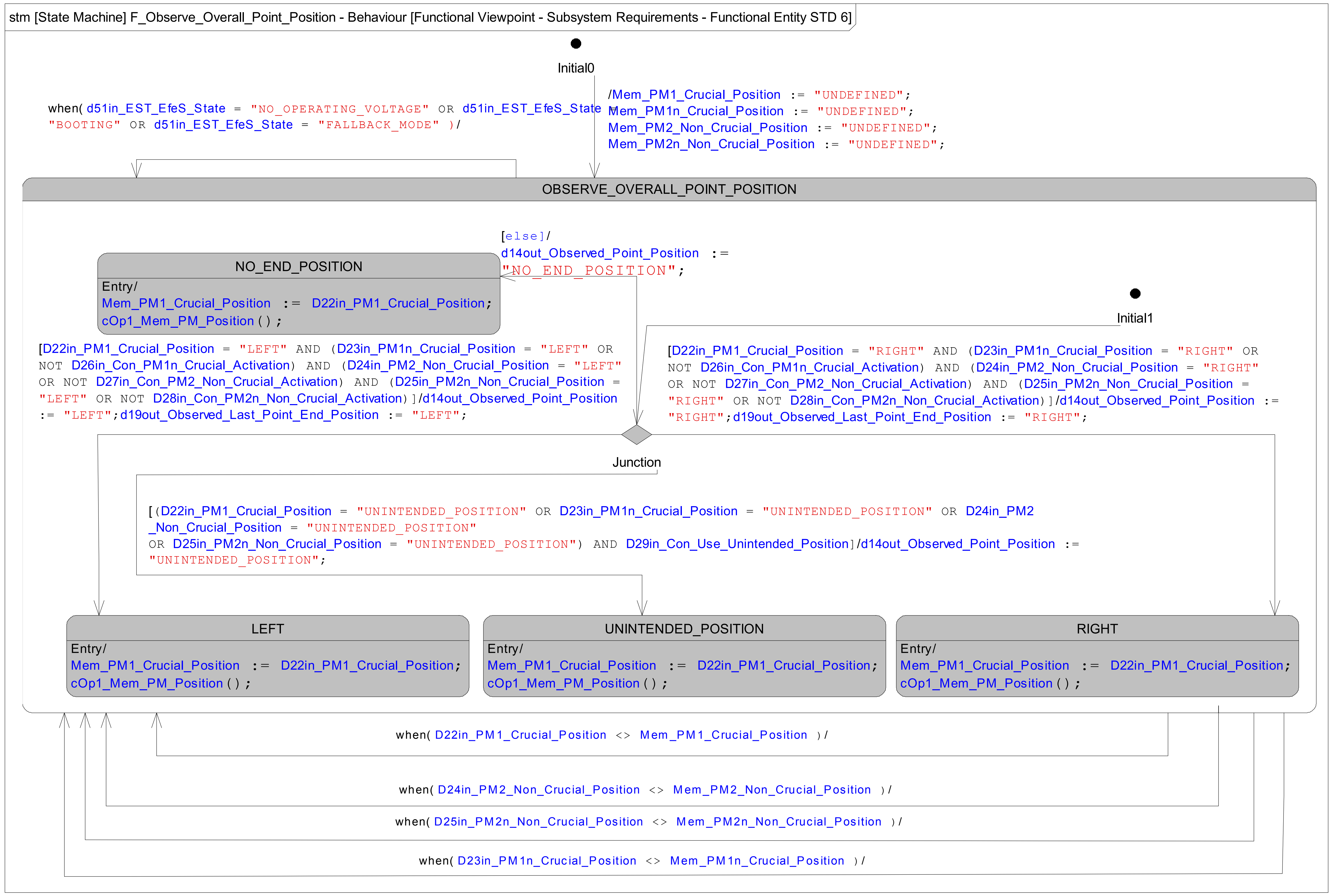
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.6961	Req	/{Initial0 - OPERATING}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6962	Info	OPERATING		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6963	Info	Initial1		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6964	Req	/{Initial1 - WAITING}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6965	Info	MOVING_LEFT		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6966	Req	entry/d12out_Required_PM_Position := "LEFT";{State-internal in MOVING_LEFT}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6967	Req	when(d10in_Required_Point_Position = "RIGHT" AND (d10in_Required_Point_Position <> d14in_Observed_Point_Position))[d11in_Observed_Ability_To_Move = "ABLE_TO_MOVE"]/{MOVING_LEFT - MOVING_RIGHT}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6968	Req	when(d10in_Required_Point_Position = "UNCOMMANDED")/{MOVING_LEFT - STOPPED}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6969	Req	when(d11in_Observed_Ability_To_Move = "UNABLE_TO_MOVE")/{MOVING_LEFT - STOPPED}		Option Able to move
Eu.P.6970	Req	when(d13in_Observed_Movement_Failed)/{MOVING_LEFT - STOPPED}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6972	Req	when(d14in_Observed_Point_Position = d10in_Required_Point_Position)/Mem_Last_Required_Point_Position := d14in_Observed_Point_Position;{MOVING_LEFT - STOPPED}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6973	Req	when(d14in_Observed_Point_Position = Mem_Last_Required_Point_Position)/{MOVING_LEFT - STOPPED}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6974	Info	MOVING_RIGHT		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6975	Req	entry/d12out_Required_PM_Position := "RIGHT";{State-internal in MOVING_RIGHT}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6976	Req	when(d10in_Required_Point_Position = "LEFT" AND (d10in_Required_Point_Position <> d14in_Observed_Point_Position))[d11in_Observed_Ability_To_Move = "ABLE_TO_MOVE"]/{MOVING_RIGHT - MOVING_LEFT}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6977	Req	when(d10in_Required_Point_Position = "UNCOMMANDED")/{MOVING_RIGHT - STOPPED}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6978	Req	when(d11in_Observed_Ability_To_Move = "UNABLE_TO_MOVE")/{MOVING_RIGHT - STOPPED}		Option Able to move
Eu.P.6979	Req	when(d13in_Observed_Movement_Failed)/{MOVING_RIGHT - STOPPED}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6981	Req	when(d14in_Observed_Point_Position = d10in_Required_Point_Position)/Mem_Last_Required_Point_Position := d14in_Observed_Point_Position;{MOVING_RIGHT - STOPPED}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.6982	Req	when(d14in_Observed_Point_Position = Mem_Last_Required_Point_Position)/{MOVING_RIGHT - STOPPED}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6983	Info	STOPPED		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6984	Req	entry/d12out_Required_PM_Position := "UNCOMMANDED";{State-internal in STOPPED}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6985	Req	when(d10in_Required_Point_Position = "LEFT")[d10in_Required_Point_Position <> d14in_Observed_Point_Position AND d11in_Observed_Ability_To_Move = "ABLE_TO_MOVE"]/Mem_Last_Required_Point_Position := "LEFT";{STOPPED - MOVING_LEFT}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6986	Req	when(d10in_Required_Point_Position = "RIGHT")[d10in_Required_Point_Position <> d14in_Observed_Point_Position AND d11in_Observed_Ability_To_Move = "ABLE_TO_MOVE"]/Mem_Last_Required_Point_Position := "RIGHT";{STOPPED - MOVING_RIGHT}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6980	Req	when(d14in_Observed_Point_Position <> Mem_Last_Required_Point_Position)[Mem_Last_Required_Point_Position = "RIGHT" AND D18in_Con_Use_Redrive AND d11in_Observed_Ability_To_Move = "ABLE_TO_MOVE"]/{STOPPED - MOVING_RIGHT}		Option Redrive
Eu.P.6971	Req	when(d14in_Observed_Point_Position <> Mem_Last_Required_Point_Position)[Mem_Last_Required_Point_Position = "LEFT" AND D18in_Con_Use_Redrive AND d11in_Observed_Ability_To_Move = "ABLE_TO_MOVE"]/{STOPPED - MOVING_LEFT}		Option Redrive
Eu.P.6987	Req	entry/d12out_Required_PM_Position := "UNCOMMANDED"; Mem_Last_Required_Point_Position := "UNCOMMANDED";{State-internal in OPERATING}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6988	Info	WAITING		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6989	Req	when(d51in_EST_EfeS_State = "INITIALISING")/{WAITING - STOPPED}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6990	Req	when(d51in_EST_EfeS_State = "NO_OPERATING_VOLTAGE" OR d51in_EST_EfeS_State = "BOOTING" OR d51in_EST_EfeS_State = "FALLBACK_MODE")/{OPERATING - OPERATING}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.4548	Info	F_Observe_Overall_Point_Position		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.4549	Info	[Block] F_Observe_Overall_Point_Position [Functional Viewpoint - Subsystem Requirements - Functional Entity]  ibd [Block] F_Observe_Overall_Point_Position [Functional Viewpoint - Subsystem Requirements - Functional Entity] «functional entity» F_Observe_Overall_Point_Position Operation «Operation» cOp1_Mem_PM_Position () values «BlockProperty» Mem_PM1_Crucial_Position : String «BlockProperty» Mem_PM1n_Crucial_Position : String «BlockProperty» Mem_PM2_Non_Crucial_Position : String «BlockProperty» Mem_PM2n_Non_Crucial_Position : String D22in_PM1_Crucial_Position : String D23in_PM1n_Crucial_Position : String D24in_PM2_Non_Crucial_Position : String D25in_PM2n_Non_Crucial_Position : String D26in_Con_PM1n_Crucial_Activation : Boolean D27in_Con_PM2_Non_Crucial_Activation : Boolean D28in_Con_PM2n_Non_Crucial_Activation : Boolean D29in_Con_Use_Unintended_Position : Boolean d51in_EST_EfeS_State : String d14out_Observed_Point_Position : String d19out_Observed_Last_Point_End_Position : String		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P

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

Eu.P.6309	Req	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	Basic non-4-wire single P
				Basic non-4-wire multiple P
				Basic 4-wire single P
				Basic 4-wire multiple P

Eu.P.4566	Info	d51in_EST_EfeS_State		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.4567	Info	D22in_PM1_Crucial_Position	The port D22in_PM1_Crucial_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_Arrived, Information_End_Position_Reached, Information_End_Position_Detected and Information_Unintended_Position.	Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.4568	Info	D23in_PM1n_Crucial_Position	The port D23in_PM1n_Crucial_Position represents the Point position that was observed at the interface of the n-th crucial Point machine. The Point position is derived from the InformationFlow Information_No_End_Position, Information_End_Position_Arrived and Information_Unintended_Position.	Basic non-4-wire multiple P Basic 4-wire multiple P
Eu.P.4586	Info	d14out_Observed_Point_Position		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.4979	Info	D26in_Con_PM1n_Crucial_Activation	The port D26in_Con_PM1n_Crucial_Activation provides configuration values for the Point machine PM1n crucial and representing a n-th crucial Point machine in configurations with more than one Point machine. true: Point machine PM1n Crucial is activated false: Point machine PM1n Crucial is not activated	Basic non-4-wire multiple P Basic 4-wire multiple P
Eu.P.6033	Info	D24in_PM2_Non_Crucial_Position	The port D24in_PM2_Non_Crucial_Position represents the Point position that was observed at the interface of the first non-crucial Point machine. The Point position is derived from the InformationFlowInformation_No_End_Position, Information_End_Position_Arrived and Information_Unintended_Position.	Basic non-4-wire multiple P Basic 4-wire multiple P
Eu.P.6034	Info	D25in_PM2n_Non_Crucial_Position	The port D25in_PM2n_Non_Crucial_Position represents the Point position that was observed at the interface of the n-th non-crucial Point machine. The Point position is derived from the InformationFlow Information_No_End_Position, Information_End_Position_Arrived and Information_Unintended_Position.	Basic non-4-wire multiple P Basic 4-wire multiple P
Eu.P.6035	Info	D27in_Con_PM2_Non_Crucial_Activation	The port D27in_Con_PM2_Non_Crucial_Activation provides configuration values for the Point machine PM2 non crucial, representing the 1st non crucial Point machine in configurations with more than one Point machine. true: Point machine PM2 Non Crucial is activated false: Point machine PM2 Non Crucial is not activated	Basic non-4-wire multiple P Basic 4-wire multiple P
Eu.P.6036	Info	D28in_Con_PM2n_Non_Crucial_Activation	The port D28in_Con_PM2n_Non_Crucial_Activation provides configuration values for the Point machine PM2n non crucial, representing a n-th non crucial Point machine in configurations with more than one Point machine. true: Point machine PM2n Non Crucial is activated false: Point machine PM2n Non Crucial is not activated	Basic non-4-wire multiple P Basic 4-wire multiple P

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.6037	Info	D29in_Con_Use_Unintended_Position	The port D29in_Con_Use_Unintended_Position provides a configuration value to the Subsystem - Point enabling interpretation of Unintended position. true: Point is configured for interpretation of Unintended position false: Point is not configured for interpretation of Unintended position	Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6512	Info	d19out_Observed_Last_Point_End_Position		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.4587	Info	F_Observe_Overall_Point_Position - Behaviour		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.4588	Info	Functional Viewpoint - Subsystem Requirements - Functional Entity STD 6  stm [State Machine] F_Observe_Overall_Point_Position - Behaviour [Functional Viewpoint - Subsystem Requirements - Functional Entity STD 6] Initial0 when(d51in_EST_EfeS_State = "NO_OPERATING_VOLTAGE" OR d51in_EST_EfeS_State "BOOTING" OR d51in_EST_EfeS_State = "FALLBACK_MODE")/ /Mem_PM1_Crucial_Position := "UNDEFINED"; Mem_PM1n_Crucial_Position := "UNDEFINED"; Mem_PM2_Non_Crucial_Position := "UNDEFINED"; Mem_PM2n_Non_Crucial_Position := "UNDEFINED"; OBSERVE_OVERALL_POINT_POSITION [else]/ d14out_Observed_Point_Position := "NO_END_POSITION"; NO_END_POSITION Entry/ Mem_PM1_Crucial_Position := D22in_PM1_Crucial_Position; cOp1_Mem_PM_Position ( ) ; Initial1 [D22in_PM1_Crucial_Position = "LEFT" AND (D23in_PM1n_Crucial_Position = "LEFT" OR NOT D26in_Con_PM1n_Crucial_Activation) AND (D24in_PM2_Non_Crucial_Position = "LEFT" OR NOT D27in_Con_PM2_Non_Crucial_Activation) AND (D25in_PM2n_Non_Crucial_Position = "LEFT" OR NOT D28in_Con_PM2n_Non_Crucial_Activation)]/d14out_Observed_Point_Position := "LEFT";d19out_Observed_Last_Point_End_Position := "LEFT"; [D22in_PM1_Crucial_Position = "RIGHT" AND (D23in_PM1n_Crucial_Position = "RIGHT" OR NOT D26in_Con_PM1n_Crucial_Activation) AND (D24in_PM2_Non_Crucial_Position = "RIGHT" OR NOT D27in_Con_PM2_Non_Crucial_Activation) AND (D25in_PM2n_Non_Crucial_Position = "RIGHT" OR NOT D28in_Con_PM2n_Non_Crucial_Activation)]/d14out_Observed_Point_Position := "RIGHT";d19out_Observed_Last_Point_End_Position := "RIGHT"; Junction [(D22in_PM1_Crucial_Position = "UNINTENDED_POSITION" OR D23in_PM1n_Crucial_Position = "UNINTENDED_POSITION" OR D24in_PM2_Non_Crucial_Position = "UNINTENDED_POSITION" OR D25in_PM2n_Non_Crucial_Position = "UNINTENDED_POSITION") AND D29in_Con_Use_Unintended_Position]/d14out_Observed_Point_Position := "UNINTENDED_POSITION"; LEFT Entry/ Mem_PM1_Crucial_Position := D22in_PM1_Crucial_Position; cOp1_Mem_PM_Position ( ) ; UNINTENDED_POSITION Entry/ Mem_PM1_Crucial_Position := D22in_PM1_Crucial_Position; cOp1_Mem_PM_Position ( ) ; RIGHT Entry/ Mem_PM1_Crucial_Position := D22in_PM1_Crucial_Position; cOp1_Mem_PM_Position ( ) ; when(D22in_PM1_Crucial_Position <> Mem_PM1_Crucial_Position)/ when(D24in_PM2_Non_Crucial_Position <> Mem_PM2_Non_Crucial_Position)/ when(D25in_PM2n_Non_Crucial_Position <> Mem_PM2n_Non_Crucial_Position)/ when(D23in_PM1n_Crucial_Position <> Mem_PM1n_Crucial_Position)/		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6241	Info	Initial0		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P

Eu.P.6242	Req	/Mem_PM1_Crucial_Position := "UNDEFINED";
		Mem_PM1n_Crucial_Position := "UNDEFINED";
		Mem_PM2_Non_Crucial_Position := "UNDEFINED";
		Mem_PM2n_Non_Crucial_Position := "UNDEFINED";{Initial0 - OBSERVE_OVERALL_POINT_POSITION}
				Basic non-4-wire single P
				Basic non-4-wire multiple P
				Basic 4-wire single P
				Basic 4-wire multiple P

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.6310	Info	OBSERVE_OVERALL_POINT_POSITION		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6311	Info	Initial1		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6312	Req	/{Initial1 - Junction}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6313	Info	Junction		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6314	Req	[D22in_PM1_Crucial_Position = "LEFT" AND (D23in_PM1n_Crucial_Position = "LEFT" OR NOT D26in_Con_PM1n_Crucial_Activation) AND (D24in_PM2_Non_Crucial_Position = "LEFT" OR NOT D27in_Con_PM2_Non_Crucial_Activation) AND (D25in_PM2n_Non_Crucial_Position = "LEFT" OR NOT D28in_Con_PM2n_Non_Crucial_Activation)]/d14out_Observed_Point_Position := "LEFT";d19out_Observed_Last_Point_End_Position := "LEFT";{Junction - LEFT}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6315	Req	[else]/ d14out_Observed_Point_Position := "NO_END_POSITION";{Junction - NO_END_POSITION}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6316	Req	[D22in_PM1_Crucial_Position = "RIGHT" AND (D23in_PM1n_Crucial_Position = "RIGHT" OR NOT D26in_Con_PM1n_Crucial_Activation) AND (D24in_PM2_Non_Crucial_Position = "RIGHT" OR NOT D27in_Con_PM2_Non_Crucial_Activation) AND (D25in_PM2n_Non_Crucial_Position = "RIGHT" OR NOT D28in_Con_PM2n_Non_Crucial_Activation)]/d14out_Observed_Point_Position := "RIGHT";d19out_Observed_Last_Point_End_Position := "RIGHT";{Junction - RIGHT}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6317	Req	[(D22in_PM1_Crucial_Position = "UNINTENDED_POSITION" OR D23in_PM1n_Crucial_Position = "UNINTENDED_POSITION" OR D24in_PM2_Non_Crucial_Position = "UNINTENDED_POSITION" OR D25in_PM2n_Non_Crucial_Position = "UNINTENDED_POSITION") AND D29in_Con_Use_Unintended_Position]/d14out_Observed_Point_Position := "UNINTENDED_POSITION";{Junction - UNINTENDED_POSITION}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6318	Info	LEFT		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P

Eu.P.6408	Req	entry/
		Mem_PM1_Crucial_Position := D22in_PM1_Crucial_Position;
		cOp1_Mem_PM_Position();{State-internal in LEFT}
				Basic non-4-wire single P
				Basic non-4-wire multiple P
				Basic 4-wire single P
				Basic 4-wire multiple P

Eu.P.6319	Info	NO_END_POSITION		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P

Eu.P.6409	Req	entry/
		Mem_PM1_Crucial_Position := D22in_PM1_Crucial_Position;
		cOp1_Mem_PM_Position();{State-internal in NO_END_POSITION}
				Basic non-4-wire single P
				Basic non-4-wire multiple P
				Basic 4-wire single P
				Basic 4-wire multiple P

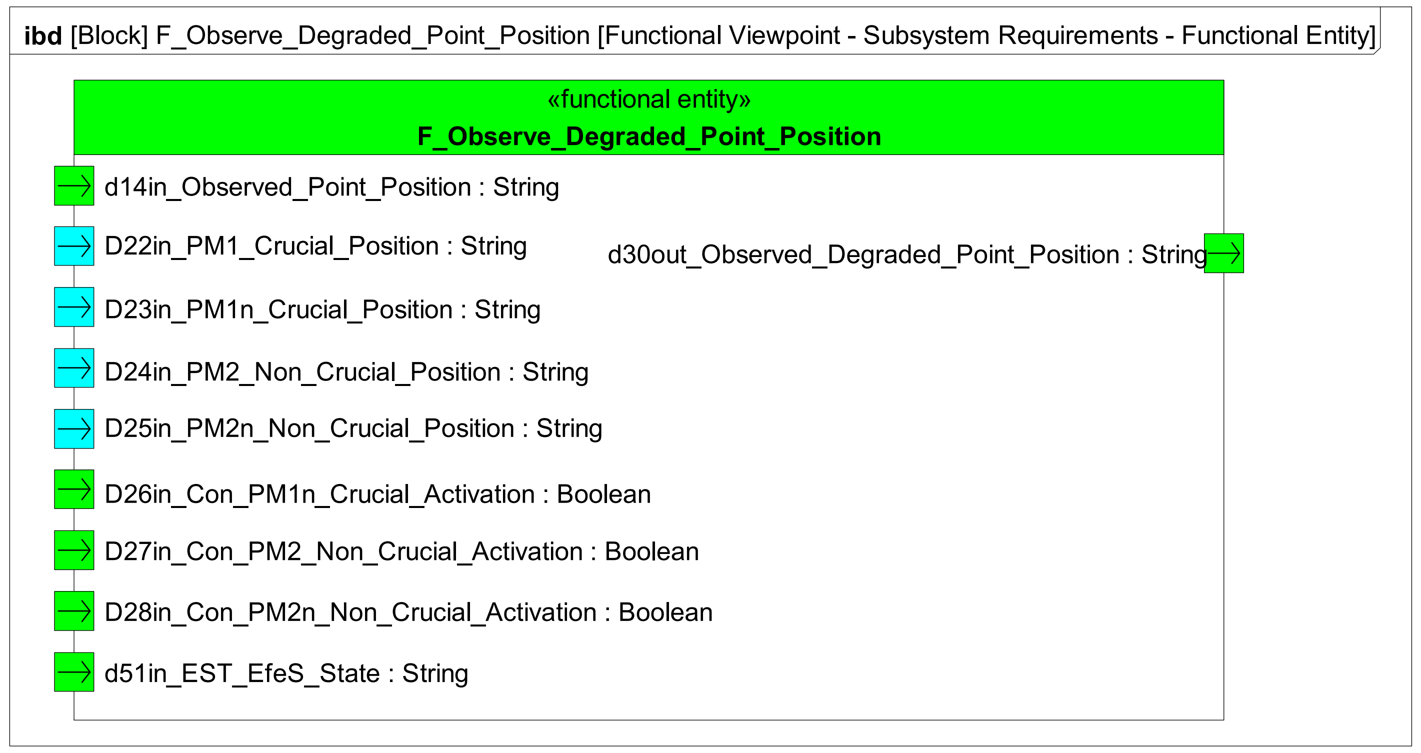
Eu.P.6320	Info	RIGHT		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P

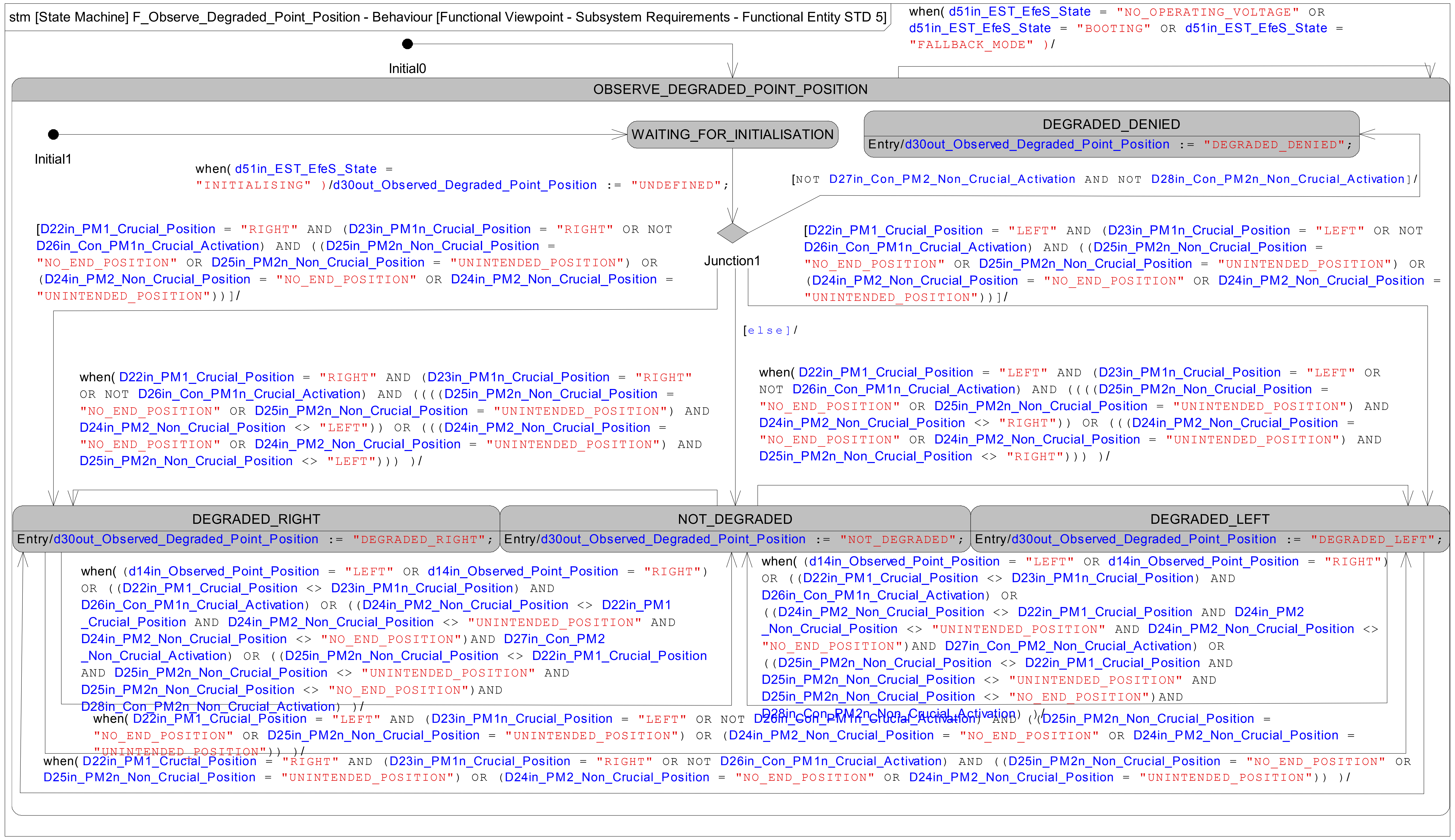
Eu.P.6410	Req	entry/
		Mem_PM1_Crucial_Position := D22in_PM1_Crucial_Position;
		cOp1_Mem_PM_Position();{State-internal in RIGHT}
				Basic non-4-wire single P
				Basic non-4-wire multiple P
				Basic 4-wire single P
				Basic 4-wire multiple P

Eu.P.6321	Info	UNINTENDED_POSITION		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P

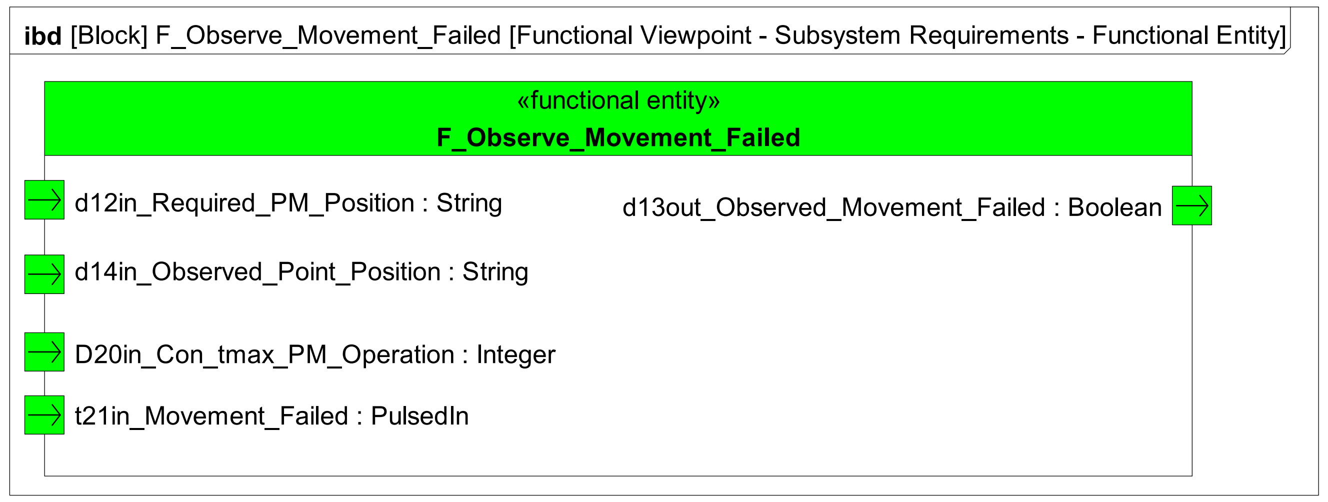
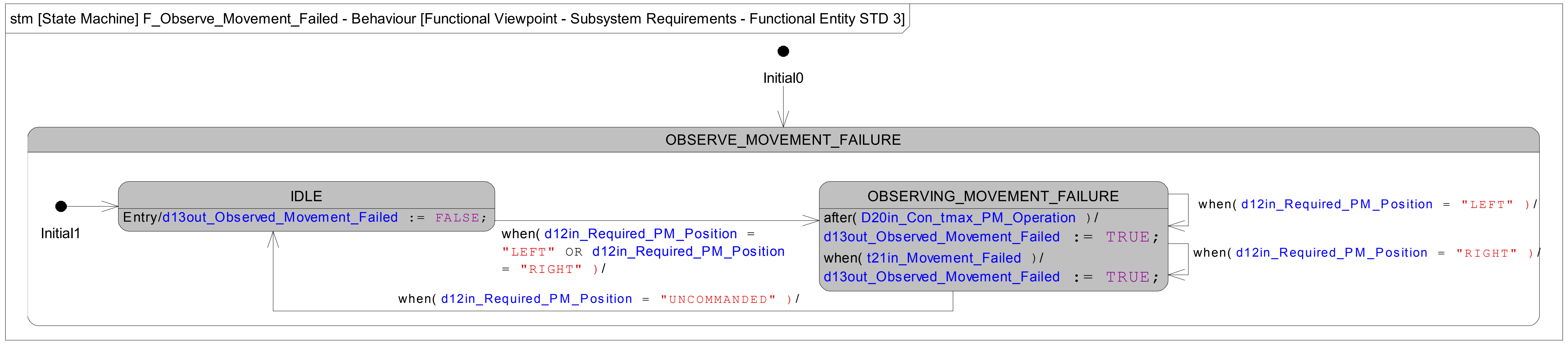
Eu.P.6411	Req	entry/
		Mem_PM1_Crucial_Position := D22in_PM1_Crucial_Position;
		cOp1_Mem_PM_Position();{State-internal in UNINTENDED_POSITION}
				Basic non-4-wire single P
				Basic non-4-wire multiple P
				Basic 4-wire single P
				Basic 4-wire multiple P

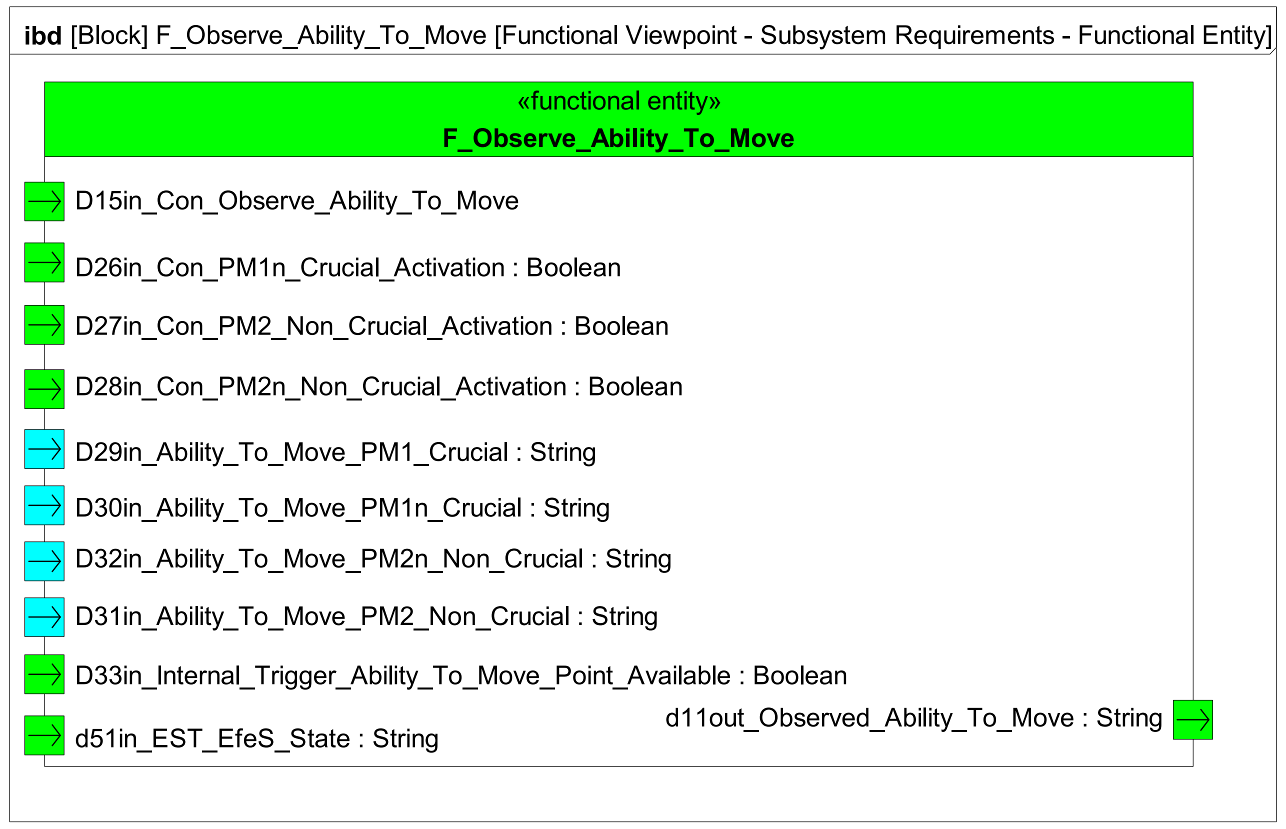
Eu.P.6322	Req	when(D22in_PM1_Crucial_Position <> Mem_PM1_Crucial_Position)/{OBSERVE_OVERALL_POINT_POSITION - OBSERVE_OVERALL_POINT_POSITION}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6323	Req	when(D23in_PM1n_Crucial_Position <> Mem_PM1n_Crucial_Position)/{OBSERVE_OVERALL_POINT_POSITION - OBSERVE_OVERALL_POINT_POSITION}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.6324	Req	when(D24in_PM2_Non_Crucial_Position <> Mem_PM2_Non_Crucial_Position)/{OBSERVE_OVERALL_POINT_POSITION - OBSERVE_OVERALL_POINT_POSITION}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6325	Req	when(D25in_PM2n_Non_Crucial_Position <> Mem_PM2n_Non_Crucial_Position)/{OBSERVE_OVERALL_POINT_POSITION - OBSERVE_OVERALL_POINT_POSITION}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6326	Req	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}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6002	Info	F_Observe_Degraded_Point_Position		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6003	Info	[Block] F_Observe_Degraded_Point_Position [Functional Viewpoint - Subsystem Requirements - Functional Entity]  ibd [Block] F_Observe_Degraded_Point_Position [Functional Viewpoint - Subsystem Requirements - Functional Entity] «functional entity» F_Observe_Degraded_Point_Position d14in_Observed_Point_Position : String D22in_PM1_Crucial_Position : String d30out_Observed_Degraded_Point_Position : String D23in_PM1n_Crucial_Position : String D24in_PM2_Non_Crucial_Position : String D25in_PM2n_Non_Crucial_Position : String D26in_Con_PM1n_Crucial_Activation : Boolean D27in_Con_PM2_Non_Crucial_Activation : Boolean D28in_Con_PM2n_Non_Crucial_Activation : Boolean d51in_EST_EfeS_State : String		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6004	Info	D22in_PM1_Crucial_Position	The port D22in_PM1_Crucial_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_Arrived, Information_End_Position_Reached, Information_End_Position_Detected and Information_Unintended_Position.	Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6005	Info	D23in_PM1n_Crucial_Position	The port D23in_PM1n_Crucial_Position represents the Point position that was observed at the interface of the n-th crucial Point machine. The Point position is derived from the InformationFlow Information_No_End_Position, Information_End_Position_Arrived and Information_Unintended_Position.	Basic non-4-wire multiple P Basic 4-wire multiple P
Eu.P.6006	Info	D24in_PM2_Non_Crucial_Position	The port D24in_PM2_Non_Crucial_Position represents the Point position that was observed at the interface of the first non-crucial Point machine. The Point position is derived from the InformationFlow Information_No_End_Position, Information_End_Position_Arrived and Information_Unintended_Position.	Basic non-4-wire multiple P Basic 4-wire multiple P
Eu.P.6007	Info	D25in_PM2n_Non_Crucial_Position	The port D25in_PM2n_Non_Crucial_Position represents the Point position that was observed at the interface of the n-th non-crucial Point machine. The Point position is derived from the InformationFlow Information_No_End_Position, Information_End_Position_Arrived and Information_Unintended_Position.	Basic non-4-wire multiple P Basic 4-wire multiple P
Eu.P.6008	Info	d30out_Observed_Degraded_Point_Position		Basic non-4-wire multiple P Basic 4-wire multiple P
Eu.P.6009	Info	d51in_EST_EfeS_State		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P

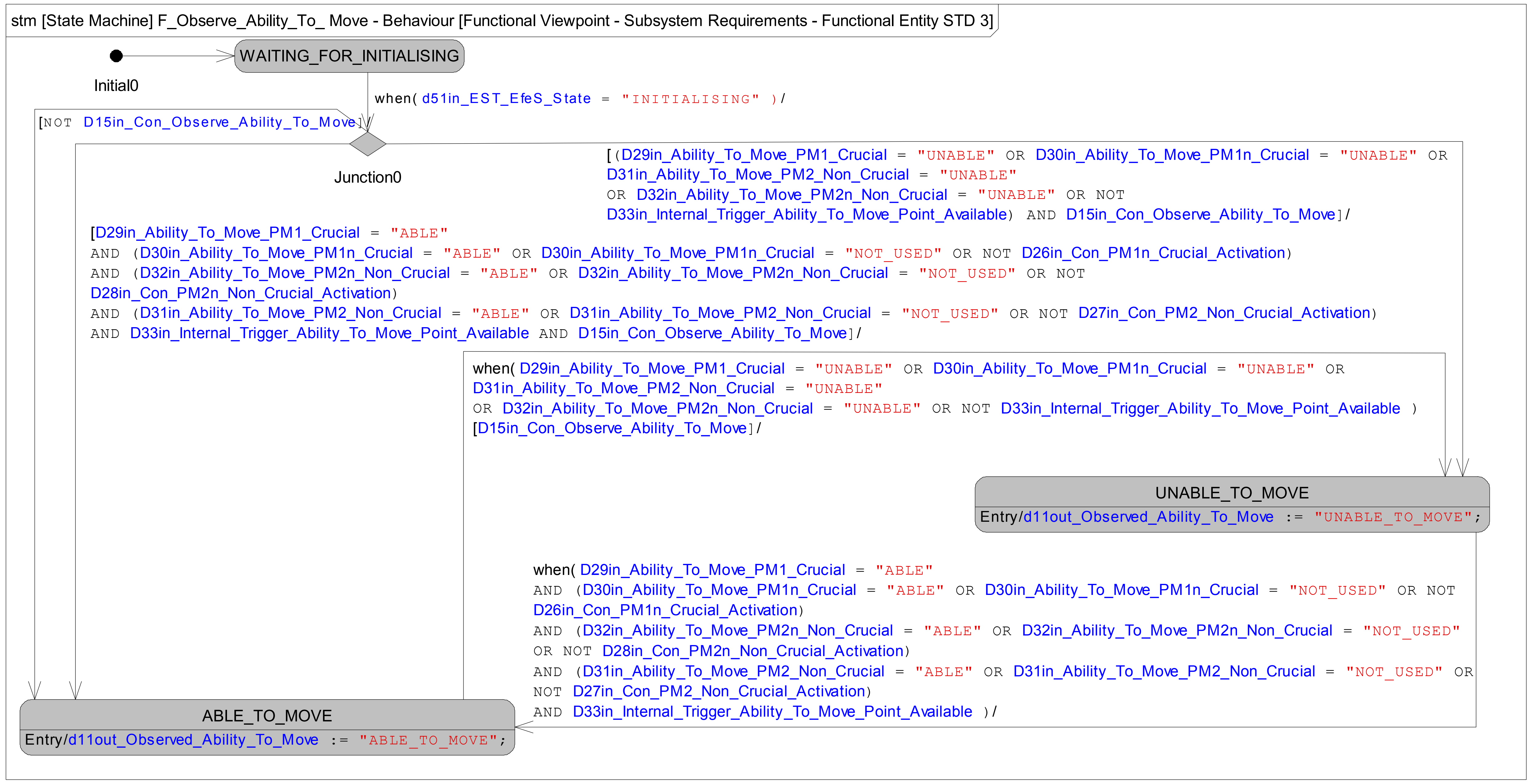
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.6010	Info	D26in_Con_PM1n_Crucial_Activation	The port D26in_Con_PM1n_Crucial_Activation provides configuration values for the Point machine PM1n crucial and representing a n-th crucial Point machine in configurations with more than one Point machine. true: Point machine PM1n Crucial is activated false: Point machine PM1n Crucial is not activated	Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6011	Info	D27in_Con_PM2_Non_Crucial_Activation	The port D27in_Con_PM2_Non_Crucial_Activation provides configuration values for the Point machine PM2 non crucial, representing the 1st non crucial Point machine in configurations with more than one Point machine. true: Point machine PM2 Crucial is activated false: Point machine PM2 Crucial is not activated	Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6012	Info	D28in_Con_PM2n_Non_Crucial_Activation	The port D28in_Con_PM2n_Non_Crucial_Activation provides configuration values for the Point machine PM2n non crucial, representing a n-th non crucial Point machine in configurations with more than one Point machine. true: Point machine PM2n Non Crucial is activated false: Point machine PM2n Non Crucial is not activated	Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6013	Info	d14in_Observed_Point_Position		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6014	Info	F_Observe_Degraded_Point_Position - Behaviour		Basic non-4-wire multiple P Basic 4-wire multiple P
Eu.P.6015	Info	Functional Viewpoint - Subsystem Requirements - Functional Entity STD 5  stm [State Machine] F_Observe_Degraded_Point_Position - Behaviour [Functional Viewpoint - Subsystem Requirements - Functional Entity STD 5] when(d51in_EST_EfeS_State = "NO_OPERATING_VOLTAGE" OR d51in_EST_EfeS_State = "BOOTING" OR d51in_EST_EfeS_State = "FALLBACK_MODE")/ Initial0 OBSERVE_DEGRADED_POINT_POSITION Initial1 WAITING_FOR_INITIALISATION DEGRADED_DENIED Entry/d30out_Observed_Degraded_Point_Position := "DEGRADED_DENIED"; when(d51in_EST_EfeS_State = "INITIALISING")/d30out_Observed_Degraded_Point_Position := "UNDEFINED"; [NOT D27in_Con_PM2_Non_Crucial_Activation AND NOT D28in_Con_PM2n_Non_Crucial_Activation]/ [D22in_PM1_Crucial_Position = "RIGHT" AND (D23in_PM1n_Crucial_Position = "RIGHT" OR NOT D26in_Con_PM1n_Crucial_Activation) AND ((D25in_PM2n_Non_Crucial_Position = "NO_END_POSITION" OR D25in_PM2n_Non_Crucial_Position = "UNINTENDED_POSITION") OR (D24in_PM2_Non_Crucial_Position = "NO_END_POSITION" OR D24in_PM2_Non_Crucial_Position = "UNINTENDED_POSITION"))]/ Junction1 [D22in_PM1_Crucial_Position = "LEFT" AND (D23in_PM1n_Crucial_Position = "LEFT" OR NOT D26in_Con_PM1n_Crucial_Activation) AND ((D25in_PM2n_Non_Crucial_Position = "NO_END_POSITION" OR D25in_PM2n_Non_Crucial_Position = "UNINTENDED_POSITION") OR (D24in_PM2_Non_Crucial_Position = "NO_END_POSITION" OR D24in_PM2_Non_Crucial_Position = "UNINTENDED_POSITION"))]/ [else]/ when(D22in_PM1_Crucial_Position = "RIGHT" AND (D23in_PM1n_Crucial_Position = "RIGHT" OR NOT D26in_Con_PM1n_Crucial_Activation) AND ((((D25in_PM2n_Non_Crucial_Position = "NO_END_POSITION" OR D25in_PM2n_Non_Crucial_Position = "UNINTENDED_POSITION") AND D24in_PM2_Non_Crucial_Position <> "LEFT")) OR (((D24in_PM2_Non_Crucial_Position = "NO_END_POSITION" OR D24in_PM2_Non_Crucial_Position = "UNINTENDED_POSITION") AND D25in_PM2n_Non_Crucial_Position <> "LEFT"))))/ when(D22in_PM1_Crucial_Position = "LEFT" AND (D23in_PM1n_Crucial_Position = "LEFT" OR NOT D26in_Con_PM1n_Crucial_Activation) AND ((((D25in_PM2n_Non_Crucial_Position = "NO_END_POSITION" OR D25in_PM2n_Non_Crucial_Position = "UNINTENDED_POSITION") AND D24in_PM2_Non_Crucial_Position <> "RIGHT")) OR (((D24in_PM2_Non_Crucial_Position = "NO_END_POSITION" OR D24in_PM2_Non_Crucial_Position = "UNINTENDED_POSITION") AND D25in_PM2n_Non_Crucial_Position <> "RIGHT"))))/ DEGRADED_RIGHT Entry/d30out_Observed_Degraded_Point_Position := "DEGRADED_RIGHT"; NOT_DEGRADED Entry/d30out_Observed_Degraded_Point_Position := "NOT_DEGRADED"; DEGRADED_LEFT Entry/d30out_Observed_Degraded_Point_Position := "DEGRADED_LEFT"; when((d14in_Observed_Point_Position = "LEFT" OR d14in_Observed_Point_Position = "RIGHT") OR ((D22in_PM1_Crucial_Position <> D23in_PM1n_Crucial_Position) AND D26in_Con_PM1n_Crucial_Activation) OR ((D24in_PM2_Non_Crucial_Position <> D22in_PM1 _Crucial_Position AND D24in_PM2_Non_Crucial_Position <> "UNINTENDED_POSITION" AND D24in_PM2_Non_Crucial_Position <> "NO_END_POSITION")AND D27in_Con_PM2 _Non_Crucial_Activation) OR ((D25in_PM2n_Non_Crucial_Position <> D22in_PM1_Crucial_Position AND D25in_PM2n_Non_Crucial_Position <> "UNINTENDED_POSITION" AND D25in_PM2n_Non_Crucial_Position <> "NO_END_POSITION")AND D28in_Con_PM2n_Non_Crucial_Activation))/ when((d14in_Observed_Point_Position = "LEFT" OR d14in_Observed_Point_Position = "RIGHT") OR ((D22in_PM1_Crucial_Position <> D23in_PM1n_Crucial_Position) AND D26in_Con_PM1n_Crucial_Activation) OR ((D24in_PM2_Non_Crucial_Position <> D22in_PM1_Crucial_Position AND D24in_PM2 _Non_Crucial_Position <> "UNINTENDED_POSITION" AND D24in_PM2_Non_Crucial_Position <> "NO_END_POSITION")AND D27in_Con_PM2_Non_Crucial_Activation) OR ((D25in_PM2n_Non_Crucial_Position <> D22in_PM1_Crucial_Position AND D25in_PM2n_Non_Crucial_Position <> "UNINTENDED_POSITION" AND D25in_PM2n_Non_Crucial_Position <> "NO_END_POSITION")AND D28in_Con_PM2n_Non_Crucial_Activation))/ when(D22in_PM1_Crucial_Position = "LEFT" AND (D23in_PM1n_Crucial_Position = "LEFT" OR NOT D26in_Con_PM1n_Crucial_Activation) AND ((D25in_PM2n_Non_Crucial_Position = "NO_END_POSITION" OR D25in_PM2n_Non_Crucial_Position = "UNINTENDED_POSITION") OR (D24in_PM2_Non_Crucial_Position = "NO_END_POSITION" OR D24in_PM2_Non_Crucial_Position = "UNINTENDED_POSITION")))/ when(D22in_PM1_Crucial_Position = "RIGHT" AND (D23in_PM1n_Crucial_Position = "RIGHT" OR NOT D26in_Con_PM1n_Crucial_Activation) AND ((D25in_PM2n_Non_Crucial_Position = "NO_END_POSITION" OR D25in_PM2n_Non_Crucial_Position = "UNINTENDED_POSITION") OR (D24in_PM2_Non_Crucial_Position = "NO_END_POSITION" OR D24in_PM2_Non_Crucial_Position = "UNINTENDED_POSITION")))/		Basic non-4-wire multiple P Basic 4-wire multiple P

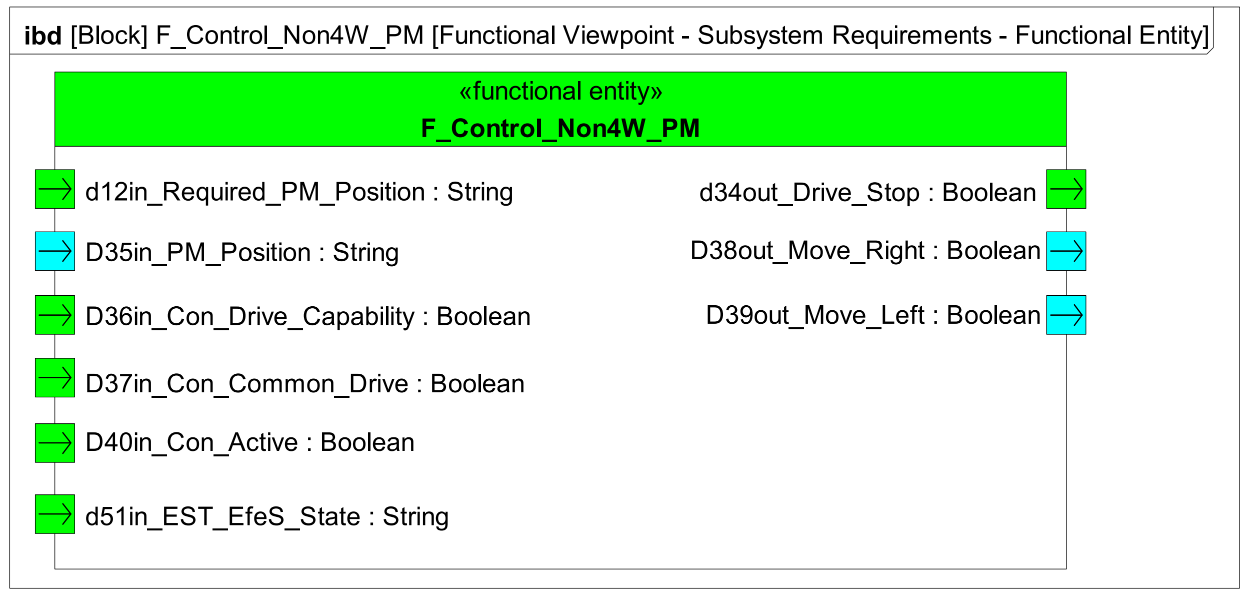
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.6230	Info	Initial0		Basic non-4-wire multiple P Basic 4-wire multiple P
Eu.P.6231	Req	/{Initial0 - OBSERVE_DEGRADED_POINT_POSITION}		Basic non-4-wire multiple P Basic 4-wire multiple P
Eu.P.6232	Info	OBSERVE_DEGRADED_POINT_POSITION		Basic non-4-wire multiple P Basic 4-wire multiple P
Eu.P.6016	Info	DEGRADED_LEFT		Basic non-4-wire multiple P Basic 4-wire multiple P
Eu.P.6017	Req	entry/d30out_Observed_Degraded_Point_Position := "DEGRADED_LEFT";{State-internal in DEGRADED_LEFT}		Basic non-4-wire multiple P Basic 4-wire multiple P
Eu.P.6018	Req	when((d14in_Observed_Point_Position = "LEFT" OR d14in_Observed_Point_Position = "RIGHT") OR ((D22in_PM1_Crucial_Position <> D23in_PM1n_Crucial_Position) AND D26in_Con_PM1n_Crucial_Activation) OR ((D24in_PM2_Non_Crucial_Position <> D22in_PM1_Crucial_Position AND D24in_PM2_Non_Crucial_Position <> "UNINTENDED_POSITION" AND D24in_PM2_Non_Crucial_Position <> "NO_END_POSITION")AND D27in_Con_PM2_Non_Crucial_Activation) OR ((D25in_PM2n_Non_Crucial_Position <> D22in_PM1_Crucial_Position AND D25in_PM2n_Non_Crucial_Position <> "UNINTENDED_POSITION" AND D25in_PM2n_Non_Crucial_Position <> "NO_END_POSITION")AND D28in_Con_PM2n_Non_Crucial_Activation))/{DEGRADED_LEFT - NOT_DEGRADED}		Basic non-4-wire multiple P Basic 4-wire multiple P
Eu.P.6019	Req	when(D22in_PM1_Crucial_Position = "RIGHT" AND (D23in_PM1n_Crucial_Position = "RIGHT" OR NOT D26in_Con_PM1n_Crucial_Activation) AND ((D25in_PM2n_Non_Crucial_Position = "NO_END_POSITION" OR D25in_PM2n_Non_Crucial_Position = "UNINTENDED_POSITION") OR (D24in_PM2_Non_Crucial_Position = "NO_END_POSITION" OR D24in_PM2_Non_Crucial_Position = "UNINTENDED_POSITION")))/{DEGRADED_LEFT - DEGRADED_RIGHT}		Basic non-4-wire multiple P Basic 4-wire multiple P
Eu.P.6020	Info	DEGRADED_RIGHT		Basic non-4-wire multiple P Basic 4-wire multiple P
Eu.P.6021	Req	entry/d30out_Observed_Degraded_Point_Position := "DEGRADED_RIGHT";{State-internal in DEGRADED_RIGHT}		Basic non-4-wire multiple P Basic 4-wire multiple P
Eu.P.6022	Req	when((d14in_Observed_Point_Position = "LEFT" OR d14in_Observed_Point_Position = "RIGHT") OR ((D22in_PM1_Crucial_Position <> D23in_PM1n_Crucial_Position) AND D26in_Con_PM1n_Crucial_Activation) OR ((D24in_PM2_Non_Crucial_Position <> D22in_PM1 _Crucial_Position AND D24in_PM2_Non_Crucial_Position <> "UNINTENDED_POSITION" AND D24in_PM2_Non_Crucial_Position <> "NO_END_POSITION")AND D27in_Con_PM2_Non_Crucial_Activation) OR ((D25in_PM2n_Non_Crucial_Position <> D22in_PM1_Crucial_Position AND D25in_PM2n_Non_Crucial_Position <> "UNINTENDED_POSITION" AND D25in_PM2n_Non_Crucial_Position <> "NO_END_POSITION")AND D28in_Con_PM2n_Non_Crucial_Activation))/{DEGRADED_RIGHT - NOT_DEGRADED}		Basic non-4-wire multiple P Basic 4-wire multiple P
Eu.P.6023	Req	when(D22in_PM1_Crucial_Position = "LEFT" AND (D23in_PM1n_Crucial_Position = "LEFT" OR NOT D26in_Con_PM1n_Crucial_Activation) AND ((D25in_PM2n_Non_Crucial_Position = "NO_END_POSITION" OR D25in_PM2n_Non_Crucial_Position = "UNINTENDED_POSITION") OR (D24in_PM2_Non_Crucial_Position = "NO_END_POSITION" OR D24in_PM2_Non_Crucial_Position = "UNINTENDED_POSITION")))/{DEGRADED_RIGHT - DEGRADED_LEFT}		Basic non-4-wire multiple P Basic 4-wire multiple P
Eu.P.6024	Info	Initial1		Basic non-4-wire multiple P Basic 4-wire multiple P
Eu.P.6025	Req	/{Initial1 - WAITING_FOR_INITIALISATION}		Basic non-4-wire multiple P Basic 4-wire multiple P
Eu.P.6233	Info	Junction1		Basic non-4-wire multiple P Basic 4-wire multiple P
Eu.P.6234	Req	[D22in_PM1_Crucial_Position = "LEFT" AND (D23in_PM1n_Crucial_Position = "LEFT" OR NOT D26in_Con_PM1n_Crucial_Activation) AND ((D25in_PM2n_Non_Crucial_Position = "NO_END_POSITION" OR D25in_PM2n_Non_Crucial_Position = "UNINTENDED_POSITION") OR (D24in_PM2_Non_Crucial_Position = "NO_END_POSITION" OR D24in_PM2_Non_Crucial_Position = "UNINTENDED_POSITION"))]/{Junction1 - DEGRADED_LEFT}		Basic non-4-wire multiple P Basic 4-wire multiple P
Eu.P.6235	Req	[D22in_PM1_Crucial_Position = "RIGHT" AND (D23in_PM1n_Crucial_Position = "RIGHT" OR NOT D26in_Con_PM1n_Crucial_Activation) AND ((D25in_PM2n_Non_Crucial_Position = "NO_END_POSITION" OR D25in_PM2n_Non_Crucial_Position = "UNINTENDED_POSITION") OR (D24in_PM2_Non_Crucial_Position = "NO_END_POSITION" OR D24in_PM2_Non_Crucial_Position = "UNINTENDED_POSITION"))]/{Junction1 - DEGRADED_RIGHT}		Basic non-4-wire multiple P Basic 4-wire multiple P
Eu.P.6236	Req	[else]/{Junction1 - NOT_DEGRADED}		Basic non-4-wire multiple P Basic 4-wire multiple P
Eu.P.6237	Req	[NOT D27in_Con_PM2_Non_Crucial_Activation AND NOT D28in_Con_PM2n_Non_Crucial_Activation]/{Junction1 - DEGRADED_DENIED}		Basic non-4-wire multiple P Basic 4-wire multiple P
Eu.P.6026	Info	NOT_DEGRADED		Basic non-4-wire multiple P Basic 4-wire multiple P
Eu.P.6027	Req	entry/d30out_Observed_Degraded_Point_Position := "NOT_DEGRADED";{State-internal in NOT_DEGRADED}		Basic non-4-wire multiple P Basic 4-wire multiple P
Eu.P.6028	Req	when(D22in_PM1_Crucial_Position = "LEFT" AND (D23in_PM1n_Crucial_Position = "LEFT" OR NOT D26in_Con_PM1n_Crucial_Activation) AND ((((D25in_PM2n_Non_Crucial_Position = "NO_END_POSITION" OR D25in_PM2n_Non_Crucial_Position = "UNINTENDED_POSITION") AND D24in_PM2_Non_Crucial_Position <> "RIGHT")) OR (((D24in_PM2_Non_Crucial_Position = "NO_END_POSITION" OR D24in_PM2_Non_Crucial_Position = "UNINTENDED_POSITION") AND D25in_PM2n_Non_Crucial_Position <> "RIGHT"))))/{NOT_DEGRADED - DEGRADED_LEFT}		Basic non-4-wire multiple P Basic 4-wire multiple P
Eu.P.6029	Req	when(D22in_PM1_Crucial_Position = "RIGHT" AND (D23in_PM1n_Crucial_Position = "RIGHT" OR NOT D26in_Con_PM1n_Crucial_Activation) AND ((((D25in_PM2n_Non_Crucial_Position = "NO_END_POSITION" OR D25in_PM2n_Non_Crucial_Position = "UNINTENDED_POSITION") AND D24in_PM2_Non_Crucial_Position <> "LEFT")) OR (((D24in_PM2_Non_Crucial_Position = "NO_END_POSITION" OR D24in_PM2_Non_Crucial_Position = "UNINTENDED_POSITION") AND D25in_PM2n_Non_Crucial_Position <> "LEFT"))))/{NOT_DEGRADED - DEGRADED_RIGHT}		Basic non-4-wire multiple P Basic 4-wire multiple P
Eu.P.6238	Info	WAITING_FOR_INITIALISATION		Basic non-4-wire multiple P Basic 4-wire multiple P
Eu.P.6239	Req	when(d51in_EST_EfeS_State = "INITIALISING")/d30out_Observed_Degraded_Point_Position := "UNDEFINED";{WAITING_FOR_INITIALISATION - Junction1}		Basic non-4-wire multiple P Basic 4-wire multiple P
Eu.P.6240	Req	when(d51in_EST_EfeS_State = "NO_OPERATING_VOLTAGE" OR d51in_EST_EfeS_State = "BOOTING" OR d51in_EST_EfeS_State = "FALLBACK_MODE")/{OBSERVE_DEGRADED_POINT_POSITION - OBSERVE_DEGRADED_POINT_POSITION}		Basic non-4-wire multiple P Basic 4-wire multiple P
Eu.P.6406	Info	DEGRADED_DENIED		Basic non-4-wire multiple P Basic 4-wire multiple P
Eu.P.6407	Req	entry/d30out_Observed_Degraded_Point_Position := "DEGRADED_DENIED";{State-internal in DEGRADED_DENIED}		Basic non-4-wire multiple P Basic 4-wire multiple P
Eu.P.6993	Info	F_Observe_Movement_Failed		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.6994	Info	[Block] F_Observe_Movement_Failed [Functional Viewpoint - Subsystem Requirements - Functional Entity]  ibd [Block] F_Observe_Movement_Failed [Functional Viewpoint - Subsystem Requirements - Functional Entity] «functional entity» F_Observe_Movement_Failed d12in_Required_PM_Position : String d13out_Observed_Movement_Failed : Boolean d14in_Observed_Point_Position : String D20in_Con_tmax_PM_Operation : Integer t21in_Movement_Failed : PulsedIn		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6995	Info	d12in_Required_PM_Position		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6996	Info	d13out_Observed_Movement_Failed		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6997	Info	d14in_Observed_Point_Position		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6998	Info	D20in_Con_tmax_PM_Operation	The port D20in_Con_tmax_PM_Operation refines the time value Con_tmax_Point_Operation.	Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7015	Info	t21in_Movement_Failed	The port t21in_Movement_Failed represents all internal occurrences of conditions to the Subsystem - Point	Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6999	Info	F_Observe_Movement_Failed - Behaviour		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7000	Info	Functional Viewpoint - Subsystem Requirements - Functional Entity STD 3  stm [State Machine] F_Observe_Movement_Failed - Behaviour [Functional Viewpoint - Subsystem Requirements - Functional Entity STD 3] Initial0 OBSERVE_MOVEMENT_FAILURE Initial1 IDLE Entry/d13out_Observed_Movement_Failed := FALSE; when(d12in_Required_PM_Position = "LEFT" OR d12in_Required_PM_Position = "RIGHT")/ OBSERVING_MOVEMENT_FAILURE after(D20in_Con_tmax_PM_Operation)/ d13out_Observed_Movement_Failed := TRUE; when(t21in_Movement_Failed)/ d13out_Observed_Movement_Failed := TRUE; when(d12in_Required_PM_Position = "LEFT")/ when(d12in_Required_PM_Position = "RIGHT")/ when(d12in_Required_PM_Position = "UNCOMMANDED")/		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7001	Info	Initial0		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7002	Req	/{Initial0 - OBSERVE_MOVEMENT_FAILURE}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7003	Info	OBSERVE_MOVEMENT_FAILURE		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P

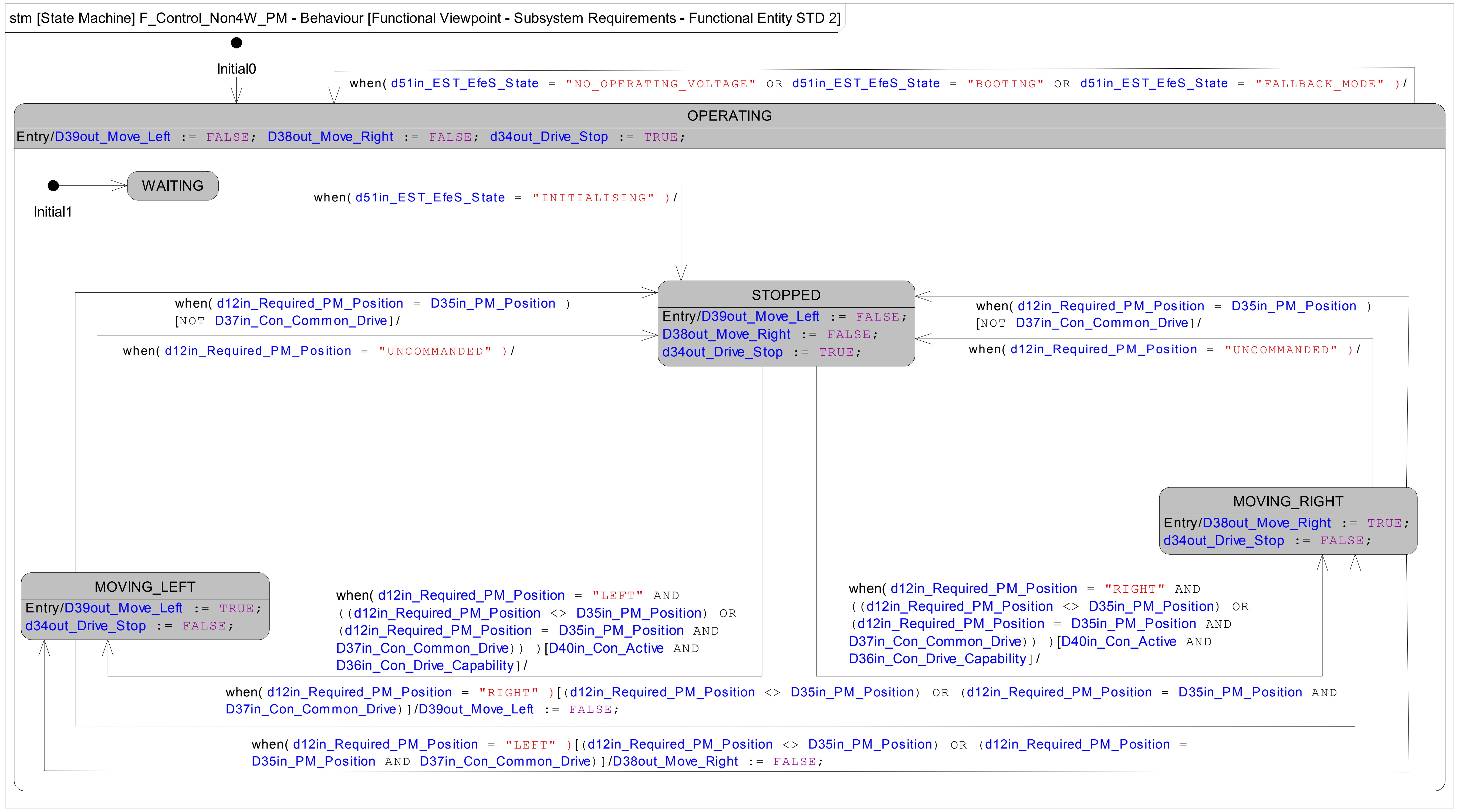
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.7004	Info	IDLE		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7005	Req	entry/d13out_Observed_Movement_Failed := FALSE;{State-internal in IDLE}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7006	Req	when(d12in_Required_PM_Position = "LEFT" OR d12in_Required_PM_Position = "RIGHT")/{IDLE - OBSERVING_MOVEMENT_FAILURE}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7007	Info	Initial1		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7008	Req	/{Initial1 - IDLE}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7009	Info	OBSERVING_MOVEMENT_FAILURE		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7010	Req	after(D20in_Con_tmax_PM_Operation)/ d13out_Observed_Movement_Failed := TRUE;{State-internal in OBSERVING_MOVEMENT_FAILURE}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7011	Req	when(t21in_Movement_Failed)/ d13out_Observed_Movement_Failed := TRUE;{State-internal in OBSERVING_MOVEMENT_FAILURE}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7012	Req	when(d12in_Required_PM_Position = "LEFT")/{OBSERVING_MOVEMENT_FAILURE - OBSERVING_MOVEMENT_FAILURE}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7013	Req	when(d12in_Required_PM_Position = "RIGHT")/{OBSERVING_MOVEMENT_FAILURE - OBSERVING_MOVEMENT_FAILURE}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7014	Req	when(d12in_Required_PM_Position = "UNCOMMANDED")/{OBSERVING_MOVEMENT_FAILURE - IDLE}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.5980	Info	F_Observe_Ability_To_Move		Basic 4-wire single P Basic 4-wire multiple P
Eu.P.5981	Info	[Block] F_Observe_Ability_To_Move [Functional Viewpoint - Subsystem Requirements - Functional Entity]  ibd [Block] F_Observe_Ability_To_Move [Functional Viewpoint - Subsystem Requirements - Functional Entity] «functional entity» F_Observe_Ability_To_Move D15in_Con_Observe_Ability_To_Move D26in_Con_PM1n_Crucial_Activation : Boolean D27in_Con_PM2_Non_Crucial_Activation : Boolean D28in_Con_PM2n_Non_Crucial_Activation : Boolean D29in_Ability_To_Move_PM1_Crucial : String D30in_Ability_To_Move_PM1n_Crucial : String D32in_Ability_To_Move_PM2n_Non_Crucial : String D31in_Ability_To_Move_PM2_Non_Crucial : String D33in_Internal_Trigger_Ability_To_Move_Point_Available : Boolean d51in_EST_EfeS_State : String d11out_Observed_Ability_To_Move : String		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.5982	Info	D29in_Ability_To_Move_PM1_Crucial	The port D29in_Ability_To_Move_PM1_Crucial refines the InformationFlow Information_Ability_To_Move_Point from the first Point machine to Subsystem - Point. Note - Drive voltage levels monitored shall be defined by national specifications. In future phases of the System Pillar, national specifications will be replaced by harmonised specifications.	Option Able to move
Eu.P.5983	Info	D30in_Ability_To_Move_PM1n_Crucial	The port D30in_Ability_To_Move_PM1n_Crucial refines the InformationFlow Information_Ability_To_Move_Point from the n-th Point machine to Subsystem - Point. Note - Drive voltage levels monitored shall be defined by national specifications. In future phases of the System Pillar, national specifications will be replaced by harmonised specifications.	Option Able to move
Eu.P.5984	Info	D31in_Ability_To_Move_PM2_Non_Crucial	The port D31in_Ability_To_Move_PM2 _Non_Crucial refines the InformationFlow Information_Ability_To_Move_Point from the n-th Point machine to Subsystem - Point. Note - Drive voltage levels monitored shall be defined by national specifications. In future phases of the System Pillar, national specifications will be replaced by harmonised specifications.	Option Able to move
Eu.P.5985	Info	D32in_Ability_To_Move_PM2n_Non_Crucial	The port D32in_Ability_To_Move_PM2n_Non_Crucial refines the InformationFlow Information_Ability_To_Move_Point from the n-th Point machine to Subsystem - Point. Note - Drive voltage levels monitored shall be defined by national specifications. In future phases of the System Pillar, national specifications will be replaced by harmonised specifications.	Option Able to move
Eu.P.5986	Info	D33in_Internal_Trigger_Ability_To_Move_Point_Available	The port D33in_Internal_Trigger_Ability_To_Move_Point_Available represents all internal occurrences of conditions to the Subsystem - Point, leading to unability to move.	Option Able to move
Eu.P.5987	Info	d11out_Observed_Ability_To_Move		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.5988	Info	d51in_EST_EfeS_State		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6224	Info	D26in_Con_PM1n_Crucial_Activation	The port D26in_Con_PM1n_Crucial_Activation provides configuration values for the Point machine PM1n crucial and representing a n-th crucial Point machine in configurations with more than one Point machine. true: Point machine PM1n Crucial is activated false: Point machine PM1n Crucial is not activated	Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6225	Info	D27in_Con_PM2_Non_Crucial_Activation	The port D27in_Con_PM2_Non_Crucial_Activation provides configuration values for the Point machine PM2 non crucial, representing the 1st non crucial Point machine in configurations with more than one Point machine. true: Point machine PM2 Non Crucial is activated false: Point machine PM2 Non Crucial is not activated	Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6226	Info	D28in_Con_PM2n_Non_Crucial_Activation	The port D28in_Con_PM2n_Non_Crucial_Activation provides configuration values for the Point machine PM2n non crucial, representing a n-th non crucial Point machine in configurations with more than one Point machine. true: Point machine PM2n Non Crucial is activated false: Point machine PM2n Non Crucial is not activated	Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.5992	Info	F_Observe_Ability_To_ Move - Behaviour		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.5993	Info	Functional Viewpoint - Subsystem Requirements - Functional Entity STD 3  stm [State Machine] F_Observe_Ability_To_ Move - Behaviour [Functional Viewpoint - Subsystem Requirements - Functional Entity STD 3] WAITING_FOR_INITIALISING Initial0 when(d51in_EST_EfeS_State = "INITIALISING")/ [NOT D15in_Con_Observe_Ability_To_Move]/ Junction0 [(D29in_Ability_To_Move_PM1_Crucial = "UNABLE" OR D30in_Ability_To_Move_PM1n_Crucial = "UNABLE" OR D31in_Ability_To_Move_PM2_Non_Crucial = "UNABLE" OR D32in_Ability_To_Move_PM2n_Non_Crucial = "UNABLE" OR NOT D33in_Internal_Trigger_Ability_To_Move_Point_Available) AND D15in_Con_Observe_Ability_To_Move]/ [D29in_Ability_To_Move_PM1_Crucial = "ABLE" AND (D30in_Ability_To_Move_PM1n_Crucial = "ABLE" OR D30in_Ability_To_Move_PM1n_Crucial = "NOT_USED" OR NOT D26in_Con_PM1n_Crucial_Activation) AND (D32in_Ability_To_Move_PM2n_Non_Crucial = "ABLE" OR D32in_Ability_To_Move_PM2n_Non_Crucial = "NOT_USED" OR NOT D28in_Con_PM2n_Non_Crucial_Activation) AND (D31in_Ability_To_Move_PM2_Non_Crucial = "ABLE" OR D31in_Ability_To_Move_PM2_Non_Crucial = "NOT_USED" OR NOT D27in_Con_PM2_Non_Crucial_Activation) AND D33in_Internal_Trigger_Ability_To_Move_Point_Available AND D15in_Con_Observe_Ability_To_Move]/ when(D29in_Ability_To_Move_PM1_Crucial = "UNABLE" OR D30in_Ability_To_Move_PM1n_Crucial = "UNABLE" OR D31in_Ability_To_Move_PM2_Non_Crucial = "UNABLE" OR D32in_Ability_To_Move_PM2n_Non_Crucial = "UNABLE" OR NOT D33in_Internal_Trigger_Ability_To_Move_Point_Available) [D15in_Con_Observe_Ability_To_Move]/ UNABLE_TO_MOVE Entry/d11out_Observed_Ability_To_Move := "UNABLE_TO_MOVE"; when(D29in_Ability_To_Move_PM1_Crucial = "ABLE" AND (D30in_Ability_To_Move_PM1n_Crucial = "ABLE" OR D30in_Ability_To_Move_PM1n_Crucial = "NOT_USED" OR NOT D26in_Con_PM1n_Crucial_Activation) AND (D32in_Ability_To_Move_PM2n_Non_Crucial = "ABLE" OR D32in_Ability_To_Move_PM2n_Non_Crucial = "NOT_USED" OR NOT D28in_Con_PM2n_Non_Crucial_Activation) AND (D31in_Ability_To_Move_PM2_Non_Crucial = "ABLE" OR D31in_Ability_To_Move_PM2_Non_Crucial = "NOT_USED" OR NOT D27in_Con_PM2_Non_Crucial_Activation) AND D33in_Internal_Trigger_Ability_To_Move_Point_Available)/ ABLE_TO_MOVE Entry/d11out_Observed_Ability_To_Move := "ABLE_TO_MOVE";		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6227	Info	Initial0		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6228	Req	/{Initial0 - WAITING_FOR_INITIALISING}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6229	Info	WAITING_FOR_INITIALISING		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6405	Req	when(d51in_EST_EfeS_State = "INITIALISING")/{WAITING_FOR_INITIALISING - Junction0}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6396	Info	ABLE_TO_MOVE		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6397	Req	entry/d11out_Observed_Ability_To_Move := "ABLE_TO_MOVE";{State-internal in ABLE_TO_MOVE}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6398	Req	when(D29in_Ability_To_Move_PM1_Crucial = "UNABLE" OR D30in_Ability_To_Move_PM1n_Crucial = "UNABLE" OR D31in_Ability_To_Move_PM2_Non_Crucial = "UNABLE" OR D32in_Ability_To_Move_PM2n_Non_Crucial = "UNABLE" OR NOT D33in_Internal_Trigger_Ability_To_Move_Point_Available)[D15in_Con_Observe_Ability_To_Move]/{ABLE_TO_MOVE - UNABLE_TO_MOVE}		Option Able to move
Eu.P.6399	Info	Junction0		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6400	Req	[D29in_Ability_To_Move_PM1_Crucial = "ABLE" AND (D30in_Ability_To_Move_PM1n_Crucial = "ABLE" OR D30in_Ability_To_Move_PM1n_Crucial = "NOT_USED" OR NOT D26in_Con_PM1n_Crucial_Activation) AND (D32in_Ability_To_Move_PM2n_Non_Crucial = "ABLE" OR D32in_Ability_To_Move_PM2n_Non_Crucial = "NOT_USED" OR NOT D28in_Con_PM2n_Non_Crucial_Activation) AND (D31in_Ability_To_Move_PM2_Non_Crucial = "ABLE" OR D31in_Ability_To_Move_PM2_Non_Crucial = "NOT_USED" OR NOT D27in_Con_PM2_Non_Crucial_Activation) AND D33in_Internal_Trigger_Ability_To_Move_Point_Available AND D15in_Con_Observe_Ability_To_Move]/{Junction0 - ABLE_TO_MOVE}		Option Able to move
Eu.P.6401	Req	[(D29in_Ability_To_Move_PM1_Crucial = "UNABLE" OR D30in_Ability_To_Move_PM1n_Crucial = "UNABLE" OR D31in_Ability_To_Move_PM2_Non_Crucial = "UNABLE" OR D32in_Ability_To_Move_PM2n_Non_Crucial = "UNABLE" OR NOT D33in_Internal_Trigger_Ability_To_Move_Point_Available) AND D15in_Con_Observe_Ability_To_Move]/{Junction0 - UNABLE_TO_MOVE}		Option Able to move

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.6992	Req	[NOT D15in_Con_Observe_Ability_To_Move]/{Junction0 - ABLE_TO_MOVE}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6402	Info	UNABLE_TO_MOVE		Option Able to move
Eu.P.6403	Req	entry/d11out_Observed_Ability_To_Move := "UNABLE_TO_MOVE";{State-internal in UNABLE_TO_MOVE}		Option Able to move
Eu.P.6404	Req	when(D29in_Ability_To_Move_PM1_Crucial = "ABLE" AND (D30in_Ability_To_Move_PM1n_Crucial = "ABLE" OR D30in_Ability_To_Move_PM1n_Crucial = "NOT_USED" OR NOT D26in_Con_PM1n_Crucial_Activation) AND (D32in_Ability_To_Move_PM2n_Non_Crucial = "ABLE" OR D32in_Ability_To_Move_PM2n_Non_Crucial = "NOT_USED" OR NOT D28in_Con_PM2n_Non_Crucial_Activation) AND (D31in_Ability_To_Move_PM2_Non_Crucial = "ABLE" OR D31in_Ability_To_Move_PM2_Non_Crucial = "NOT_USED" OR NOT D27in_Con_PM2_Non_Crucial_Activation) AND D33in_Internal_Trigger_Ability_To_Move_Point_Available)/{UNABLE_TO_MOVE - ABLE_TO_MOVE}		Option Able to move
Eu.P.6991	Info	D15in_Con_Observe_Ability_To_Move		Option Able to move
Eu.P.5925	Info	F_Control_Non4W_PM		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.5926	Info	[Block] F_Control_Point_Machine_Position [Functional Viewpoint - Subsystem Requirements - Functional Entity]  ibd [Block] F_Control_Non4W_PM [Functional Viewpoint - Subsystem Requirements - Functional Entity] «functional entity» F_Control_Non4W_PM d12in_Required_PM_Position : String D35in_PM_Position : String D36in_Con_Drive_Capability : Boolean D37in_Con_Common_Drive : Boolean D40in_Con_Active : Boolean d51in_EST_EfeS_State : String d34out_Drive_Stop : Boolean D38out_Move_Right : Boolean D39out_Move_Left : Boolean		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.5927	Info	D35in_PM_Position	The port D35in_PM_Position provides the Point machine position to the Subsystem - Point. The port D35in_PM_Position refines the InformationFlow Information_No_End_Position, Information_End_Position_Arrived and Information_Unintended_Position.	Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.5928	Info	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)	Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.5929	Info	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).	Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.5930	Info	d12in_Required_PM_Position		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.5932	Info	d34out_Drive_Stop		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.5934	Info	d51in_EST_EfeS_State		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.5936	Info	D37in_Con_Common_Drive	The port D37in_Con_Common_Drive provides a configuration value to the Point for Redrive point functionality. true: Point has Common Drive functionality false: Point has Individual Drive functionality	Option Common Drive
Eu.P.6222	Info	D40in_Con_Active	The port D40in_Con_Active provides configuration values for the Point machine. The Point machine in single Point machine configurations and the first crucial Point machine in multiple Point machine	Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
			configurations is always activated. true: Point machine is activated false: Point machine is not activated	
Eu.P.6223	Info	D36in_Con_Drive_Capability	The port D36in_Con_Drive_Capability provides a configuration value for the Point machine telling the Subsystem - Point whether the Point machine is configured with full drive capability or as Point detector only. true: full drive capability false: Point detector only	Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.5938	Info	F_Control_Non4W_PM - Behaviour		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P

Eu.P.5939	Info	Functional Viewpoint - Subsystem Requirements - Functional Entity STD 2
		 stm [State Machine] F_Control_Non4W_PM - Behaviour [Functional Viewpoint - Subsystem Requirements - Functional Entity STD 2]
		Initial0
		when(d51in_EST_EfeS_State = "NO_OPERATING_VOLTAGE" OR d51in_EST_EfeS_State = "BOOTING" OR d51in_EST_EfeS_State = "FALLBACK_MODE")/
		OPERATING
		Entry/D39out_Move_Left := FALSE; D38out_Move_Right := FALSE; d34out_Drive_Stop := TRUE;
		WAITING
		Initial1
		when(d51in_EST_EfeS_State = "INITIALISING")/
		STOPPED
		Entry/D39out_Move_Left := FALSE; D38out_Move_Right := FALSE; d34out_Drive_Stop := TRUE;
		when(d12in_Required_PM_Position = D35in_PM_Position) [NOT D37in_Con_Common_Drive]/
		when(d12in_Required_PM_Position = "UNCOMMANDED")/
		when(d12in_Required_PM_Position = D35in_PM_Position) [NOT D37in_Con_Common_Drive]/
		when(d12in_Required_PM_Position = "UNCOMMANDED")/
		MOVING_RIGHT
		Entry/D38out_Move_Right := TRUE; d34out_Drive_Stop := FALSE;
		MOVING_LEFT
		Entry/D39out_Move_Left := TRUE; d34out_Drive_Stop := FALSE;
		when(d12in_Required_PM_Position = "LEFT" AND ((d12in_Required_PM_Position <> D35in_PM_Position) OR (d12in_Required_PM_Position = D35in_PM_Position AND D37in_Con_Common_Drive)))[D40in_Con_Active AND D36in_Con_Drive_Capability]/
		when(d12in_Required_PM_Position = "RIGHT" AND ((d12in_Required_PM_Position <> D35in_PM_Position) OR (d12in_Required_PM_Position = D35in_PM_Position AND D37in_Con_Common_Drive)))[D40in_Con_Active AND D36in_Con_Drive_Capability]/
		when(d12in_Required_PM_Position = "RIGHT")[(d12in_Required_PM_Position <> D35in_PM_Position) OR (d12in_Required_PM_Position = D35in_PM_Position AND D37in_Con_Common_Drive)]/D39out_Move_Left := FALSE;
		when(d12in_Required_PM_Position = "LEFT")[(d12in_Required_PM_Position <> D35in_PM_Position) OR (d12in_Required_PM_Position = D35in_PM_Position AND D37in_Con_Common_Drive)]/D38out_Move_Right := FALSE;
				Basic non-4-wire single P
				Basic non-4-wire multiple P
				Basic 4-wire single P
				Basic 4-wire multiple P

Eu.P.5940	Info	Initial0		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.5941	Req	/{Initial0 - OPERATING}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.5942	Info	OPERATING		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.5947	Info	Initial1		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.5948	Req	/{Initial1 - WAITING}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.5955	Info	MOVING_LEFT		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P

Eu.P.5957	Req	entry/D39out_Move_Left := TRUE;
		d34out_Drive_Stop := FALSE;{State-internal in MOVING_LEFT}
				Basic non-4-wire single P
				Basic non-4-wire multiple P
				Basic 4-wire single P
				Basic 4-wire multiple P

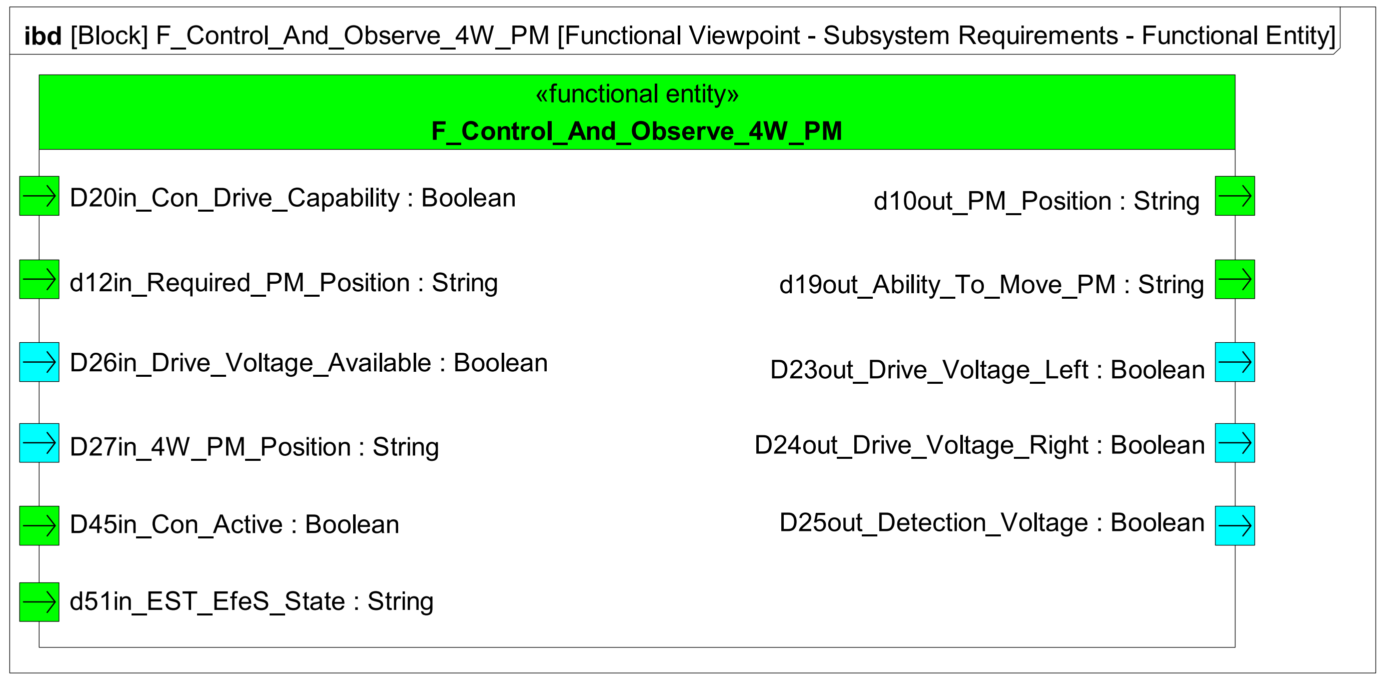
Eu.P.5960	Req	when(d12in_Required_PM_Position = "RIGHT")[(d12in_Required_PM_Position <> D35in_PM_Position) OR (d12in_Required_PM_Position = D35in_PM_Position AND D37in_Con_Common_Drive)]/D39out_Move_Left := FALSE;{MOVING_LEFT - MOVING_RIGHT}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.5961	Req	when(d12in_Required_PM_Position = "UNCOMMANDED")/{MOVING_LEFT - STOPPED}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6301	Req	when(d12in_Required_PM_Position = D35in_PM_Position)[NOT D37in_Con_Common_Drive]/{MOVING_LEFT - STOPPED}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.5965	Info	MOVING_RIGHT		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P

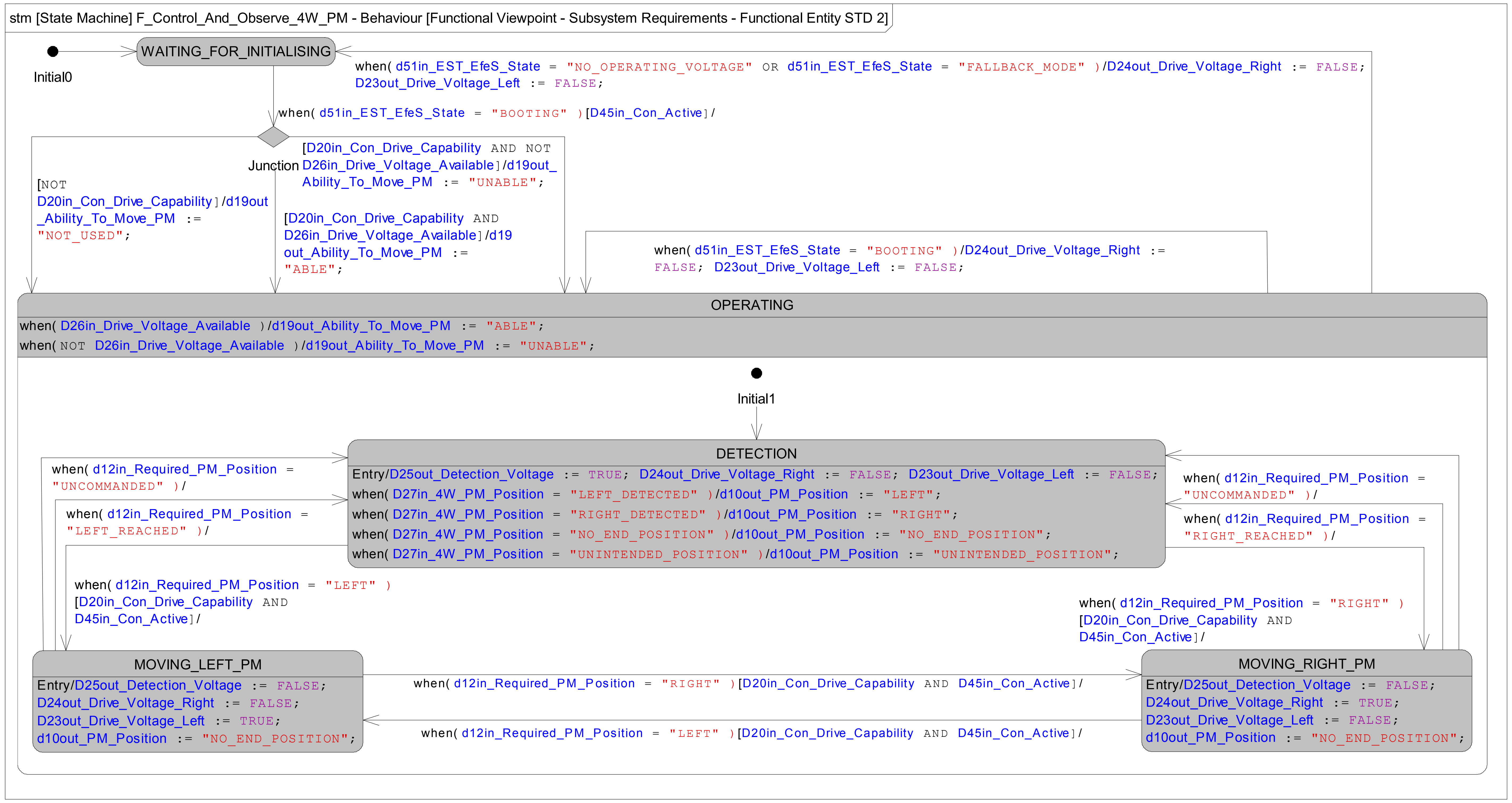
Eu.P.5967	Req	entry/D38out_Move_Right := TRUE;
		d34out_Drive_Stop := FALSE;{State-internal in MOVING_RIGHT}
				Basic non-4-wire single P
				Basic non-4-wire multiple P
				Basic 4-wire single P
				Basic 4-wire multiple P

Eu.P.5970	Req	when(d12in_Required_PM_Position = "LEFT")[(d12in_Required_PM_Position <> D35in_PM_Position) OR (d12in_Required_PM_Position = D35in_PM_Position AND D37in_Con_Common_Drive)]/D38out_Move_Right := FALSE;{MOVING_RIGHT - MOVING_LEFT}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.5971	Req	when(d12in_Required_PM_Position = "UNCOMMANDED")/{MOVING_RIGHT - STOPPED}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6303	Req	when(d12in_Required_PM_Position = D35in_PM_Position)[NOT D37in_Con_Common_Drive]/{MOVING_RIGHT - STOPPED}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.5975	Info	STOPPED		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.5976	Req	when(d12in_Required_PM_Position = "LEFT" AND ((d12in_Required_PM_Position <> D35in_PM_Position) OR (d12in_Required_PM_Position = D35in_PM_Position AND D37in_Con_Common_Drive)))[D40in_Con_Active AND D36in_Con_Drive_Capability]/{STOPPED - MOVING_LEFT}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.5977	Req	when(d12in_Required_PM_Position = "RIGHT" AND ((d12in_Required_PM_Position <> D35in_PM_Position) OR (d12in_Required_PM_Position = D35in_PM_Position AND D37in_Con_Common_Drive)))[D40in_Con_Active AND D36in_Con_Drive_Capability]/{STOPPED - MOVING_RIGHT}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P

Eu.P.6304	Req	entry/D39out_Move_Left := FALSE;
		D38out_Move_Right := FALSE;
		d34out_Drive_Stop := TRUE;{State-internal in STOPPED}
				Basic non-4-wire single P
				Basic non-4-wire multiple P
				Basic 4-wire single P
				Basic 4-wire multiple P

Eu.P.6392	Req	entry/D39out_Move_Left := FALSE; D38out_Move_Right := FALSE; d34out_Drive_Stop := TRUE;{State-internal in OPERATING}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6393	Info	WAITING		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6394	Req	when(d51in_EST_EfeS_State = "INITIALISING")/{WAITING - STOPPED}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.6395	Req	when(d51in_EST_EfeS_State = "NO_OPERATING_VOLTAGE" OR d51in_EST_EfeS_State = "BOOTING" OR d51in_EST_EfeS_State = "FALLBACK_MODE")/{OPERATING - OPERATING}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.5810	Info	F_Control_And_Observe_4W_PM		Basic 4-wire single P Basic 4-wire multiple P
Eu.P.5811	Info	[Block] F_Control_And_Observe_4W_PM [Technical Viewpoint - Subsystem Requirements - Technical Functional Entity]  ibd [Block] F_Control_And_Observe_4W_PM [Functional Viewpoint - Subsystem Requirements - Functional Entity] «functional entity» F_Control_And_Observe_4W_PM D20in_Con_Drive_Capability : Boolean d12in_Required_PM_Position : String D26in_Drive_Voltage_Available : Boolean D27in_4W_PM_Position : String D45in_Con_Active : Boolean d51in_EST_EfeS_State : String d10out_PM_Position : String d19out_Ability_To_Move_PM : String D23out_Drive_Voltage_Left : Boolean D24out_Drive_Voltage_Right : Boolean D25out_Detection_Voltage : Boolean		Basic 4-wire single P Basic 4-wire multiple P
Eu.P.5812	Info	D20in_Con_Drive_Capability	The port D20in_Con_Drive_Capability provides configuration values for the Point machine. true: Point machine has drive capability false: Point machine has no drive capability	Basic 4-wire single P Basic 4-wire multiple P
Eu.P.5813	Info	d19out_Ability_To_Move_PM		Basic 4-wire single P Basic 4-wire multiple P
Eu.P.5814	Info	d10out_PM_Position		Basic 4-wire single P Basic 4-wire multiple P
Eu.P.5817	Info	D27in_4W_PM_Position	The port D27in_4W_PM_Position refines the Informationflow Information_No_End_Position, Information_End_Position_Reached, Information_End_Position_Detected and Information_Unintended_Position and is a representative for the position of an undefined number of Point machines, in case they are implemented with a 4-wire interface. Note: The same 4-wire input pattern doesn't always represent the same Informationflow. The interpretation of all patterns is fully defined in Eu.P.6797.	Basic 4-wire single P Basic 4-wire multiple P
Eu.P.5818	Info	D26in_Drive_Voltage_Available	The port D26in_Drive_Voltage_Available refines the Informationflow Information_Ability_To_Move_Point and contains the information if Drive voltage can be provided to the Point machine via the 4-wire interface. The following values are valid: “True” (driving possible), “False” (driving not possible). Note: The value “True” does not mean that the Drive voltage is currently being applied to the Point machine, it only refers to its availability to the subsystem.	Basic 4-wire single P Basic 4-wire multiple P
Eu.P.5819	Info	D25out_Detection_Voltage	The port D25out_Detection_Voltage refines the Informationflow Detection_Voltage and contains the information if Detection voltage is provided to the Point machine via the 4-wire interface. The following values are valid: "True" (detection), "False" (no detection)	Basic 4-wire single P Basic 4-wire multiple P
Eu.P.5820	Info	D24out_Drive_Voltage_Right	The port D24out_Drive_Voltage_Right refines the Informationflow Moving and contains the information if Drive voltage for driving right is provided to the Point machine via the 4-wire interface. The following values are valid: "True" (driving right), "False" (not driving right)	Basic 4-wire single P Basic 4-wire multiple P
Eu.P.5821	Info	D23out_Drive_Voltage_Left	The port D23out_Drive_Voltage_Left refines the Informationflow Moving and contains the information if Drive voltage for driving left is	Basic 4-wire single P Basic 4-wire multiple P

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
			provided to the Point machine via the 4-wire interface. The following values are valid: "True" (driving left), "False" (not driving left)	
Eu.P.6209	Info	d12in_Required_PM_Position		Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6211	Info	D45in_Con_Active	The port D45in_Con_Active provides configuration values for the Point machine. true: Point machine is activated false: Point machine is not activated	Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6212	Info	d51in_EST_EfeS_State		Basic 4-wire single P Basic 4-wire multiple P
Eu.P.5822	Info	F_Control_And_Observe_4W_PM - Behaviour		Basic 4-wire single P Basic 4-wire multiple P
Eu.P.5823	Info	Functional Viewpoint - Subsystem Requirements - Functional Entity STD 2 		Basic 4-wire single P Basic 4-wire multiple P
Eu.P.5826	Info	OPERATING		Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6221	Req	when(d51in_EST_EfeS_State = "NO_OPERATING_VOLTAGE" OR d51in_EST_EfeS_State = "FALLBACK_MODE")/D24out_Drive_Voltage_Right := FALSE; D23out_Drive_Voltage_Left := FALSE;{OPERATING - WAITING_FOR_INITIALISING}		Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7078	Req	when(d51in_EST_EfeS_State = "BOOTING")/D24out_Drive_Voltage_Right := FALSE; D23out_Drive_Voltage_Left := FALSE;{OPERATING - OPERATING}		Basic 4-wire single P Basic 4-wire multiple P
Eu.P.5915	Info	DETECTION		Basic 4-wire single P Basic 4-wire multiple P
Eu.P.5918	Req	entry/D25out_Detection_Voltage := TRUE; D24out_Drive_Voltage_Right := FALSE; D23out_Drive_Voltage_Left := FALSE;{State-internal in DETECTION}		Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6935	Req	when(D27in_4W_PM_Position = "LEFT_DETECTED")/d10out_PM_Position := "LEFT";{State-internal in DETECTION}		Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6936	Req	when(D27in_4W_PM_Position = "NO_END_POSITION")/d10out_PM_Position := "NO_END_POSITION";{State-internal in DETECTION}		Basic 4-wire single P Basic 4-wire multiple P

stm [State Machine] F_Control_And_Observe_4W_PM - Behaviour [Functional Viewpoint - Subsystem Requirements - Functional Entity STD 2]
WAITING_FOR_INITIALISING
Initial0
when(d51in_EST_EfeS_State = "NO_OPERATING_VOLTAGE" OR d51in_EST_EfeS_State = "FALLBACK_MODE")/D24out_Drive_Voltage_Right := FALSE; D23out_Drive_Voltage_Left := FALSE;
when(d51in_EST_EfeS_State = "BOOTING")[D45in_Con_Active]/
Junction
[D20in_Con_Drive_Capability AND NOT D26in_Drive_Voltage_Available]/d19out_Ability_To_Move_PM := "UNABLE";
[NOT D20in_Con_Drive_Capability]/d19out_Ability_To_Move_PM := "NOT_USED";
[D20in_Con_Drive_Capability AND D26in_Drive_Voltage_Available]/d19out_Ability_To_Move_PM := "ABLE";
when(d51in_EST_EfeS_State = "BOOTING")/D24out_Drive_Voltage_Right := FALSE; D23out_Drive_Voltage_Left := FALSE;
OPERATING
when(D26in_Drive_Voltage_Available)/d19out_Ability_To_Move_PM := "ABLE";
when(NOT D26in_Drive_Voltage_Available)/d19out_Ability_To_Move_PM := "UNABLE";
Initial1
DETECTION
Entry/D25out_Detection_Voltage := TRUE; D24out_Drive_Voltage_Right := FALSE; D23out_Drive_Voltage_Left := FALSE;
when(D27in_4W_PM_Position = "LEFT_DETECTED")/d10out_PM_Position := "LEFT";
when(D27in_4W_PM_Position = "RIGHT_DETECTED")/d10out_PM_Position := "RIGHT";
when(D27in_4W_PM_Position = "NO_END_POSITION")/d10out_PM_Position := "NO_END_POSITION";
when(D27in_4W_PM_Position = "UNINTENDED_POSITION")/d10out_PM_Position := "UNINTENDED_POSITION";
when(d12in_Required_PM_Position = "UNCOMMANDED")/
when(d12in_Required_PM_Position = "LEFT_REACHED")/
when(d12in_Required_PM_Position = "UNCOMMANDED")/
when(d12in_Required_PM_Position = "RIGHT_REACHED")/
when(d12in_Required_PM_Position = "LEFT")[D20in_Con_Drive_Capability AND D45in_Con_Active]/
when(d12in_Required_PM_Position = "RIGHT")[D20in_Con_Drive_Capability AND D45in_Con_Active]/
MOVING_LEFT_PM
Entry/D25out_Detection_Voltage := FALSE; D24out_Drive_Voltage_Right := FALSE; D23out_Drive_Voltage_Left := TRUE; d10out_PM_Position := "NO_END_POSITION";
when(d12in_Required_PM_Position = "RIGHT")[D20in_Con_Drive_Capability AND D45in_Con_Active]/
when(d12in_Required_PM_Position = "LEFT")[D20in_Con_Drive_Capability AND D45in_Con_Active]/
MOVING_RIGHT_PM
Entry/D25out_Detection_Voltage := FALSE; D24out_Drive_Voltage_Right := TRUE; D23out_Drive_Voltage_Left := FALSE; d10out_PM_Position := "NO_END_POSITION";

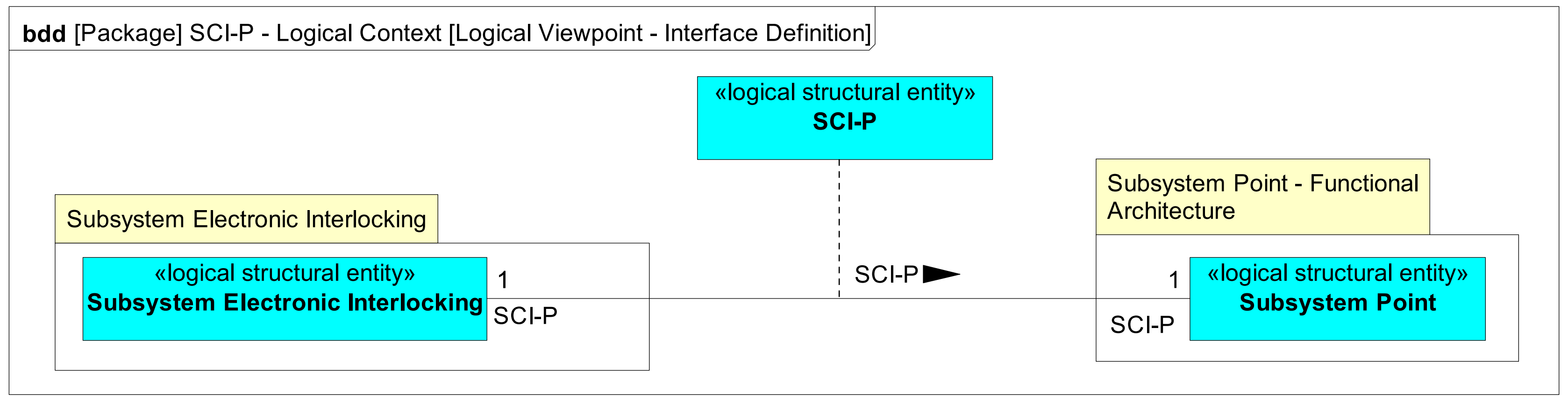
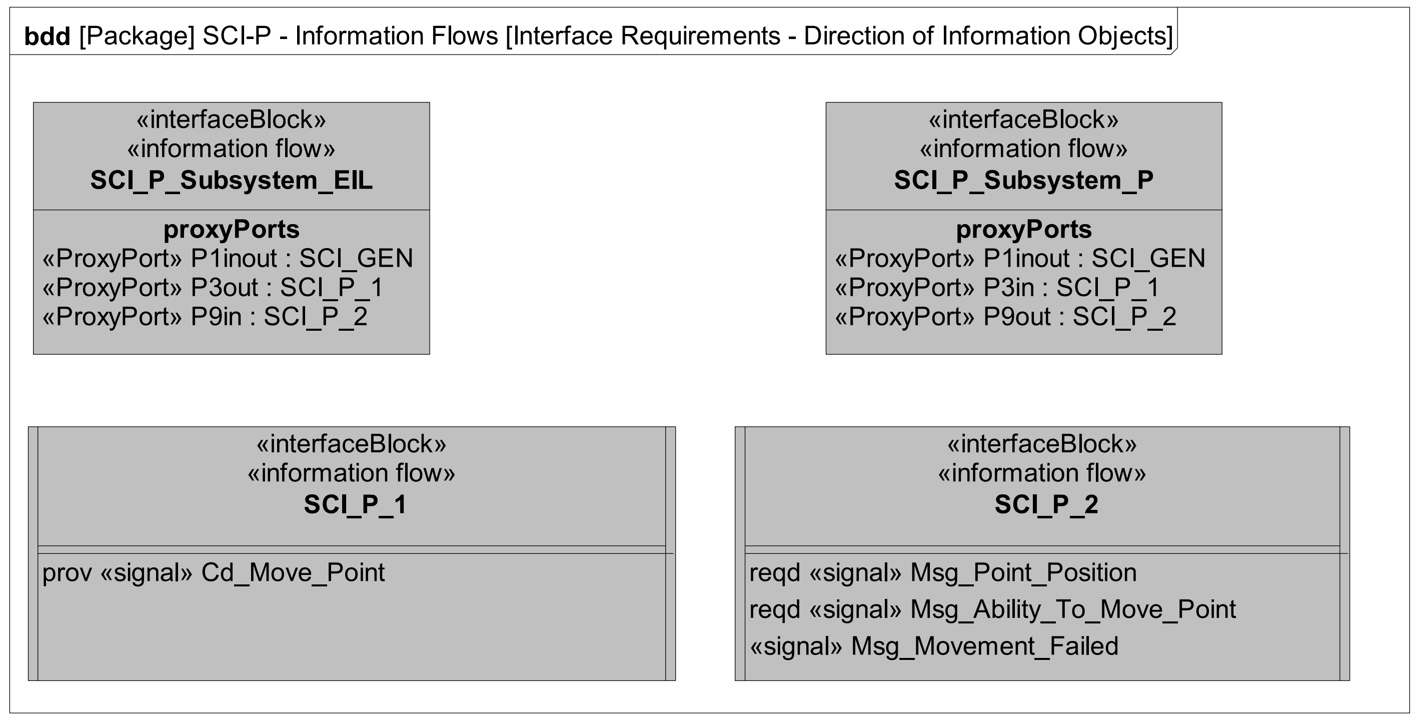
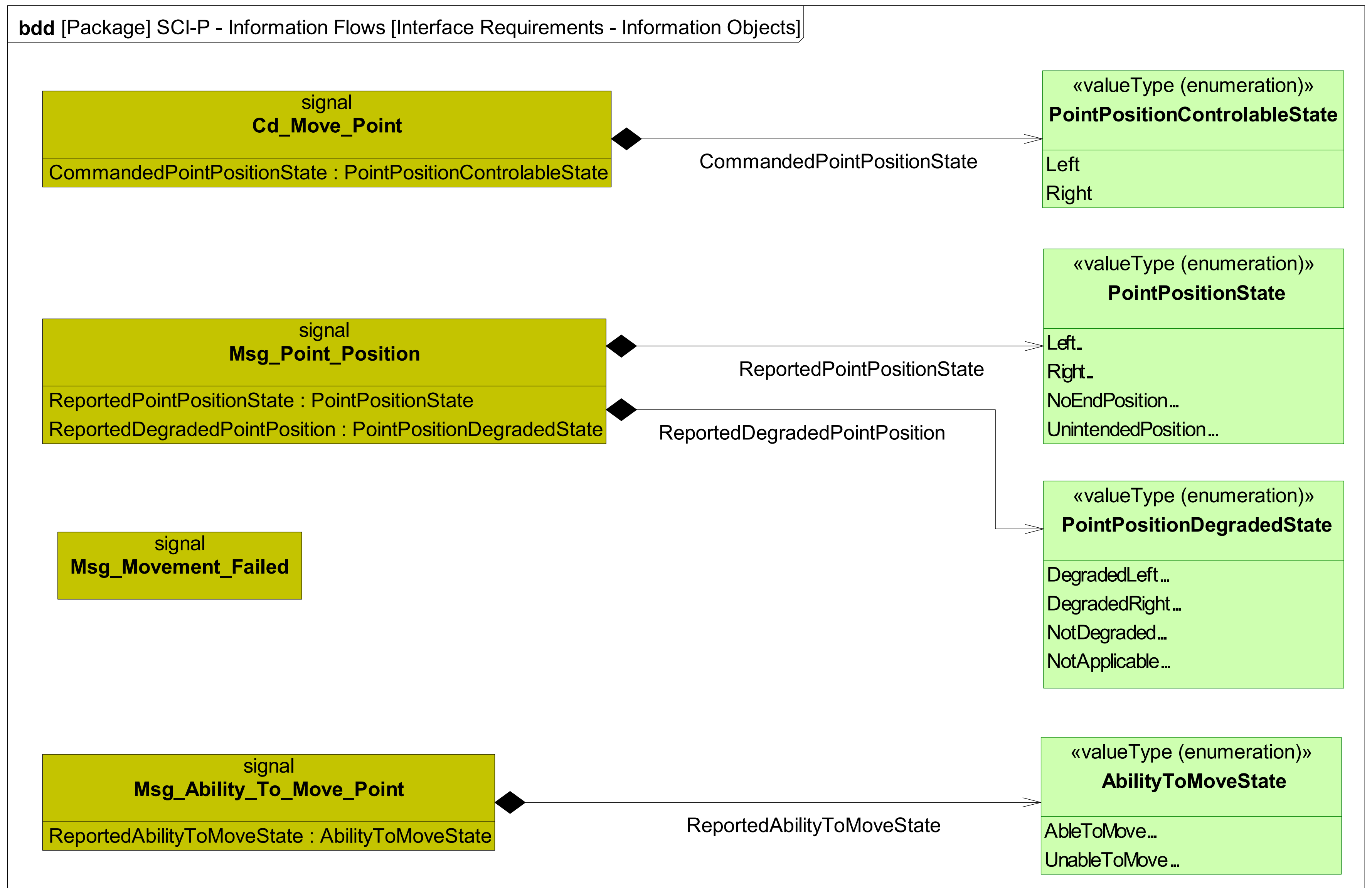
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.6937	Req	when(D27in_4W_PM_Position = "RIGHT_DETECTED")/d10out_PM_Position := "RIGHT";{State-internal in DETECTION}		Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6938	Req	when(D27in_4W_PM_Position = "UNINTENDED_POSITION")/d10out_PM_Position := "UNINTENDED_POSITION";{State-internal in DETECTION}		Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6939	Req	when(d12in_Required_PM_Position = "LEFT")[D20in_Con_Drive_Capability AND D45in_Con_Active]/{DETECTION - MOVING_LEFT_PM}		Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6940	Req	when(d12in_Required_PM_Position = "RIGHT")[D20in_Con_Drive_Capability AND D45in_Con_Active]/{DETECTION - MOVING_RIGHT_PM}		Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6941	Info	Initial1		Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6942	Req	/{Initial1 - DETECTION}		Basic 4-wire single P Basic 4-wire multiple P
Eu.P.5888	Info	MOVING_LEFT_PM		Basic 4-wire single P Basic 4-wire multiple P

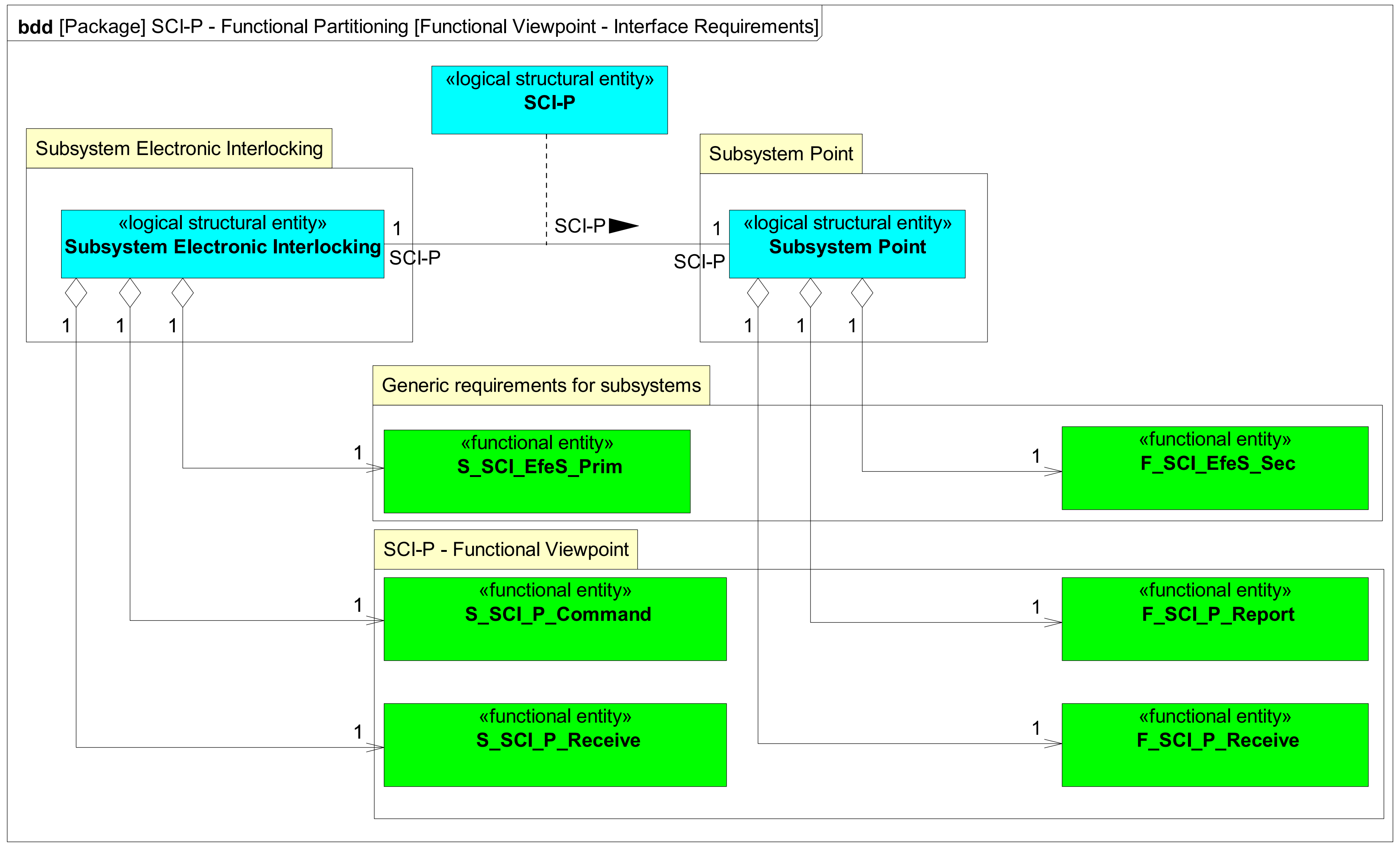
Eu.P.5890	Req	entry/D25out_Detection_Voltage := FALSE;
		D24out_Drive_Voltage_Right := FALSE;
		D23out_Drive_Voltage_Left := TRUE;
		d10out_PM_Position := "NO_END_POSITION";{State-internal in MOVING_LEFT_PM}
				Basic 4-wire single P
				Basic 4-wire multiple P

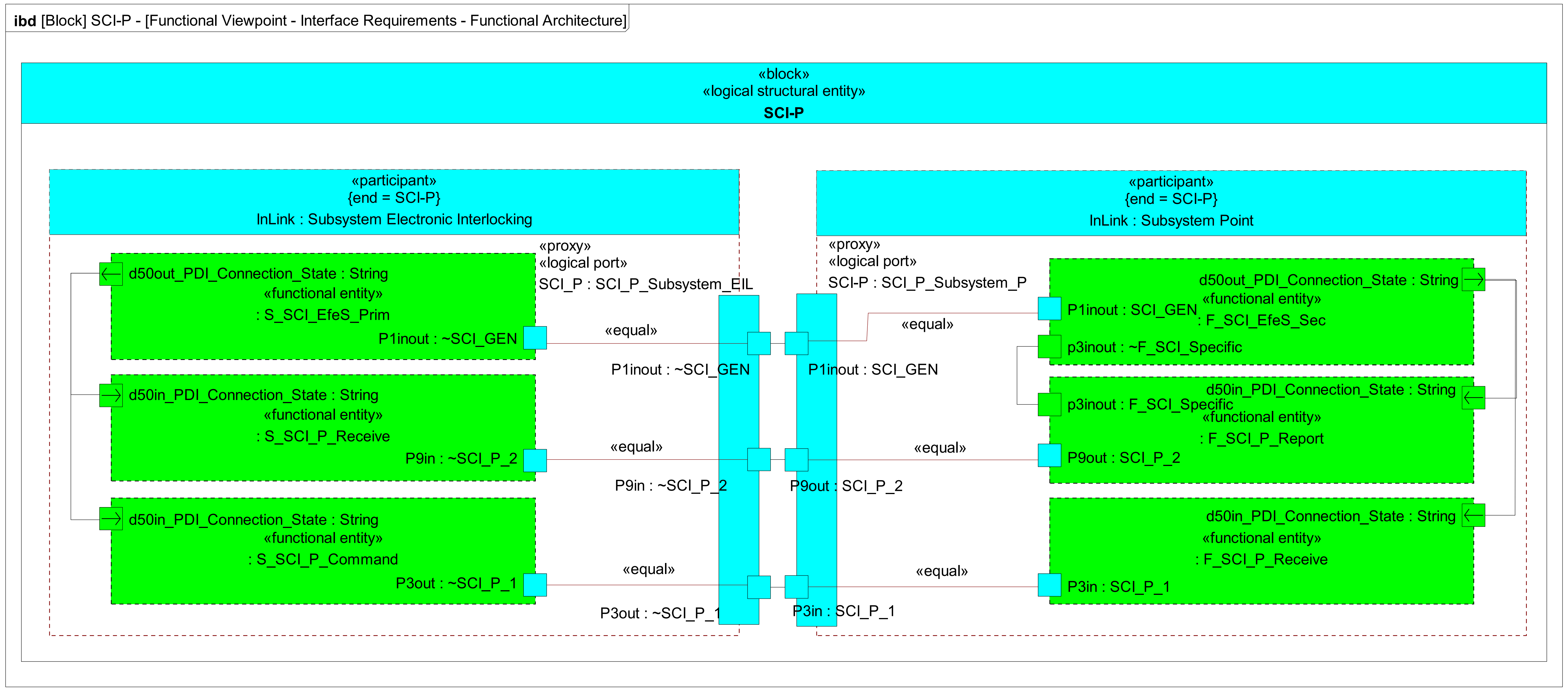
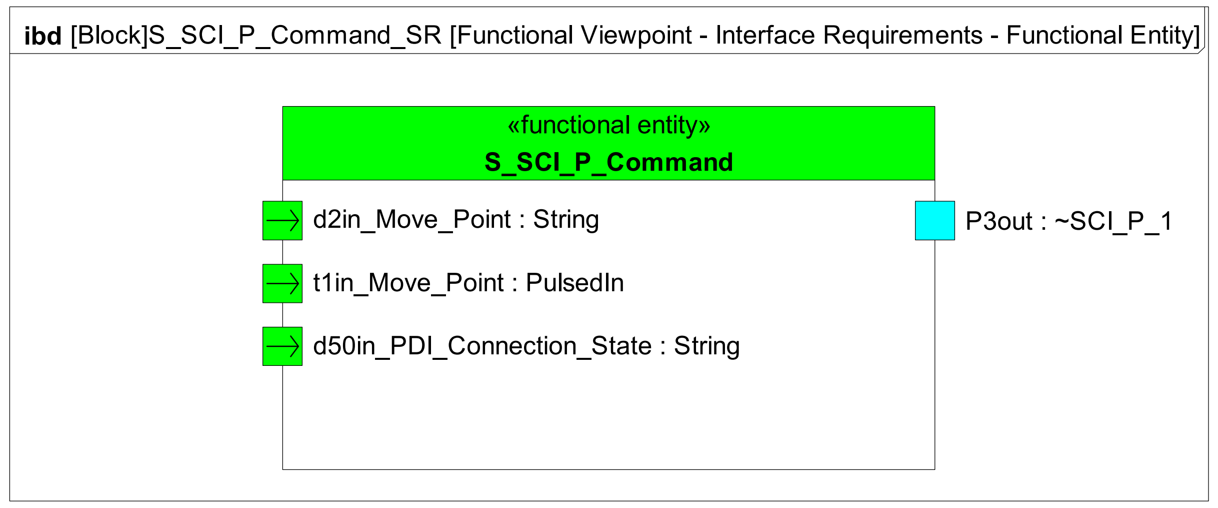
Eu.P.5877	Req	when(d12in_Required_PM_Position = "RIGHT")[D20in_Con_Drive_Capability AND D45in_Con_Active]/{MOVING_LEFT_PM - MOVING_RIGHT_PM}		Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6943	Req	when(d12in_Required_PM_Position = "LEFT_REACHED")/{MOVING_LEFT_PM - DETECTION}		Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6944	Req	when(d12in_Required_PM_Position = "UNCOMMANDED")/{MOVING_LEFT_PM - DETECTION}		Basic 4-wire single P Basic 4-wire multiple P
Eu.P.5905	Info	MOVING_RIGHT_PM		Basic 4-wire single P Basic 4-wire multiple P

Eu.P.5907	Req	entry/D25out_Detection_Voltage := FALSE;
		D24out_Drive_Voltage_Right := TRUE;
		D23out_Drive_Voltage_Left := FALSE;
		d10out_PM_Position := "NO_END_POSITION";{State-internal in MOVING_RIGHT_PM}
				Basic 4-wire single P
				Basic 4-wire multiple P

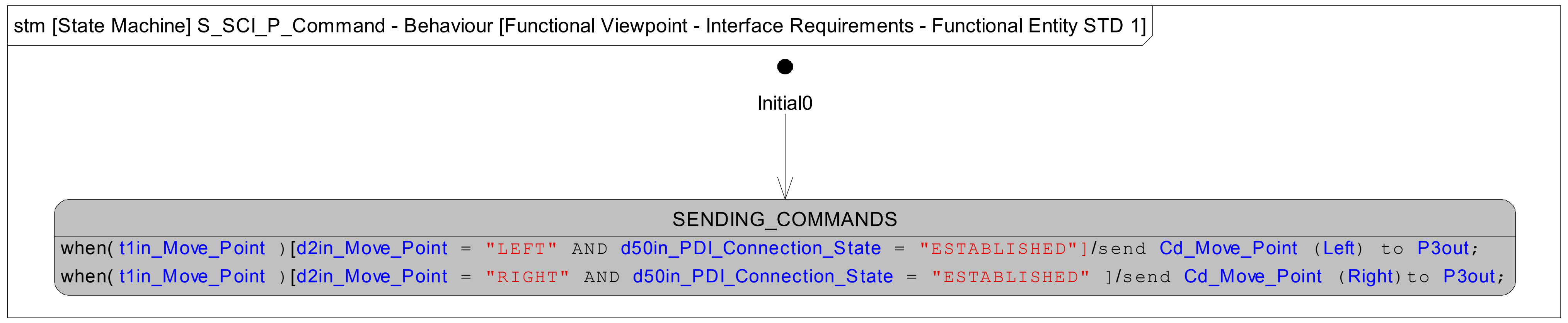
Eu.P.5894	Req	when(d12in_Required_PM_Position = "LEFT")[D20in_Con_Drive_Capability AND D45in_Con_Active]/{MOVING_RIGHT_PM - MOVING_LEFT_PM}		Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6945	Req	when(d12in_Required_PM_Position = "RIGHT_REACHED")/{MOVING_RIGHT_PM - DETECTION}		Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6946	Req	when(d12in_Required_PM_Position = "UNCOMMANDED")/{MOVING_RIGHT_PM - DETECTION}		Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7134	Req	when(D26in_Drive_Voltage_Available)/d19out_Ability_To_Move_PM := "ABLE";{State-internal in OPERATING}		Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7135	Req	when(NOT D26in_Drive_Voltage_Available)/d19out_Ability_To_Move_PM := "UNABLE";{State-internal in OPERATING}		Basic 4-wire single P Basic 4-wire multiple P
Eu.P.5909	Info	Initial0		Basic 4-wire single P Basic 4-wire multiple P
Eu.P.5910	Req	/{Initial0 - WAITING_FOR_INITIALISING}		Basic 4-wire single P Basic 4-wire multiple P
Eu.P.5911	Info	WAITING_FOR_INITIALISING		Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6220	Req	when(d51in_EST_EfeS_State = "BOOTING")[D45in_Con_Active]/{WAITING_FOR_INITIALISING - Junction}		Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6487	Info	Junction		Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6488	Req	[D20in_Con_Drive_Capability AND D26in_Drive_Voltage_Available]/d19out_Ability_To_Move_PM := "ABLE";{Junction - OPERATING}		Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6489	Req	[NOT D20in_Con_Drive_Capability]/d19out_Ability_To_Move_PM := "NOT_USED";{Junction - OPERATING}		Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6490	Req	[D20in_Con_Drive_Capability AND NOT D26in_Drive_Voltage_Available]/d19out_Ability_To_Move_PM := "UNABLE";{Junction - OPERATING}		Basic 4-wire single P Basic 4-wire multiple P
Eu.P.889	Head	**3.4 Subsystem Point - Interfaces**		
Eu.P.913	Head	**3.4.1 SCI-P (Subsystem - Electronic Interlocking)**		
Eu.P.6154	Head	**3.4.1.1 SCI-P - Logical Viewpoint**		
Eu.P.6281	Head	**3.4.1.1.1 SCI-P - Logical Context**		

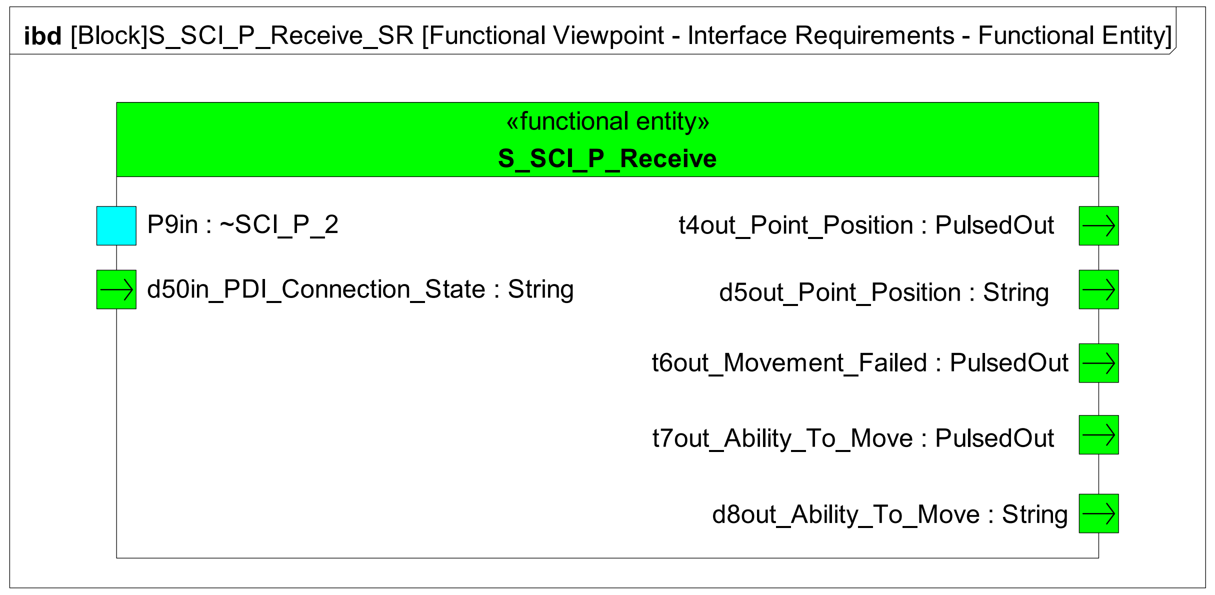
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.3287	Info	[Package] SCI-P - Logical Context [Logical Viewpoint - Interface Definition]  bdd [Package] SCI-P - Logical Context [Logical Viewpoint - Interface Definition] «logical structural entity» SCI-P Subsystem Electronic Interlocking «logical structural entity» Subsystem Electronic Interlocking 1 SCI-P SCI-P Subsystem Point - Functional Architecture 1 SCI-P «logical structural entity» Subsystem Point		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6041	Head	**3.4.1.2 SCI-P - Information Flows**		
Eu.P.3064	Info	The generic commands and messages through the SCI_P_Subsystem_EIL are specified in Eu.Doc.119.		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6042	Info	[Package] SCI-P - Information Flows [Interface Requirements - Direction of Information Objects]  bdd [Package] SCI-P - Information Flows [Interface Requirements - Direction of Information Objects] «interfaceBlock» «information flow» SCI_P_Subsystem_EIL proxyPorts «ProxyPort» P1inout : SCI_GEN «ProxyPort» P3out : SCI_P_1 «ProxyPort» P9in : SCI_P_2 «interfaceBlock» «information flow» SCI_P_Subsystem_P proxyPorts «ProxyPort» P1inout : SCI_GEN «ProxyPort» P3in : SCI_P_1 «ProxyPort» P9out : SCI_P_2 «interfaceBlock» «information flow» SCI_P_1 prov «signal» Cd_Move_Point «interfaceBlock» «information flow» SCI_P_2 reqd «signal» Msg_Point_Position reqd «signal» Msg_Ability_To_Move_Point «signal» Msg_Movement_Failed		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6044	Info	[Package] SCI-P - Information Flows [Interface Requirements - Information Objects]  bdd [Package] SCI-P - Information Flows [Interface Requirements - Information Objects] signal Cd_Move_Point CommandedPointPositionState : PointPositionControlableState CommandedPointPositionState «valueType (enumeration)» PointPositionControlableState Left Right signal Msg_Point_Position ReportedPointPositionState : PointPositionState ReportedDegradedPointPosition : PointPositionDegradedState ReportedPointPositionState ReportedDegradedPointPosition «valueType (enumeration)» PointPositionState Left… Right… NoEndPosition… UnintendedPosition… «valueType (enumeration)» PointPositionDegradedState DegradedLeft… DegradedRight… NotDegraded… NotApplicable… signal Msg_Movement_Failed signal Msg_Ability_To_Move_Point ReportedAbilityToMoveState : AbilityToMoveState ReportedAbilityToMoveState «valueType (enumeration)» AbilityToMoveState AbleToMove… UnableToMove…		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.6183	Info	Cd_Move_Point	Command (Cd) from Subsystem - Electronic Interlocking to Subsystem - Point to move the Point into the commanded position.	Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6185	Info	Msg_Ability_To_Move_Point	Message (Msg) from Subsystem - Point to Subsystem - Electronic Interlocking about the current Ability to move point.	Option Able to move
Eu.P.6187	Info	Msg_Point_Position	Message (Msg) from Subsystem - Point to Subsystem - Electronic Interlocking about the current Point position.	Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6190	Info	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 timer 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.	Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6045	Head	**3.4.1.3 SCI-P - Functional Viewpoint**		
Eu.P.6280	Head	**3.4.1.3.1 SCI-P - Functional Partitioning**		
Eu.P.6155	Info	[Package] SCI-P - Functional Partitioning [Functional Viewpoint - Interface Requirements]  bdd [Package] SCI-P - Functional Partitioning [Functional Viewpoint - Interface Requirements] «logical structural entity» SCI-P Subsystem Electronic Interlocking «logical structural entity» Subsystem Electronic Interlocking 1 SCI-P SCI-P Subsystem Point 1 SCI-P «logical structural entity» Subsystem Point 1 1 1 1 1 1 Generic requirements for subsystems 1 «functional entity» S_SCI_EfeS_Prim 1 «functional entity» F_SCI_EfeS_Sec SCI-P - Functional Viewpoint 1 «functional entity» S_SCI_P_Command 1 «functional entity» F_SCI_P_Report 1 «functional entity» S_SCI_P_Receive 1 «functional entity» F_SCI_P_Receive		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6244	Head	**3.4.1.3.2 SCI-P - Functional Architecture**		
Eu.P.3279	Info	SCI-P		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P

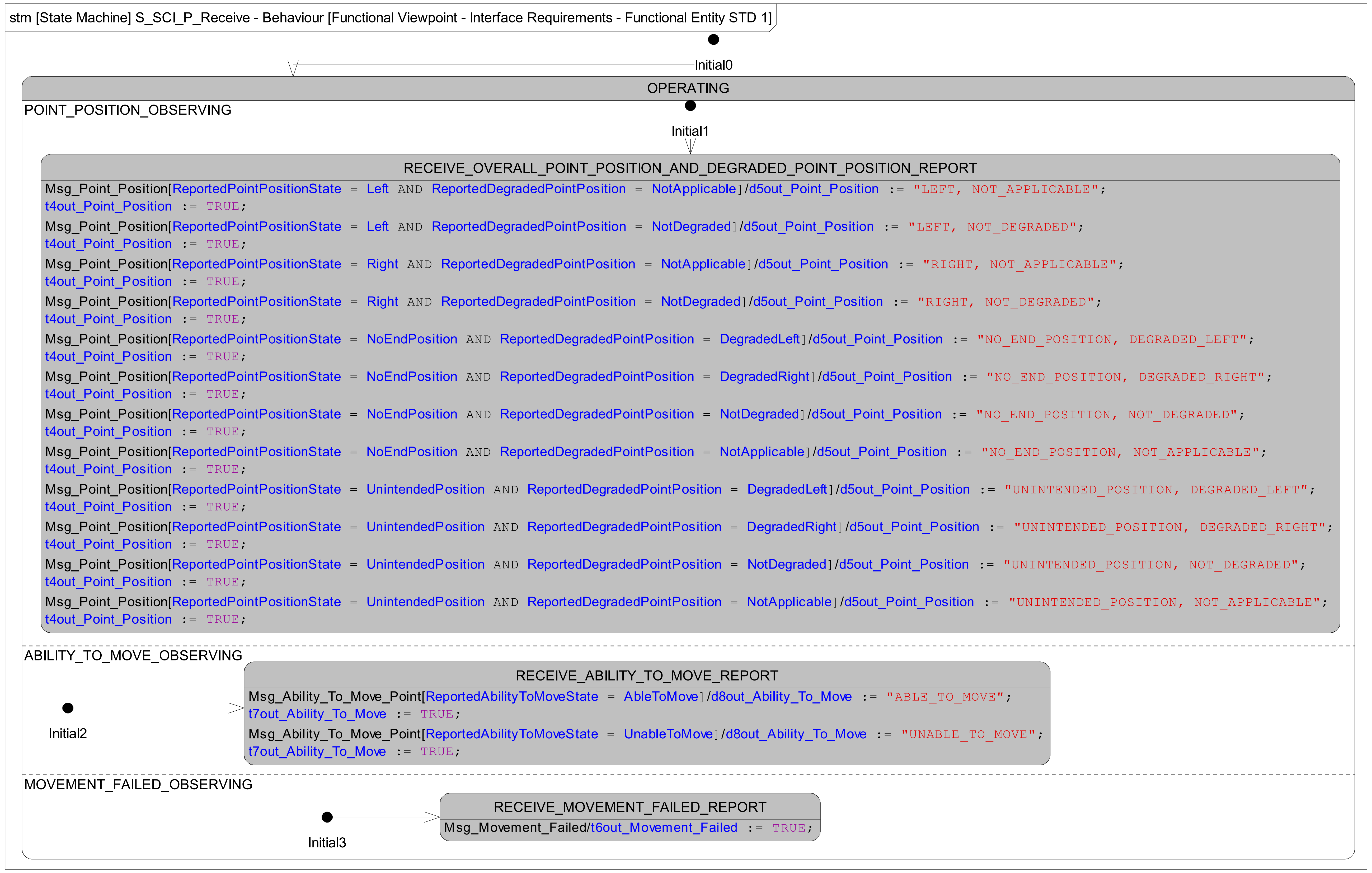
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.3286	Info	[Block] SCI-P - [Functional Viewpoint - Interface Requirements - Functional Architecture]  ibd [Block] SCI-P - [Functional Viewpoint - Interface Requirements - Functional Architecture] «block» «logical structural entity» SCI-P «participant» {end = SCI-P} InLink : Subsystem Electronic Interlocking «participant» {end = SCI-P} InLink : Subsystem Point d50out_PDI_Connection_State : String «functional entity» : S_SCI_EfeS_Prim P1inout : ~SCI_GEN «proxy» «logical port» SCI_P : SCI_P_Subsystem_EIL «equal» P1inout : ~SCI_GEN d50in_PDI_Connection_State : String «functional entity» : S_SCI_P_Receive P9in : ~SCI_P_2 «equal» P9in : ~SCI_P_2 d50in_PDI_Connection_State : String «functional entity» : S_SCI_P_Command P3out : ~SCI_P_1 «equal» P3out : ~SCI_P_1 «proxy» «logical port» SCI-P : SCI_P_Subsystem_P «equal» P1inout : SCI_GEN P1inout : SCI_GEN d50out_PDI_Connection_State : String «functional entity» : F_SCI_EfeS_Sec p3inout : ~F_SCI_Specific p3inout : F_SCI_Specific d50in_PDI_Connection_State : String «functional entity» : F_SCI_P_Report P9out : SCI_P_2 «equal» P9out : SCI_P_2 d50in_PDI_Connection_State : String «functional entity» : F_SCI_P_Receive P3in : SCI_P_1 «equal» P3in : SCI_P_1		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6245	Head	**3.4.1.3.3 SCI-P - Functional Entities**		
Eu.P.4704	Info	S_SCI_P_Command		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.4711	Info	[Block] S_SCI_P_Command_SR [Functional Viewpoint - Interface Requirements - Functional Entity]  ibd [Block]S_SCI_P_Command_SR [Functional Viewpoint - Interface Requirements - Functional Entity] «functional entity» S_SCI_P_Command d2in_Move_Point : String t1in_Move_Point : PulsedIn d50in_PDI_Connection_State : String P3out : ~SCI_P_1		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.4707	Info	d2in_Move_Point		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.4706	Info	d50in_PDI_Connection_State		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6153	Info	t1in_Move_Point		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6477	Info	P3out	The port P3out exchanges information objects according to SCI_P_1.	Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.4712	Info	S_SCI_P_Command - Behaviour		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P

Eu.P.4729	Info	Functional Viewpoint - Interface Requirements - Functional Entity STD 1
		
		 stm [State Machine] S_SCI_P_Command - Behaviour [Functional Viewpoint - Interface Requirements - Functional Entity STD 1]
		Initial0
		SENDING_COMMANDS
		when(t1in_Move_Point)[d2in_Move_Point = "LEFT" AND d50in_PDI_Connection_State = "ESTABLISHED"]/send Cd_Move_Point (Left) to P3out;
		when(t1in_Move_Point)[d2in_Move_Point = "RIGHT" AND d50in_PDI_Connection_State = "ESTABLISHED"]/send Cd_Move_Point (Right) to P3out;
				Basic non-4-wire single P
				Basic non-4-wire multiple P
				Basic 4-wire single P
				Basic 4-wire multiple P

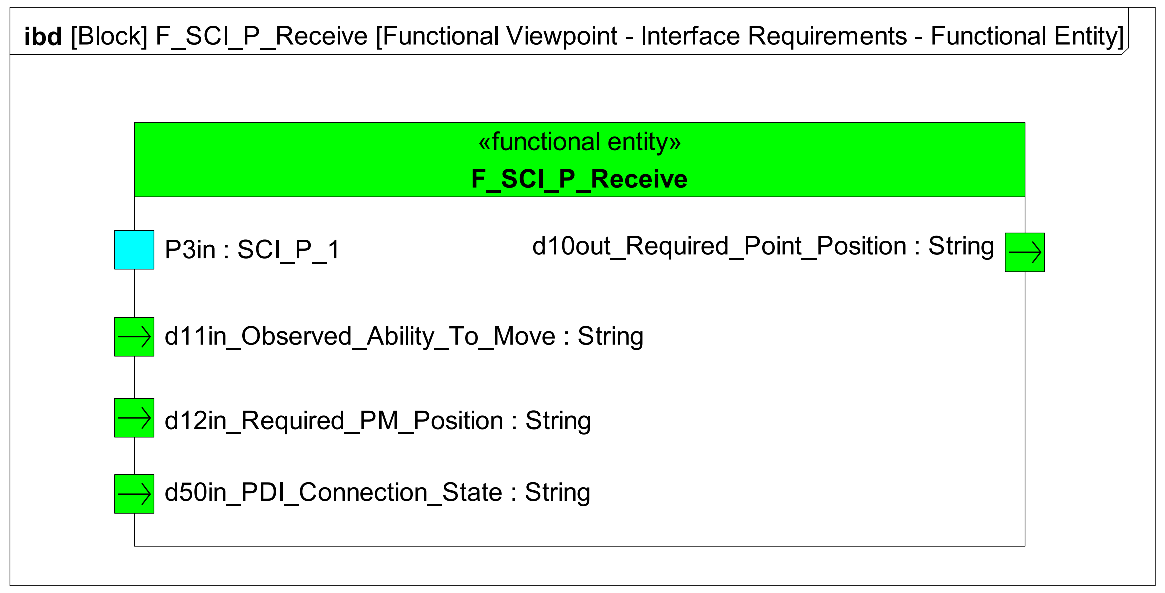
Eu.P.4713	Info	Initial0		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.4714	Req	/{Initial0 - SENDING_COMMANDS}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6131	Info	SENDING_COMMANDS		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6132	Req	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}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6133	Req	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}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7034	Info	S_SCI_P_Receive		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7035	Info	[Block]S_SCI_P_Receive_SR [Functional Viewpoint - Interface Requirements - Functional Entity]  ibd [Block]S_SCI_P_Receive_SR [Functional Viewpoint - Interface Requirements - Functional Entity] «functional entity» S_SCI_P_Receive P9in : ~SCI_P_2 d50in_PDI_Connection_State : String t4out_Point_Position : PulsedOut d5out_Point_Position : String t6out_Movement_Failed : PulsedOut t7out_Ability_To_Move : PulsedOut d8out_Ability_To_Move : String		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7036	Info	d50in_PDI_Connection_State		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7037	Info	d5out_Point_Position		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7038	Info	d8out_Ability_To_Move		Option Able to move

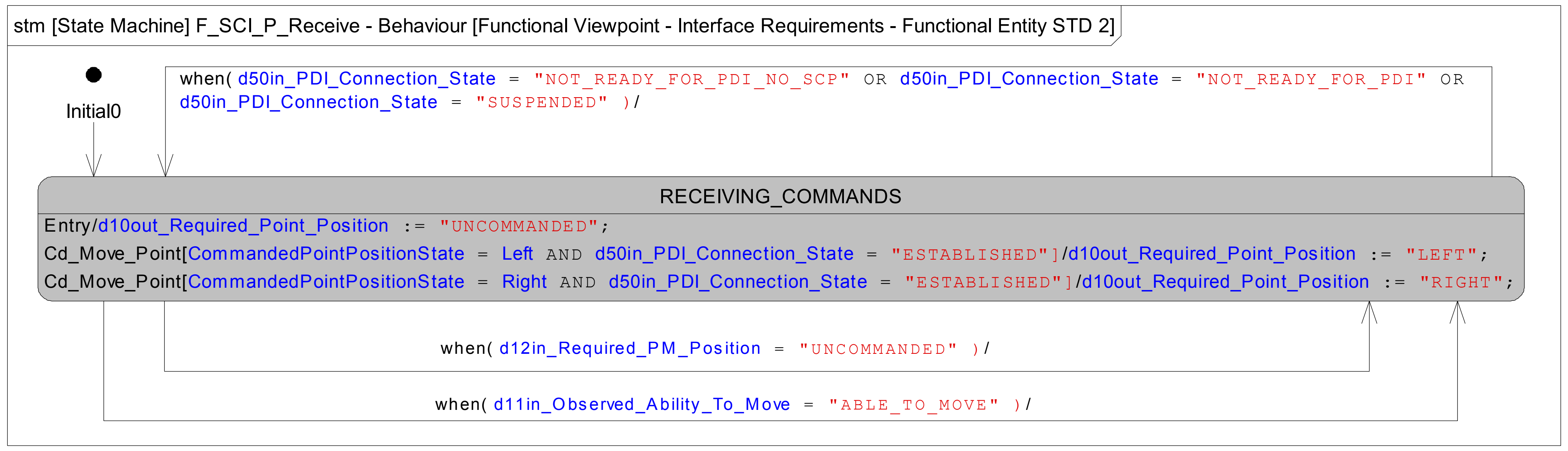
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.7039	Info	P9in	The port P9in exchanges information objects according to SCI_P_2.	Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7072	Info	t4out_Point_Position		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7073	Info	t6out_Movement_Failed		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7074	Info	t7out_Ability_To_Move		Option Able to move
Eu.P.7040	Info	S_SCI_P_Receive - Behaviour		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P

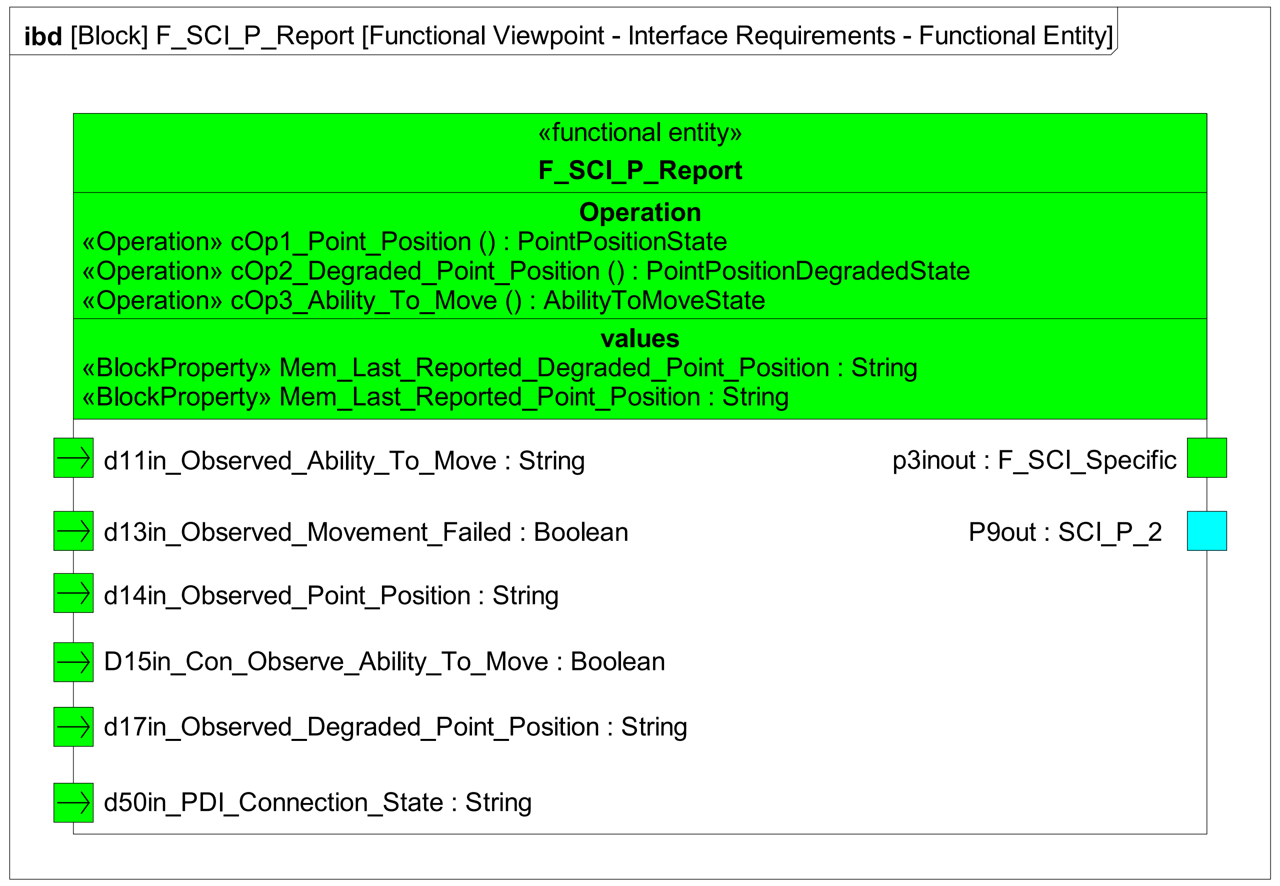
Eu.P.7041	Info	Functional Viewpoint - Interface Requirements - Functional Entity STD 1
		 stm [State Machine] S_SCI_P_Receive - Behaviour [Functional Viewpoint - Interface Requirements - Functional Entity STD 1]
		Initial0
		OPERATING
		POINT_POSITION_OBSERVING
		Initial1
		RECEIVE_OVERALL_POINT_POSITION_AND_DEGRADED_POINT_POSITION_REPORT
		Msg_Point_Position[ReportedPointPositionState = Left AND ReportedDegradedPointPosition = NotApplicable]/d5out_Point_Position := "LEFT, NOT_APPLICABLE"; t4out_Point_Position := TRUE;
		Msg_Point_Position[ReportedPointPositionState = Left AND ReportedDegradedPointPosition = NotDegraded]/d5out_Point_Position := "LEFT, NOT_DEGRADED"; t4out_Point_Position := TRUE;
		Msg_Point_Position[ReportedPointPositionState = Right AND ReportedDegradedPointPosition = NotApplicable]/d5out_Point_Position := "RIGHT, NOT_APPLICABLE"; t4out_Point_Position := TRUE;
		Msg_Point_Position[ReportedPointPositionState = Right AND ReportedDegradedPointPosition = NotDegraded]/d5out_Point_Position := "RIGHT, NOT_DEGRADED"; t4out_Point_Position := TRUE;
		Msg_Point_Position[ReportedPointPositionState = NoEndPosition AND ReportedDegradedPointPosition = DegradedLeft]/d5out_Point_Position := "NO_END_POSITION, DEGRADED_LEFT"; t4out_Point_Position := TRUE;
		Msg_Point_Position[ReportedPointPositionState = NoEndPosition AND ReportedDegradedPointPosition = DegradedRight]/d5out_Point_Position := "NO_END_POSITION, DEGRADED_RIGHT"; t4out_Point_Position := TRUE;
		Msg_Point_Position[ReportedPointPositionState = NoEndPosition AND ReportedDegradedPointPosition = NotDegraded]/d5out_Point_Position := "NO_END_POSITION, NOT_DEGRADED"; t4out_Point_Position := TRUE;
		Msg_Point_Position[ReportedPointPositionState = NoEndPosition AND ReportedDegradedPointPosition = NotApplicable]/d5out_Point_Position := "NO_END_POSITION, NOT_APPLICABLE"; t4out_Point_Position := TRUE;
		Msg_Point_Position[ReportedPointPositionState = UnintendedPosition AND ReportedDegradedPointPosition = DegradedLeft]/d5out_Point_Position := "UNINTENDED_POSITION, DEGRADED_LEFT"; t4out_Point_Position := TRUE;
		Msg_Point_Position[ReportedPointPositionState = UnintendedPosition AND ReportedDegradedPointPosition = DegradedRight]/d5out_Point_Position := "UNINTENDED_POSITION, DEGRADED_RIGHT"; t4out_Point_Position := TRUE;
		Msg_Point_Position[ReportedPointPositionState = UnintendedPosition AND ReportedDegradedPointPosition = NotDegraded]/d5out_Point_Position := "UNINTENDED_POSITION, NOT_DEGRADED"; t4out_Point_Position := TRUE;
		Msg_Point_Position[ReportedPointPositionState = UnintendedPosition AND ReportedDegradedPointPosition = NotApplicable]/d5out_Point_Position := "UNINTENDED_POSITION, NOT_APPLICABLE"; t4out_Point_Position := TRUE;
		ABILITY_TO_MOVE_OBSERVING
		Initial2
		RECEIVE_ABILITY_TO_MOVE_REPORT
		Msg_Ability_To_Move_Point[ReportedAbilityToMoveState = AbleToMove]/d8out_Ability_To_Move := "ABLE_TO_MOVE"; t7out_Ability_To_Move := TRUE;
		Msg_Ability_To_Move_Point[ReportedAbilityToMoveState = UnableToMove]/d8out_Ability_To_Move := "UNABLE_TO_MOVE"; t7out_Ability_To_Move := TRUE;
		MOVEMENT_FAILED_OBSERVING
		Initial3
		RECEIVE_MOVEMENT_FAILED_REPORT
		Msg_Movement_Failed/t6out_Movement_Failed := TRUE;
				Basic non-4-wire single P
				Basic non-4-wire multiple P
				Basic 4-wire single P
				Basic 4-wire multiple P

Eu.P.7042	Info	Initial0		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7043	Req	/{Initial0 - OPERATING}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.7044	Info	OPERATING		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7045	Info	ABILITY_TO_MOVE_OBSERVING		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7046	Info	Initial2		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7047	Req	/{Initial2 - RECEIVE_ABILITY_TO_MOVE_REPORT}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7048	Info	RECEIVE_ABILITY_TO_MOVE_REPORT		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7049	Req	Msg_Ability_To_Move_Point[ReportedAbilityToMoveState = UnableToMove]/d8out_Ability_To_Move := "UNABLE_TO_MOVE"; t7out_Ability_To_Move := TRUE;{State-internal in RECEIVE_ABILITY_TO_MOVE_REPORT}		Option Able to move
Eu.P.7050	Req	Msg_Ability_To_Move_Point[ReportedAbilityToMoveState = AbleToMove]/d8out_Ability_To_Move := "ABLE_TO_MOVE"; t7out_Ability_To_Move := TRUE;{State-internal in RECEIVE_ABILITY_TO_MOVE_REPORT}		Option Able to move
Eu.P.7051	Info	MOVEMENT_FAILED_OBSERVING		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7052	Info	Initial3		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7053	Req	/{Initial3 - RECEIVE_MOVEMENT_FAILED_REPORT}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7054	Info	RECEIVE_MOVEMENT_FAILED_REPORT		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7055	Req	Msg_Movement_Failed/t6out_Movement_Failed := TRUE;{State-internal in RECEIVE_MOVEMENT_FAILED_REPORT}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7056	Info	POINT_POSITION_OBSERVING		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7057	Info	Initial1		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7058	Req	/{Initial1 - RECEIVE_OVERALL_POINT_POSITION_AND_DEGRADED_POINT_POSITION_REPORT}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7059	Info	RECEIVE_OVERALL_POINT_POSITION_AND_DEGRADED_POINT_POSITION_REPORT		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7060	Req	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}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7061	Req	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}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7062	Req	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}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.7063	Req	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}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7064	Req	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}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7065	Req	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}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7066	Req	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}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7067	Req	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}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7068	Req	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}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7069	Req	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}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7070	Req	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}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7071	Req	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}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6412	Info	F_SCI_P_Receive		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6413	Info	[Block] F_SCI_P_Receive [Functional Viewpoint - Interface Requirements - Functional Entity]  ibd [Block] F_SCI_P_Receive [Functional Viewpoint - Interface Requirements - Functional Entity] «functional entity» F_SCI_P_Receive P3in : SCI_P_1 d10out_Required_Point_Position : String d11in_Observed_Ability_To_Move : String d12in_Required_PM_Position : String d50in_PDI_Connection_State : String		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6415	Info	d10out_Required_Point_Position	The port d10out_Required_Point_Positiont contains the Point Target Position. The following values are valid: "LEFT", "RIGHT".	Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6429	Info	d50in_PDI_Connection_State		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6430	Info	d12in_Required_PM_Position		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7018	Info	d11in_Observed_Ability_To_Move		Option Able to move

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.6470	Info	P3in	The port P3in exchanges information objects according to SCI_P_1.	Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6431	Info	F_SCI_P_Receive - Behaviour		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6432	Info	Functional Viewpoint - Interface Requirements - Functional Entity STD 2  stm [State Machine] F_SCI_P_Receive - Behaviour [Functional Viewpoint - Interface Requirements - Functional Entity STD 2] Initial0 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 Entry/d10out_Required_Point_Position := "UNCOMMANDED"; Cd_Move_Point[CommandedPointPositionState = Left AND d50in_PDI_Connection_State = "ESTABLISHED"]/d10out_Required_Point_Position := "LEFT"; Cd_Move_Point[CommandedPointPositionState = Right AND d50in_PDI_Connection_State = "ESTABLISHED"]/d10out_Required_Point_Position := "RIGHT"; when(d12in_Required_PM_Position = "UNCOMMANDED")/ when(d11in_Observed_Ability_To_Move = "ABLE_TO_MOVE")/		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6433	Info	Initial0		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6434	Info	/{Initial0 - RECEIVING_COMMANDS}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6435	Info	RECEIVING_COMMANDS		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6469	Req	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}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7019	Req	Cd_Move_Point[CommandedPointPositionState = Left AND d50in_PDI_Connection_State = "ESTABLISHED"]/d10out_Required_Point_Position := "LEFT";{State-internal in RECEIVING_COMMANDS}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7020	Req	Cd_Move_Point[CommandedPointPositionState = Right AND d50in_PDI_Connection_State = "ESTABLISHED"]/d10out_Required_Point_Position := "RIGHT";{State-internal in RECEIVING_COMMANDS}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7021	Req	entry/d10out_Required_Point_Position := "UNCOMMANDED";{State-internal in RECEIVING_COMMANDS}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7022	Req	when(d11in_Observed_Ability_To_Move = "ABLE_TO_MOVE")/{RECEIVING_COMMANDS - RECEIVING_COMMANDS}		Option Able to move
Eu.P.7023	Req	when(d12in_Required_PM_Position = "UNCOMMANDED")/{RECEIVING_COMMANDS - RECEIVING_COMMANDS}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6522	Info	F_SCI_P_Report		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P

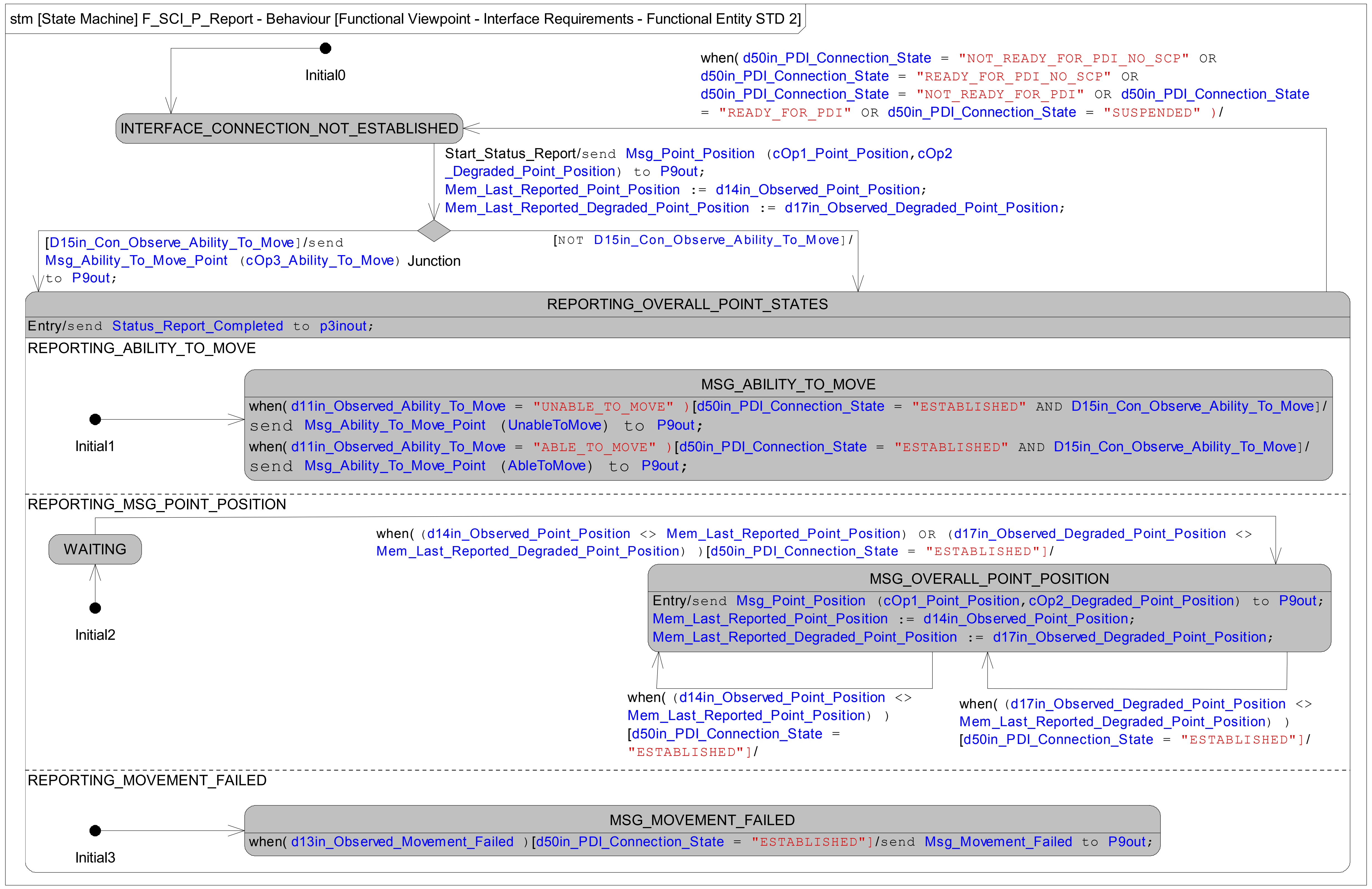
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.6523	Info	[Block] F_SCI_P_Report [Functional Viewpoint - Interface Requirements - Functional Entity]  ibd [Block] F_SCI_P_Report [Functional Viewpoint - Interface Requirements - Functional Entity] «functional entity» F_SCI_P_Report Operation «Operation» cOp1_Point_Position () : PointPositionState «Operation» cOp2_Degraded_Point_Position () : PointPositionDegradedState «Operation» cOp3_Ability_To_Move () : AbilityToMoveState values «BlockProperty» Mem_Last_Reported_Degraded_Point_Position : String «BlockProperty» Mem_Last_Reported_Point_Position : String d11in_Observed_Ability_To_Move : String p3inout : F_SCI_Specific d13in_Observed_Movement_Failed : Boolean P9out : SCI_P_2 d14in_Observed_Point_Position : String D15in_Con_Observe_Ability_To_Move : Boolean d17in_Observed_Degraded_Point_Position : String d50in_PDI_Connection_State : String		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P

Eu.P.6524	Info	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	Basic non-4-wire single P
				Basic non-4-wire multiple P
				Basic 4-wire single P
				Basic 4-wire multiple P

Eu.P.6525	Info	if (d17in_Observed_Degraded_Point_Position = "DEGRADED_LEFT") then
		    return PointPositionDegradedState.DegradedLeft;
		elseif (d17in_Observed_Degraded_Point_Position = "DEGRADED_RIGHT") then
		    return PointPositionDegradedState.DegradedRight;
		elseif (d17in_Observed_Degraded_Point_Position = "DEGRADED_DENIED") then
		    return PointPositionDegradedState.NotApplicable;
		else
		    return PointPositionDegradedState.NotDegraded;
		end if
			cOp2_Degraded_Point_Position	Basic non-4-wire single P
				Basic non-4-wire multiple P
				Basic 4-wire single P
				Basic 4-wire multiple P

Eu.P.6526	Info	if (d11in_Observed_Ability_To_Move = "ABLE_TO_MOVE") then
		    return AbilityToMoveState.AbleToMove;
		elseif (d11in_Observed_Ability_To_Move = "UNABLE_TO_MOVE") then
		    return AbilityToMoveState.UnableToMove;
		end if
			cOp3_Ability_To_Move	Option Able to move

Eu.P.6527	Info	d17in_Observed_Degraded_Point_Position		Basic non-4-wire multiple P Basic 4-wire multiple P
Eu.P.6528	Info	d50in_PDI_Connection_State		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6529	Info	d14in_Observed_Point_Position		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6530	Info	d11in_Observed_Ability_To_Move		Option Able to move
Eu.P.6556	Info	P9out	The port P9out exchanges information objects according to SCI_P_2.	Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6557	Info	p3inout		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7024	Info	d13in_Observed_Movement_Failed		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7025	Info	D15in_Con_Observe_Ability_To_Move		Option Able to move

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.6531	Info	F_SCI_P_Report - Behaviour		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6532	Info	Functional Viewpoint - Interface Requirements - Functional Entity STD 2 		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6533	Info	Initial0		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6534	Req	/{Initial0 - INTERFACE_CONNECTION_NOT_ESTABLISHED}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6535	Info	INTERFACE_CONNECTION_NOT_ESTABLISHED		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P

Eu.P.6536	Req	Start_Status_Report/send Msg_Point_Position (cOp1_Point_Position,cOp2_Degraded_Point_Position) to P9out;
		Mem_Last_Reported_Point_Position := d14in_Observed_Point_Position;
		Mem_Last_Reported_Degraded_Point_Position := d17in_Observed_Degraded_Point_Position;{INTERFACE_CONNECTION_NOT_ESTABLISHED - Junction}
				Basic non-4-wire single P
				Basic non-4-wire multiple P
				Basic 4-wire single P
				Basic 4-wire multiple P

Eu.P.6537	Info	REPORTING_OVERALL_POINT_STATES		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6538	Info	REPORTING_ABILITY_TO_MOVE		Option Able to move
Eu.P.6539	Info	Initial1		Option Able to move
Eu.P.6540	Req	/{Initial1 - MSG_ABILITY_TO_MOVE}		Option Able to move

stm [State Machine] F_SCI_P_Report - Behaviour [Functional Viewpoint - Interface Requirements - Functional Entity STD 2]
Initial0
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")/
INTERFACE_CONNECTION_NOT_ESTABLISHED
Start_Status_Report/send Msg_Point_Position (cOp1_Point_Position,cOp2_Degraded_Point_Position) to P9out; Mem_Last_Reported_Point_Position := d14in_Observed_Point_Position; Mem_Last_Reported_Degraded_Point_Position := d17in_Observed_Degraded_Point_Position;
[D15in_Con_Observe_Ability_To_Move]/send Msg_Ability_To_Move_Point (cOp3_Ability_To_Move) to P9out;
Junction
[NOT D15in_Con_Observe_Ability_To_Move]/
REPORTING_OVERALL_POINT_STATES
Entry/send Status_Report_Completed to p3inout;
REPORTING_ABILITY_TO_MOVE
MSG_ABILITY_TO_MOVE
when(d11in_Observed_Ability_To_Move = "UNABLE_TO_MOVE")[d50in_PDI_Connection_State = "ESTABLISHED" AND D15in_Con_Observe_Ability_To_Move]/ send Msg_Ability_To_Move_Point (UnableToMove) to P9out;
when(d11in_Observed_Ability_To_Move = "ABLE_TO_MOVE")[d50in_PDI_Connection_State = "ESTABLISHED" AND D15in_Con_Observe_Ability_To_Move]/ send Msg_Ability_To_Move_Point (AbleToMove) to P9out;
Initial1
REPORTING_MSG_POINT_POSITION
WAITING
Initial2
when((d14in_Observed_Point_Position <> Mem_Last_Reported_Point_Position) OR (d17in_Observed_Degraded_Point_Position <> Mem_Last_Reported_Degraded_Point_Position))[d50in_PDI_Connection_State = "ESTABLISHED"]/
MSG_OVERALL_POINT_POSITION
Entry/send Msg_Point_Position (cOp1_Point_Position,cOp2_Degraded_Point_Position) to P9out; Mem_Last_Reported_Point_Position := d14in_Observed_Point_Position; Mem_Last_Reported_Degraded_Point_Position := d17in_Observed_Degraded_Point_Position;
when((d14in_Observed_Point_Position <> Mem_Last_Reported_Point_Position)) [d50in_PDI_Connection_State = "ESTABLISHED"]/
when((d17in_Observed_Degraded_Point_Position <> Mem_Last_Reported_Degraded_Point_Position)) [d50in_PDI_Connection_State = "ESTABLISHED"]/
REPORTING_MOVEMENT_FAILED
MSG_MOVEMENT_FAILED
when(d13in_Observed_Movement_Failed)[d50in_PDI_Connection_State = "ESTABLISHED"]/send Msg_Movement_Failed to P9out;
Initial3

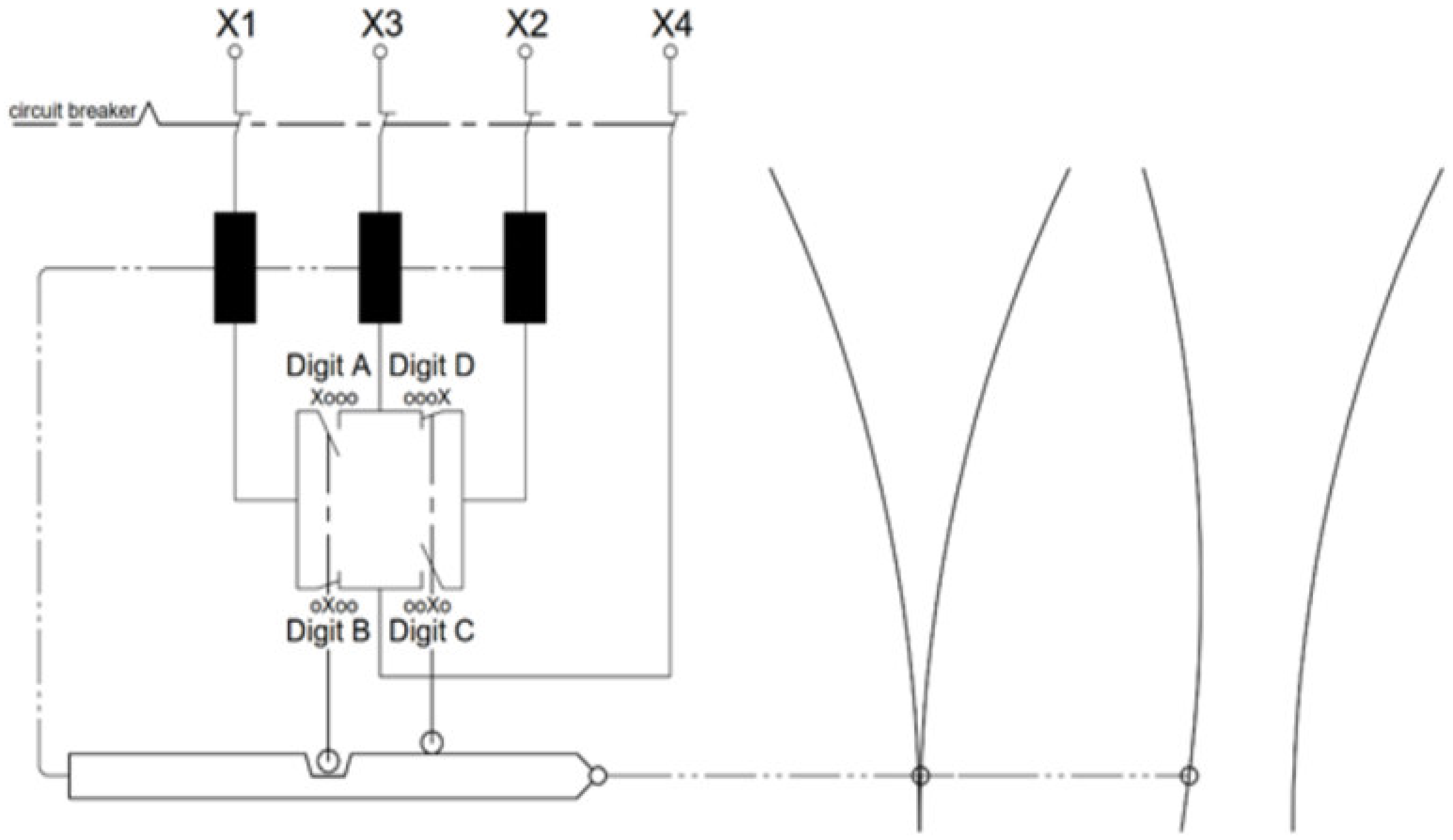
ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.6541	Info	MSG_ABILITY_TO_MOVE		Option Able to move
Eu.P.6542	Req	when(d11in_Observed_Ability_To_Move = "ABLE_TO_MOVE")[d50in_PDI_Connection_State = "ESTABLISHED" AND D15in_Con_Observe_Ability_To_Move]/ send Msg_Ability_To_Move_Point (AbleToMove) to P9out;{State-internal in MSG_ABILITY_TO_MOVE}		Option Able to move
Eu.P.6543	Req	when(d11in_Observed_Ability_To_Move = "UNABLE_TO_MOVE")[d50in_PDI_Connection_State = "ESTABLISHED" AND D15in_Con_Observe_Ability_To_Move]/ send Msg_Ability_To_Move_Point (UnableToMove) to P9out;{State-internal in MSG_ABILITY_TO_MOVE}		Option Able to move
Eu.P.6544	Info	REPORTING_MSG_POINT_POSITION		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6545	Info	Initial2		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6546	Req	/{Initial2 - WAITING}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6547	Info	MSG_OVERALL_POINT_POSITION		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P

Eu.P.6548	Req	entry/send Msg_Point_Position (cOp1_Point_Position,cOp2_Degraded_Point_Position) to P9out;
		Mem_Last_Reported_Point_Position := d14in_Observed_Point_Position;
		Mem_Last_Reported_Degraded_Point_Position := d17in_Observed_Degraded_Point_Position;{State-internal in MSG_OVERALL_POINT_POSITION}
				Basic non-4-wire single P
				Basic non-4-wire multiple P
				Basic 4-wire single P
				Basic 4-wire multiple P

Eu.P.6549	Req	when((d17in_Observed_Degraded_Point_Position <> Mem_Last_Reported_Degraded_Point_Position))[d50in_PDI_Connection_State = "ESTABLISHED"]/{MSG_OVERALL_POINT_POSITION - MSG_OVERALL_POINT_POSITION}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6550	Req	when((d14in_Observed_Point_Position <> Mem_Last_Reported_Point_Position))[d50in_PDI_Connection_State = "ESTABLISHED"]/{MSG_OVERALL_POINT_POSITION - MSG_OVERALL_POINT_POSITION}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6551	Info	WAITING		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6553	Req	when((d14in_Observed_Point_Position <> Mem_Last_Reported_Point_Position) OR (d17in_Observed_Degraded_Point_Position <> Mem_Last_Reported_Degraded_Point_Position))[d50in_PDI_Connection_State = "ESTABLISHED"]/{WAITING - MSG_OVERALL_POINT_POSITION}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6554	Req	entry/send Status_Report_Completed to p3inout;{State-internal in REPORTING_OVERALL_POINT_STATES}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6555	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")/{REPORTING_OVERALL_POINT_STATES - INTERFACE_CONNECTION_NOT_ESTABLISHED}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7029	Info	REPORTING_MOVEMENT_FAILED		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7030	Info	Initial3		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7031	Req	/{Initial3 - MSG_MOVEMENT_FAILED}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7032	Info	MSG_MOVEMENT_FAILED		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7033	Req	when(d13in_Observed_Movement_Failed)[d50in_PDI_Connection_State = "ESTABLISHED"]/send Msg_Movement_Failed to P9out;{State-internal in MSG_MOVEMENT_FAILED}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7026	Info	Junction		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.7027	Req	[NOT D15in_Con_Observe_Ability_To_Move]/{Junction - REPORTING_OVERALL_POINT_STATES}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7028	Req	[D15in_Con_Observe_Ability_To_Move]/send Msg_Ability_To_Move_Point (cOp3_Ability_To_Move) to P9out;{Junction - REPORTING_OVERALL_POINT_STATES}		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.929	Head	**3.4.2 SMI-P (Subsystem - Maintenance and Data Management)**		
Eu.P.3066	Info	The generic FlowSpecification and the related FlowProperties through SMI-P are specified in Eu.Doc.120.		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.925	Head	**3.4.3 SDI-P (Subsystem - Maintenance and Data Management)**		
Eu.P.3065	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.		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6156	Head	**3.4.4 SSI-P (Subsystem - Security Services Platform)**		
Eu.P.6157	Info	The generic content through SSI-P is specified in Eu.Doc.117.		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.910	Head	**3.4.5 P4 (Basic Data Identifier)**		
Eu.P.3063	Info	The generic FlowSpecification and the related FlowProperties through P4 are specified in Eu.Doc.20.		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.890	Head	**3.4.6 P1 (Maintainer)**		
Eu.P.3173	Info	The generic FlowProperties through P1 are specified in Eu.Doc.20.		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6206	Info	The defined FlowProperties through P1 are mandatory only when the physical interfaces related to the specific maintainer information are available on the Subsystem – Point. Example: The FlowProperty Point_Unintended_Position is only mandatory when the Subsystem – Point is equipped with a point machine interface that can detect unintended position.		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.891	Info	Maintainer	Definition of the InformationFlow (by FlowSpecification) for Maintenance/Operation/Display P1 (Maintainer).	Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.896	Req	Point_Moving	Displays the moving of the point at the local status display.	Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.1377	Req	End_Position_R	Displays the status of the detection of point end position on the right hand.	Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.894	Req	End_Position_L	Displays the status of the detection of point end position on the left hand.	Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.3037	Req	Point_Unintended_Position	Displays the unintended position of the point at the local status display (in an unintended position or not).	Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.4949	Req	P3_Moving_N	Displays the state for each respective physical Point Machine Output (N > 0). (Moving, Stopped).	Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.4950	Req	P3_Status_N	Displays the state for each respective physical Point Machine Input (N > 0). (End_Position, No_End_Position, Unintended_Position).	Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.902	Head	**3.4.7 P3 (Point machine)**		
Eu.P.903	Info	Point_machine	Definition of the InformationFlow (by FlowSpecification) for the Control Interfaces P3 (Point machine). Note:	Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
			The behaviour of the interfaces P3 is described 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.904	Req	Information_End_Position_Arrived	Information object from Point machine to Subsystem - Point that the Point machine has an End position (left hand position or right hand position). Note: This information is only available in case the Point machine is implemented with a non-4-wire interface.	Basic non-4-wire single P Basic non-4-wire multiple P
Eu.P.905	Req	Information_No_End_Position	Information object from Point machine to Subsystem - Point that the Point machine has No end position. Note: This is expressed as a 4-wire pattern in case the Point machine is implemented with a 4-wire interface. The interpretation of the 4-wire patterns depends on the driving/detection state and the last commanded position. The interpretation tables can be found in Eu.P.6797.	Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.906	Req	Information_Unintended_Position	Information object from Point machine to Subsystem - Point that the Point machine has an Unintended position position. Note 1: This is expressed as a 4-wire pattern in case the Point machine is implemented with a 4-wire interface. The interpretation of the 4-wire patterns depends on the driving/detection state and the last commanded position. The interpretation tables can be found in Eu.P.6797. Note 2: This information may not be available in case the Point machine is implemented with a non-4-wire interface.	Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.907	Req	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).	Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.909	Req	Stop_Moving	Information object from Subsystem - Point to Point machine to stop Moving the Point machine.	Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.5364	Req	Information_Ability_To_Move_Point	Information object from Point machine to Subsystem - Point that the Point machine has Drive voltage (able) or has no Drive voltage (unable). Note: The reason for reporting ability to move depends on national specification and also Drive voltage levels monitored shall be defined by national specifications. In future phases of the System Pillar, national specifications will be replaced by harmonised specifications.	Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P Option Able to move
Eu.P.6274	Req	Detection_Voltage	Information object from Subsystem - Point to Point machine to switch the Point machine to Detection Mode (on or off). Note: This information is only available in case the Point machine is implemented with a 4-wire interface.	Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7016	Req	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). Note 1: This information is only available in case the Point machine is implemented with a 4-wire interface. This is expressed as a 4-wire pattern. LEFT: 1010 RIGHT: 0101	Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7017	Req	Information_End_Position_Reached	Information object from Point machine to Subsystem - Point that the Point machine has reached an End position (left hand position or right hand position), but the position has not yet been reliably detected. Note 1: This information is only available in case the Point machine is implemented with a 4-wire interface. This is expressed as a 4-wire pattern. LEFT: 1010 RIGHT: 0101	Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6797	Head	**3.4.7.1 Interpretation tables of 4-wire patterns**		Basic 4-wire single P Basic 4-wire multiple P

<table>
<tr><th>ID</th><th>Type</th><th>Requirement Part 1</th><th>Requirement Part 2</th><th>Func. Pkg.</th></tr>
<tr><td>Eu.P.6801</td><td>Info</td><td>For the 4-wire implementation, the input information is represented as 4-wire patterns.</td><td></td><td>Basic 4-wire single P
Basic 4-wire multiple P</td></tr>
<tr><td>Eu.P.6806</td><td>Info</td><td>The 4-wire pattern consists of four digits, each being in a state of „1" or „0". The state of „1" represents a closed contact in the 4-wire circuit while „0" represents an open contact in the 4-wire circuit.</td><td></td><td>Basic 4-wire single P
Basic 4-wire multiple P</td></tr>
<tr><td>Eu.P.6805</td><td>Info</td><td>There are four contact pairs, where each pair is represented by a specific digit in the 4-wire pattern (ABCD):
Contact 1+3 -> Digit A
Contact 1+4 -> Digit B
Contact 2+4 -> Digit C
Contact 2+3 -> Digit D</td><td></td><td>Basic 4-wire single P
Basic 4-wire multiple P</td></tr>
<tr><td>Eu.P.6807</td><td>Info</td><td>The figure shows a schematic representation of the 4-wire circuit

 </td><td></td><td>Basic 4-wire single P
Basic 4-wire multiple P</td></tr>
<tr><td>Eu.P.6804</td><td>Req</td><td>The interpretation of the 4-wire patterns is state-dependent. There are 5 interpretations states. The interpretation state is determined by two factors.
- Whether drive or detection voltage is provided to the point machine (moving to left, moving to right, no drive/detection)
- The last commanded position of the point machine (left, right, not commanded)</td><td></td><td>Basic 4-wire single P
Basic 4-wire multiple P</td></tr>
<tr><td>Eu.P.6803</td><td>Req</td><td>The implementation of the Subsystem Point and/or the Point machine shall be such that the last commanded position is always retained, including in case of loss of communication and in case of rebooting.
Note: The only case in which the state ‘not commanded’ occurs is after the initial start-up.</td><td></td><td>Basic 4-wire single P
Basic 4-wire multiple P</td></tr>
<tr><td>Eu.P.6703</td><td>Req</td><td>Interpretation of 4-wire patterns with detection voltage and commanded position "Right"
<table>
<tr><th>4-wire pattern</th><th>Interpretation of information on interface P3</th></tr>
<tr><td>0000</td><td>Information_No_End_Position</td></tr>
<tr><td>0001</td><td>Information_No_End_Position</td></tr>
<tr><td>0010</td><td>Supplier specific</td></tr>
<tr><td>0011</td><td>Supplier specific</td></tr>
<tr><td>0100</td><td>Information_No_End_Position</td></tr>
<tr><td>0101</td><td>Information_End_Position_Detected (Right)</td></tr>
<tr><td>0110</td><td>Supplier specific</td></tr>
<tr><td>0111</td><td>Supplier specific (not Information_No_End_Position)</td></tr>
<tr><td>1000</td><td>Supplier specific</td></tr>
<tr><td>1001</td><td>Information_Unintended_Position</td></tr>
<tr><td>1010</td><td>Information_Unintended_Position</td></tr>
<tr><td>1011</td><td>Supplier specific (not Information_No_End_Position)</td></tr>
<tr><td>1100</td><td>Supplier specific</td></tr>
<tr><td>1101</td><td>Supplier specific (not Information_No_End_Position)</td></tr>
<tr><td>1110</td><td>Supplier specific (not Information_No_End_Position)</td></tr>
<tr><td>1111</td><td>Supplier specific (not Information_No_End_Position)</td></tr>
</table></td><td></td><td>Basic 4-wire single P
Basic 4-wire multiple P</td></tr>
<tr><td>Eu.P.6704</td><td>Req</td><td>Interpretation of 4-wire patterns with detection voltage and commanded position "Left"</td><td></td><td>Basic 4-wire single P
Basic 4-wire multiple P</td></tr>
</table>

<table>
<tr><th>ID</th><th>Type</th><th>Requirement Part 1</th><th>Requirement Part 2</th><th>Func. Pkg.</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><th>4-wire pattern</th><th>Interpretation of information on interface P3</th></tr>
<tr><td>0000</td><td>Information_No_End_Position</td></tr>
<tr><td>0001</td><td>Supplier specific</td></tr>
<tr><td>0010</td><td>Information_No_End_Position</td></tr>
<tr><td>0011</td><td>Supplier specific</td></tr>
<tr><td>0100</td><td>Supplier specific</td></tr>
<tr><td>0101</td><td>Information_Unintended_Position</td></tr>
<tr><td>0110</td><td>Supplier specific</td></tr>
<tr><td>0111</td><td>Supplier specific (not Information_No_End_Position)</td></tr>
<tr><td>1000</td><td>Information_No_End_Position</td></tr>
<tr><td>1001</td><td>Information_Unintended_Position</td></tr>
<tr><td>1010</td><td>Information_End_Position_Detected (Left)</td></tr>
<tr><td>1011</td><td>Supplier specific (not Information_No_End_Position)</td></tr>
<tr><td>1100</td><td>Supplier specific</td></tr>
<tr><td>1101</td><td>Supplier specific (not Information_No_End_Position)</td></tr>
<tr><td>1110</td><td>Supplier specific (not Information_No_End_Position)</td></tr>
<tr><td>1111</td><td>Supplier specific (not Information_No_End_Position)</td></tr>
</table>
</td><td></td><td></td></tr>
<tr><td>Eu.P.6798</td><td>Req</td><td>Interpretation of 4-wire patterns with detection voltage and no commanded position
<table>
<tr><th>4-wire pattern</th><th>Interpretation of information on interface P3</th></tr>
<tr><td>0000</td><td>Information_No_End_Position</td></tr>
<tr><td>0001</td><td>Information_No_End_Position</td></tr>
<tr><td>0010</td><td>Information_No_End_Position</td></tr>
<tr><td>0011</td><td>Supplier specific</td></tr>
<tr><td>0100</td><td>Information_No_End_Position</td></tr>
<tr><td>0101</td><td>Information_End_Position_Detected (Right)</td></tr>
<tr><td>0110</td><td>Supplier specific</td></tr>
<tr><td>0111</td><td>Supplier specific</td></tr>
<tr><td>1000</td><td>Information_No_End_Position</td></tr>
<tr><td>1001</td><td>Information_Unintended_Position</td></tr>
<tr><td>1010</td><td>Information_End_Position_Detected (Left)</td></tr>
<tr><td>1011</td><td>Supplier specific</td></tr>
<tr><td>1100</td><td>Supplier specific</td></tr>
<tr><td>1101</td><td>Supplier specific</td></tr>
<tr><td>1110</td><td>Supplier specific</td></tr>
<tr><td>1111</td><td>Supplier specific</td></tr>
</table>
</td><td></td><td>Basic 4-wire single P
Basic 4-wire multiple P</td></tr>
<tr><td>Eu.P.6799</td><td>Req</td><td>Interpretation of 4-wire patterns when driving and moving to "Right"
<table>
<tr><th>4-wire pattern</th><th>Interpretation of information on interface P3</th></tr>
<tr><td>0101</td><td>Information_End_Position_Reached (Right)</td></tr>
<tr><td>All other</td><td>Information_No_End_Position</td></tr>
</table>
</td><td></td><td>Basic 4-wire single P
Basic 4-wire multiple P</td></tr>
<tr><td>Eu.P.6800</td><td>Req</td><td>Interpretation of 4-wire patterns when driving and moving to "Left"
<table>
<tr><th>4-wire pattern</th><th>Interpretation of information on interface P3</th></tr>
<tr><td>1010</td><td>Information_End_Position_Reached (Left)</td></tr>
<tr><td>All other</td><td>Information_No_End_Position</td></tr>
</table>
</td><td></td><td>Basic 4-wire single P
Basic 4-wire multiple P</td></tr>
<tr><td>Eu.P.6802</td><td>Req</td><td>For certain 4-wire patterns, a supplier specific interpretation is allowed. When one of these patterns is detected on the 4-wire interface to the Point Machine, the Subsystem Point may react to it in three ways:
- interpret the abstract state ‘Information_No_End_Position’.
- interpret the abstract state ‘Information_Unintended_Position’
- detect a fatal error (T5in_SIL_Not_Fulfilled in generic EfeS state machine)

The supplier may choose which of the 3 system reactions is most appropriate. The reaction may also be time dependent.
For the 4-wire patterns 0111, 1011, 1101, 1110 and 1111, it is not allowed to have a persistent interpretation as the abstract state ‘Information_No_End_Position’ if the commanded position is 'Right' or 'Left'.</td><td></td><td>Basic 4-wire single P
Basic 4-wire multiple P</td></tr>
<tr><td>Eu.P.233</td><td>Head</td><td><b>4 RAMSS requirements</b></td><td></td><td></td></tr>
<tr><td>Eu.P.2987</td><td>Info</td><td>The requirements for reliability, availability, maintainability, safety and security are specified in [Eu.Doc.20]</td><td></td><td>Basic non-4-wire single P
Basic non-4-wire multiple P
Basic 4-wire single P
Basic 4-wire multiple P</td></tr>
<tr><td>Eu.P.3244</td><td>Head</td><td><b>5 Technical Requirements</b></td><td></td><td></td></tr>
<tr><td>Eu.P.3245</td><td>Info</td><td>The generic technical requirements are specified in [Eu.Doc.20]</td><td></td><td>Basic non-4-wire single P
Basic non-4-wire multiple P
Basic 4-wire single P
Basic 4-wire multiple P</td></tr>
<tr><td>Eu.P.3246</td><td>Head</td><td><b>5.1 Specific technical interface requirements</b></td><td></td><td></td></tr>
<tr><td>Eu.P.3247</td><td>Head</td><td><b>5.1.1 Interface to the Point of Service Signalling (PoS-Signalling)</b></td><td></td><td></td></tr>
</table>

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
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].		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
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].		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
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].		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
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].		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.3251	Head	**5.1.2 Interface to the point machine**		
Eu.P.6271	Req	The interface to the point machine can be implemented as one of 2 variants: - non-4-wire - 4-wire		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
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.		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
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 the power supply for driving is disconnected. • If the point machine monitors its power supply for driving, a loss of the power supply for driving will report as "Information_Ability_To_Move_Point" on P3. (Note: A point machine may not be implemented with the function to monitor its power supply for driving).		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
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.		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.4942	Req	On a Point machine with detection functionality only, the information objects Moving and Stop_Moving are not available on P3.		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.4943	Req	The timing behaviour related to multiple Point Machines for a single Point is supplier specific and is not specified by EULYNX as part of the application layer, this shall be handled by the physical implementation.		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
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.		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
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.		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.3264	Req	The Subsystem - Point shall start the reversal operation within 500 ms after receiving a command telegram.		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.3265	Req	The Subsystem - Point shall start the redrive operation within 500 ms after detecting No end position.		Option Redrive
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:		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.3258	Req	• The applicable timers defined in chapter Definition of time values (Eu.P.2286).		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P

ID	Type	Requirement Part 1	Requirement Part 2	Func. Pkg.
Eu.P.4947	Req	• The number of implemented Point Machines		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.4948	Req	• For each configured Point Machine, whether it is implemented with full functionality or as point detector only. The 1st PM must be configured with drive capability.		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6293	Req	• Whether the Subsystem - Point is configured to perform redrive		Option Redrive
Eu.P.6294	Req	• Whether the Subsystem - Point is configured to detect and report unintended position		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6295	Req	• For each configured Point Machine, whether it is crucial or non-crucial		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.6636	Req	• The implementation variant of the interface to the point machine (non-4-wire/4-wire)		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.7108	Req	• Whether the Subsystem - Point is configured to detect and report Ability to Move		Option Able to move
Eu.P.7109	Req	• Whether the Subsystem - Point is configured to stop driving Point Machines individually or according to Common Drive.		Option Common Drive
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:		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
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.		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.3261	Req	• The remaining configuration data is currently categorised as safety-relevant. This data shall be used to calculate the CSS.		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P
Eu.P.4546	Req	• The engineering data is safety-relevant. This data shall be used to calculate the CSS.		Basic non-4-wire single P Basic non-4-wire multiple P Basic 4-wire single P Basic 4-wire multiple P