




ERJU SYSTEM PILLAR

Systems Engineering Management Plan - Annex M1 Capella Model Element Rules



Systems Engineering Management Plan - Annex M1 Capella Model Element Rules

Author(s)	Jorge Block , Dennis Kunz
Abstract	This document contains the element modelling rules and descriptions for the usage of Capella.
Config Item	System Engineering Management Plan
Document ID	SEMP Annexes/SEMP Annex M1 Capella element rules#722475  Systems Engineering Management Plan - Annex M1 Capella Model Element Rules
Classification	Public
Status	Released
Version	2.0
Revision	722475
Last Change Date	02.10.2025
Copyright	Brussels: Europe's Rail Joint Undertaking, 2025

© Europe's Rail Joint Undertaking, 2025

This document is drafted by and belongs to EU Rail.

EU Rail encourages the distribution and re-use of this document, the technical specifications and the information it contains. EU Rail holds several intellectual property rights, such as copyright and trade mark rights, which need to be considered when this document is used.

EU Rail authorises you to re-publish, re-use, copy and store this document without changing it, provided that you indicate its source and include the following: EU Rail trade mark, title of the document, year of publication, version of document.

EU Rail makes no representation or warranty as to the accuracy or completeness of the information contained within these documents. EU Rail shall have no liability to any party as a result of the use of the information contained herein. EU Rail will have no liability whatsoever for any indirect or consequential loss or damage, and any such liability is expressly excluded.


You may study, research, implement, adapt, improve and otherwise use the information, the content and the models in the this document for your own purposes. If you decide to publish or disclose any adapted, modified or improved version of this document, any amended implementation or derivative work, then you must indicate that you have modified this document, with a reference to the document name and the terms of use of this document. You may not use EU Rail's trade marks or name in any way that may state or suggest, directly or indirectly, that EU Rail is the author of your adaptations.

EU Rail cannot be held responsible for your product, even if you have used this document and its content. It is your responsibility to verify the quality, completeness and the accuracy of the information you use, for your own purposes.

Document History

1.0 02.01.2023	Schmidt Steffen (I-NAT-SIBS-ISST)	Approved version based on Review X.X
1.1 03.07.2025	Jorge Block	Reviewed version including Change Request
2.0 24.09.2025	Jorge Block	Approved version based on Review 1.1

Approval by reviewers (captured at end of 'In Review by System Pillar')

Type of Approval	 Document Review
Comments	#1 by Jorge Block on 2025-09-24 11:28 Due date of approval reached, no answer for remaining waiting approvers.
Approvals	EDDOUS Sayfeddine (SNCF RESEAU / Directions Techniques Réseau / DGII DTR GE SF Solutions) : Approved , Renard, Marie Pierre (SMO RI MT FR ADC TGMTR3) : Approved , Marjan Haj-Mohammad-Ali-Hidalgo : Approved , michael.wild : Waiting , Simon Lambert : Approved , Schöni Ulrich (I-NAT-GST-CCS) : Waiting
Attachments	SystemEngineeringManagementPlan_M1.xls

Approval by approvers (captured at end of 'In Approval by System Pillar')


Type of Approval	 Document Approval
Approvals	SANGO Marc (SNCF / DIR TECHNOLOGIES INNOVATION ET PROJETS GROUPE / IR DIR RECHERCHE - PSF) : Approved , KEFALAS, Georgios : Approved , Renard, Marie Pierre (SMO RI MT FR ADC TGMTR3) : Approved , Schmidt Steffen (I-NAT-GST-ERTM) : Approved , SCHWAN Nico : Approved

Table of contents

1 Preamble	5
1.1 Purpose	5
1.2 Intended Audience	5
1.3 Document Context	5
1.4 Glossary	5
2 Introduction	6
2.1 Instructions for readers	6
2.2 Overview of rules	6
3 Rules for structure	10
3.1 Systems and Actors	10
3.2 Interfaces	13
4 Rules for behaviour	16
4.1 Capabilities	16
4.2 Functions	19
4.3 Functional exchanges	22
4.4 Scenarios	24
4.5 Functional chains	25
4.6 State machines	25
5 Rules for exchange items	28
6 Rules for data types and classes	30
7 Common rules	34
7.1 Language conventions	34
7.2 Rationales and assumptions	34
7.3 Tool-related considerations	35
7.4 Conditions	35
8 Appendix	37
8.1 Construction and interpretation methods	37
8.2 Compliance report	37

1 Preamble


1.1 Purpose

This guideline provides Capella model element rules, explanations and examples for the model elements used inside System Pillar architecture models. For viewpoint definition, rules and descriptions regarding diagrams in such models please consider *SPPROCESS/10 SEMP V 01_01/SEMP Annex M2 Capella diagram rules : 722475*.


1.2 Intended Audience

The content of this plan is valid for all the System Pillar tasks and domains.

1.3 Document Context

This document is an annex of the  Systems Engineering Management Plan - 02 MBSE Methodology Handbook

1.4 Glossary


This document primarily outlines the rules for model elements, while the definitions can be found in  Systems Engineering Management Plan - Annex M2 Viewpoint Guidelines.

Acronym (abbreviation)	Full text (title)
C2P	Capella2Polarion

2 Introduction

2.1 Instructions for readers

Usage of model element rules

Each rule in this document has a property called "Compliance" which defines whether a rule is mandatory or optional in the context of maintaining  SPPR-9823 - Capella2Polarion model elements. The following compliance levels are used:

Compliance level	Explanation
Required	This rule is mandatory to ensure that the Capella model elements are consistent and of high quality.
Recommended	This rule is optional, but provides recommended best practices.
Suggested	This rule is optional, but gives suggested hints and ideas.






[SPPR-11567]

2.2 Overview of rules

This chapter provides an overview of the rules to be applied per each model element.








Operational Entity

- Operational Entity
- Operational Actor

Type	 C2P-Operational Entity
ID	SPPR-10221
Linked Work Items	<p>refers to :  SPPR-2576 - Operational Entity/Actor</p> <p>has parent :  SPPR-11568 - Overview of rules</p> <p>_ is ruled by :  SPPR-10061 - Structural entity has a unique name</p> <p>_ is ruled by :  SPPR-10063 - Structural entity has a textual description</p>






Actor

- System Actor
- Logical Actor
- Physical Actor

Type	 C2P-Actor
ID	SPPR-10060
Linked Work Items	<p>refers to :  SPPR-2579 - Actor</p> <p>has parent :  SPPR-11568 - Overview of rules</p> <p>_ is ruled by :  SPPR-4007 - Actor has at least one realisation relationship</p> <p>_ is ruled by :  SPPR-4009 - Actor is configured as a human or external system</p> <p>_ is ruled by :  SPPR-10061 - Structural entity has a unique name</p> <p>_ is ruled by :  SPPR-10063 - Structural entity has a textual description</p>









System

- System Component
- Logical Component
- Physical Component (Node or Behaviour)

Type	 C2P-System
ID	SPPR-10062
Linked Work Items	refers to :  SPPR-2578 - System has parent :  SPPR-11568 - Overview of rules _ is ruled by :  SPPR-10061 - Structural entity has a unique name _ is ruled by :  SPPR-10063 - Structural entity has a textual description














Interface

- System Component Exchange
- Logical Component Exchange
- Physical Component Exchange







Type	 C2P-Interface
Layer	System Analysis (Level 3)
Linked Work Items	refers to :  SPPR-10072 - Component Exchange refers to :  SPPR-2601 - Interface has parent :  SPPR-11568 - Overview of rules _ is ruled by :  SPPR-3721 - Component exchange has an unique name _ is ruled by :  SPPR-3722 - Component port direction is consistent with functional exchange direction _ is ruled by :  SPPR-3992 - Component exchange is allocated to a physical link _ is ruled by :  SPPR-4203 - Component exchange has allocated functional exchanges

Capability






- Operational Capability
- System Capability
- Logical Capability Realisation
- Physical Capability Realisation

Type	 C2P-Capability
Layer	System Analysis (Level 3)
ID	SPPR-10086
Linked Work Items	refers to :  SPPR-2583 - System Capability has parent :  SPPR-11568 - Overview of rules _ is ruled by :  SPPR-3933 - Capability has at least one precondition _ is ruled by :  SPPR-3934 - Capability has at least one postcondition _ is ruled by :  SPPR-3986 - Capability has an unique name _ is ruled by :  SPPR-3989 - Capability has a textual description _ is ruled by :  SPPR-3991 - Capability relationships are logically consistent _ is ruled by :  SPPR-4198 - Capability has an involvement relationship to at least one actor _ is ruled by :  SPPR-10102 - Capability pre- and postcondition represents element state _ is ruled by :  SPPR-10108 - Capability has at least one scenario or at least one functional chain _ is ruled by :  SPPR-10246 - Capability has a realisation relationship _ is ruled by :  SPPR-10270 - Capability realisation has identical metadata

Operational Activity










Type	 C2P-Operational Activity
ID	SPPR-10220
Linked Work Items	<p>refers to :  SPPR-2565 - Operational Activity</p> <p>has parent :  SPPR-11568 - Overview of rules</p> <p>_ is ruled by :  SPPR-3665 - Function is allocated to a Structural entity</p> <p>_ is ruled by :  SPPR-3698 - Function has an unique name</p> <p>_ is ruled by :  SPPR-4201 - Function has at least one functional exchange to another function</p>

Operational Interaction

Type	 C2P-Operational Interaction
ID	SPPR-10222
Linked Work Items	<p>refers to :  SPPR-2047 - Functional Exchange</p> <p>has parent :  SPPR-11568 - Overview of rules</p> <p>_ is ruled by :  SPPR-3997 - Functional exchange has an unique name</p> <p>_ is ruled by :  SPPR-10099 - Functional exchange name represents the state</p>


Function








- System function
- Logical function
- Physical function

Type	 C2P-Function
ID	SPPR-10082
Linked Work Items	<p>refers to :  SPPR-2598 - Function</p> <p>has parent :  SPPR-11568 - Overview of rules</p> <p>_ is ruled by :  SPPR-3665 - Function is allocated to a Structural entity</p> <p>_ is ruled by :  SPPR-3698 - Function has an unique name</p> <p>_ is ruled by :  SPPR-3936 - Function name uses recommended verbs</p> <p>_ is ruled by :  SPPR-4201 - Function has at least one functional exchange to another function</p> <p>_ is ruled by :  SPPR-4213 - Function has a description</p> <p>_ is ruled by :  SPPR-10746 - Function has a realisation relationship</p>











Functional Exchange

- System Functional Exchange
- Logical Functional Exchange
- Physical Functional Exchange








Type	 C2P-Functional Exchange
Layer	System Analysis (Level 3)
ID	SPPR-10085

Linked Work Items	<p>refers to :  SPPR-2047 - Functional Exchange</p> <p>has parent :  SPPR-11568 - Overview of rules</p> <p>_ is ruled by :  SPPR-3723 - Functional exchange is allocated to a component exchange</p> <p>_ is ruled by :  SPPR-3938 - Functional exchange has consistent linking to function ports</p> <p>_ is ruled by :  SPPR-3997 - Functional exchange has an unique name</p> <p>_ is ruled by :  SPPR-4205 - Functional exchange has at least one allocated exchange item</p> <p>_ is ruled by :  SPPR-10099 - Functional exchange name represents the state</p>
-------------------	---





Exchange Item

Type	 C2P-Exchange Item
ID	SPPR-10105
Linked Work Items	<p>refers to :  SPPR-2025 - Exchange item</p> <p>has parent :  SPPR-11568 - Overview of rules</p> <p>_ is ruled by :  SPPR-3937 - Exchange item is allocated to a functional exchange</p> <p>_ is ruled by :  SPPR-4195 - Exchange item has an unique name</p> <p>_ is ruled by :  SPPR-4196 - Exchange item has an exchange mechanism</p> <p>_ is ruled by :  SPPR-4207 - Exchange item element has a unique name</p> <p>_ is ruled by :  SPPR-4208 - Exchange item element has a defined cardinality</p> <p>_ is ruled by :  SPPR-10107 - Exchange item has a realisation relationship</p> <p>_ is ruled by :  SPPR-10213 - Exchange item has at least one exchange item element or a textual description</p>

Scenario








Type	 C2P-Scenario
ID	SPPR-10098
Linked Work Items	<p>refers to :  SPPR-2066 - Exchange scenario</p> <p>has parent :  SPPR-11568 - Overview of rules</p> <p>_ is ruled by :  SPPR-3695 - Scenario has an unique name</p> <p>_ is ruled by :  SPPR-3696 - Scenario has a textual description</p> <p>_ is ruled by :  SPPR-3697 - Scenario has a precondition</p> <p>_ is ruled by :  SPPR-10173 - Scenario has a postcondition</p>

Functional Chain










Type	 C2P-Functional Chain
ID	SPPR-10097
Linked Work Items	<p>refers to :  SPPR-2043 - Functional Chain</p> <p>has parent :  SPPR-11568 - Overview of rules</p> <p>_ is ruled by :  SPPR-3718 - Functional chain has an unique name</p>

Class

Type	 Class
ID	SPPR-10104

Linked Work Items	refers to :  SPPR-2061 - Data Class has parent :  SPPR-11568 - Overview of rules _ is ruled by :  SPPR-3584 - Data object class has a textual description _ is ruled by :  SPPR-3662 - Data object class has an unique name _ is ruled by :  SPPR-3664 - Structured type has a at least one property _ is ruled by :  SPPR-4011 - Data object class property has an unique name _ is ruled by :  SPPR-4197 - Data object class property has a textual description
-------------------	--

State

Type	 C2P-State
ID	SPPR-10110
Linked Work Items	refers to :  SPPR-2054 - State refers to :  SPPR-2059 - State Transition has parent :  SPPR-11568 - Overview of rules _ is ruled by :  SPPR-3713 - State has a unique name _ is ruled by :  SPPR-3716 - State has a textual description _ is ruled by :  SPPR-3717 - State has at least one incoming transition and at least one outgoing transition _ is ruled by :  SPPR-10109 - Transition has a trigger _ is ruled by :  SPPR-10111 - State machine has a unique name






3 Rules for structure

3.1 Systems and Actors

Structural entity has a unique name

The name has the following further properties:

- is singular
- is written in the form <**Structural entity name**> (<**additional information**>)
- is written in Title case - the first letter of each word is capitalised and words are separated by spaces
- the last word of <Structural entity name> is a noun






Rationale	Each structural element has a distinctive name which is written in a consistent way.
Compliance	 Required
ID	SPPR-10061
Linked Work Items	rules :  SPPR-10060 - Actor rules :  SPPR-10062 - System rules :  SPPR-10221 - Operational Entity has parent :  SPPR-10069 - Systems and Actors
Example	- Railway Undertaking - Signalling System - Trackside Protection System - Point Control Supervision (Behaviour)

Structural entity has a textual description




The description has the following further properties:

- contains the overall purpose or a brief overview of responsibilities only

- is stored in the "Description" field
- Optional: contains a bulleted list of characteristics apart from functionality that define this structural entity as distinct from the others, e.g. level of safety criticality, an invented architectural layer it belongs to, and the value of any other splitting factors that were used to separate it from other entities.




Rationale	A brief summary of responsibilities/behaviour is allowed but this should not read like a list of functions, since this information should be defined only through the allocation of functions, not duplicated in a text description.
Compliance	 Required
ID	SPPR-10063
Linked Work Items	rules :  SPPR-10060 - Actor rules :  SPPR-10062 - System rules :  SPPR-10221 - Operational Entity has parent :  SPPR-10069 - Systems and Actors
Example	<p>Point Machine: "Represents any powered device that operates a point."</p> <p>Railway Undertaking: "Any public or private undertaking licensed according to EU directive 2012/34/EU, the principal business of which is to provide services for the transport of goods and/or passengers by rail with a requirement that the undertaking ensure traction; this also includes undertakings which provide traction only. Reference: directive 2012/34/EU"</p> <p>Railway Infrastructure Manager: "Any body or firm responsible in particular for establishing, managing and maintaining railway infrastructure, including traffic management and control-command and signalling; the functions of the infrastructure manager on a network or part of a network may be allocated to different bodies or firms. Reference: EU Directive 2012/34/EU"</p>

Actor has at least one realisation relationship

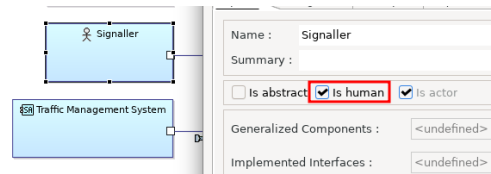
Rationale	An actor realises an actor of the next higher architectural layer via "realised actors".
Compliance	 Required
ID	SPPR-4007
Linked Work Items	rules :  SPPR-10060 - Actor has parent :  SPPR-10069 - Systems and Actors
Example	A logical actor realises a system actor via the realisation relationship.

Actor is configured as a human or external system

- Each human actor needs to be configured as Is Human and Is Actor
- Each external system needs to be configured as Is Actor.

Rationale	-
Compliance	 Required
ID	SPPR-4009
Linked Work Items	rules :  SPPR-10060 - Actor has parent :  SPPR-10069 - Systems and Actors

Example






3.2 Interfaces

Component exchange has a unique name

The name has following further properties:

- is written in the following way: **<required prefix><main part><optional suffix>**

Content	Details
<required prefix>	<p>interface to a non-human actor or between components: I_ interface to a human actor: HMI_ standardised communication interface: SCI_ standardised maintenance interface: SMI_ standardised diagnosis interface: SDI_ standardised security interface: SSI_</p>
<main part>	<p>In case of representing an existing standard, the component exchange is named after that considered standard:</p> <ul style="list-style-type: none"> • <name of the external standard> <p>In case of representing an external system interface to an actor, the component exchange is named after that considered actor:</p> <ul style="list-style-type: none"> • <name of actor> • <u>Note</u>: name of actor is written in PascalCase - all words are capitalised and written together <p>In case of representing an internal system interface between two components, the component exchange is named after the parent Structural entity:</p> <ul style="list-style-type: none"> • <name or acronym of the Structural entity>_<unique number>
<optional suffix>	<p>In case of representing additional information to distinguish further between component exchanges, the component exchange name can be extended using an optional label or number.</p> <ul style="list-style-type: none"> • <optional label>_<optional number> • <u>Note</u>: provides a custom label or number



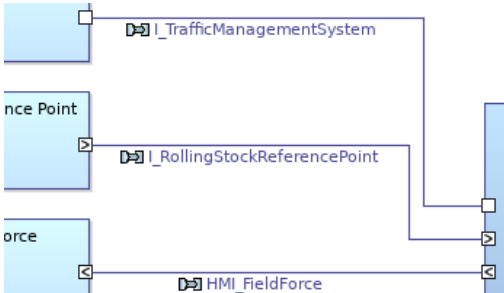
Rationale	The name is clearly distinguishable from the others and follows a consistent pattern.
Compliance	 Required
ID	SPPR-3721
Linked Work Items	<p>rules :  SPPR-10071 - Interface</p> <p>has parent :  SPPR-3720 - Interfaces</p>

Example	<p>Component exchange is named after the considered actor: I_TrafficManagementSystem HMI_Signaller</p> <p>Component exchange is named after the parent structural element: I_CCS_2</p> <p>Component exchange is named after an existing standard: SCI_P_Application (the suffix indicates the application layer according to the OSI/ISO model) I_SUBSET_139 (the name indicates the corresponding ETCS specification) I_TCP_Transport (the name indicates the TCP protocol as part of a layered interface)</p>
---------	--

Component port direction is consistent with functional exchange direction

The component port has following further properties:



- **in:** in case of incoming allocated functional exchanges only
- **out:** in case of outgoing functional exchanges only
- **inout:** in case of both incoming and outgoing functional exchanges

Rationale	The port directions of both ports are consistent and in line with the direction of the allocated functional exchanges.
Compliance	❗ Required
ID	SPPR-3722
Linked Work Items	rules :  SPPR-10071 - Interface has parent :  SPPR-3720 - Interfaces
Example	




Component exchange has allocated functional exchanges

A component exchange has functional exchanges.

Exception: In the case of representing an external standard or existing specification the component exchange doesn't have to have any functional exchange. This is important to avoid replication of existing content without any benefit.

Rationale	To provide a consistent allocation between functions and structural elements.
Compliance	❗ Required
ID	SPPR-4203
Linked Work Items	rules :  SPPR-10071 - Interface has parent :  SPPR-3720 - Interfaces
Example	SUBSET-139 includes different OSI layers for that interface represented by a component exchange (e.g. safe communication layer). It is for this not necessary to have an allocation of functional exchanges to that component exchange.



Component exchange is allocated to a physical link

Rationale	All physical component exchanges which cross the border of the system are allocated to a corresponding physical link. A physical link may transfer a specific kind of physical characteristics like power, force or information. Therefore, physical component exchanges must be compatible to these physical characteristics.
Compliance	 Recommended
ID	SPPR-3992
Linked Work Items	rules :  SPPR-10071 - Interface has parent :  SPPR-3720 - Interfaces
Example	Data exchanges using an application layer protocol are transmitted via an ethernet cable.

Physical link has an unique name

The name has following further properties:

- is freely chosen
- optional: can have the suffix "_Physical"

Rationale	The name is clearly distinguishable from the others and follows a consistent pattern.
Compliance	 Required
ID	SPPR-3928
Linked Work Items	has parent :  SPPR-3720 - Interfaces
Example	Twisted pair cable [IEEE 802.3ab] with RJ45 [IEC 60603-7] Optical fiber cable I_PlanningSystem_Physical




4 Rules for behaviour

4.1 Capabilities

Capability has an unique name

The name has the following further properties:




- is written in the form **<active verb> <noun>**
- is written in Sentence case - the first letter of the first word is capitalised and words are separated by spaces
- avoid verbs such as "manage", "handle" or "process" as they are not specific enough.

Rationale	The verb should be precise enough to clearly define the specific effect or outcome that one or more actors or stakeholders expect from the system or entity.
Compliance	 Required
ID	SPPR-3986
Linked Work Items	rules :  SPPR-10086 - Capability has parent :  SPPR-3984 - Capabilities
Example	- Set point position - Perform train movement - Activate usage restriction

Capability has a textual description

The description has the following further properties:


- is stored in the "Description" field
- follows this template: "The [ActorName] needs [structural entity] to [mandatory description of the operation on the primary object] [optional description of the influence of the secondary object] and [optional description of the involvement of the secondary actors to enable the operation on the primary object]."



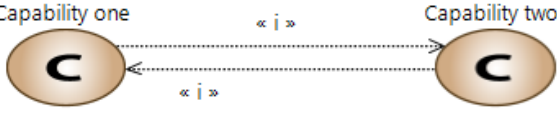
Rationale	The summary is a statement of the primary actor's / stakeholder's need to the system.
Compliance	 Required
ID	SPPR-3989
Linked Work Items	rules :  SPPR-10086 - Capability has parent :  SPPR-3984 - Capabilities
Example	The Traffic Management System needs the system to set a point to the position required by the operational plan movement.

Capability relationships are logically consistent


The capability has one or more of the following further properties:


- has realisation relationships to a capability or capability realisation on a higher level
- has include or extent relationships to another capability or capability realisation




Rationale	The set of relationships between capabilities are logically consistent and allows for no unintended overlaps between the scopes of different system capabilities.
Compliance	 Required
ID	SPPR-3991

Linked Work Items	rules :  SPPR-10086 - Capability has parent :  SPPR-3984 - Capabilities
Example	<p><<includes>> as a circular reference and a specific case of an inconsistency:</p> 


Capability has at least one precondition


Preconditions **only reflect system states** which are relevant in the context of the scenario. **Triggers of system events are not part** of preconditions, but can mentioned in the description because they help to understand the scenario's scope. Please consider  SPPR-10102 - Capability pre- and postcondition represents element state for more details.


Note: Where the capability has multiple preconditions, each precondition is included in its own line and joined to the other conditions with an AND or OR statement. Please consider also  SPPR-3952 - Conditions are written as logical expressions for the details on writing conditions.

Rationale	The precondition defines the prerequisites that must be true for the capability to begin.
Compliance	 Recommended
ID	SPPR-3933
Linked Work Items	rules :  SPPR-10086 - Capability has parent :  SPPR-3984 - Capabilities
Example	<p>[<precondition 1> OR <precondition 2> AND <precondition n></p> <p>One planned rolling stock movement extends over the Point AND The estimated point position is not in the required end position as indicated to the Signaller</p>

Capability has at least one postcondition

Postconditions **only reflect system states** which are relevant in the context of the scenario. **Triggers of system events are not part** of postconditions, but can mentioned in the description because they help to understand the scenario's scope. Please consider  SPPR-10102 - Capability pre- and postcondition represents element state for more details.

Note: Where the capability has multiple possible postconditions (a main success condition and one or more failure/diminished success conditions) then each separate postcondition should be included separated with the logical operator "OR" and the main success condition being the first listed condition. Please consider also  SPPR-3952 - Conditions are written as logical expressions for the details on writing conditions.

Rationale	The postcondition defines the desired outcome that must be true at the end of the capability execution.
Compliance	 Recommended
ID	SPPR-3934

Linked Work Items	rules : 🏷️ SPPR-10086 - Capability has parent : 📁 SPPR-3984 - Capabilities
Example	<main success condition> OR <second success condition> OR <failure condition> The usage restriction is activated AND Signaller and Traffic Management System are informed about the activation status.

Capability pre- and postcondition represents element state

The pre- and postcondition have the following further properties:

- is written in the form <entity> is <state>

Rationale	Entity could be any actor or a data concept and state is one of the defined states that the entity is allowed to occupy. At the postcondition of each capability, a measurable or observable result has been delivered to at least one entity or actor. Allow predefined value to avoid ambiguity in definition of the starting point or result of a capability
Compliance	🟢 Recommended
ID	SPPR-10102
Linked Work Items	rules : 🏷️ SPPR-10086 - Capability has parent : 📁 SPPR-3984 - Capabilities
Example	estimated point position is not in the required end position

Capability has a realisation relationship




Rationale	The capabilities and their relationships between the different levels of detail are set consistently to provide traceability.
Compliance	🔴 Required
ID	SPPR-10246
Linked Work Items	rules : 🏷️ SPPR-10086 - Capability has parent : 📁 SPPR-3984 - Capabilities
Example	A logical capability realisation realises a system capability via the realisation relationship. A physical capability realisation realises logical capability realisation via the realisation relationship.

Capability realisation has identical metadata





The capability realisation has the following further properties:

- has the same name, description, relationships, precondition and postconditions as the corresponding capability or capability realisation on higher level it represents




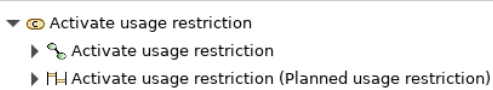
Rationale	The capabilities are defined consistently across the different levels of detail. The metadata is broadly identical to the original capability or capability realisation, except for those parts that make this refined capability realisation distinct from the other refined capability realisations.
-----------	--

Compliance	 Recommended
ID	SPPR-10270
Linked Work Items	rules :  SPPR-10086 - Capability has parent :  SPPR-3984 - Capabilities
Example	A logical capability realisation has similar metadata (e.g. name, description) like the corresponding system capability. A physical capability realisation has similar metadata (e.g. precondition, actor involvement relationships) like the corresponding logical capability realisation.

Capability has an involvement relationship to at least one actor

Rationale	Each capability has an involvement relationship to at least one actor, except where a capability is included by other capabilities (in which case, a direct involvement with an actor is not necessary, but is allowed). When a capability is specialised in that case, it inherits the actor from the super-type capability.
Compliance	 Required
ID	SPPR-4198
Linked Work Items	rules :  SPPR-10086 - Capability has parent :  SPPR-3984 - Capabilities
Example	

Capability has at least one scenario or at least one functional chain

Rationale	Each capability has at least one behavioural description using scenarios or functional chains.
Compliance	 Required
ID	SPPR-10108
Linked Work Items	rules :  SPPR-10086 - Capability has parent :  SPPR-3984 - Capabilities
Example	

4.2 Functions

Function has an unique name

The name has the following further properties:

- is written in the form **<verb> <noun>**
- the noun is the object upon which the function operates
- is written in Sentence case - the first letter of the first word is capitalised and words are separated by spaces
- the article "the" before the noun is skipped in order to keep the whole function name shorter
- in case of multiple verbs, the name has the format **<verb>[+<optional second verb>] <noun>**

If there is the special case of a passive function (e.g. of a system actor) which provides actual physical values to the system of interest, the name is written in the form **"has <property>"**. The property can contain free text.



Rationale	The names are consistent across all functions and are written in the same way.
Compliance	❗ Required
ID	SPPR-3698
Linked Work Items	rules : 🟢 SPPR-10082 - Function rules : 🟡 SPPR-10220 - Operational Activity has parent : 📄 SPPR-3705 - Functions _ is parent of : ❗ SPPR-3936 - Function name uses recommended verbs
Example	See examples in ❗ SPPR-3936 - Function name uses recommended verbs

Function name uses recommended verbs

The following verbs are recommended to be used for the names of functions.

Purpose	Name
Transformation of a physical external state into data information	Sense / Measure / Capture / Perceive (e.g. Sense data about environment)
Observations or estimations or making inferences about the state of the external plant given incoming data	Observe / Estimate / Calculate / Monitor / Evaluate / Determine (e.g. Observe position of one point)
Transformation of instructions or commands into a physical state	Actuate / Move / Sound / Lock (e.g. Sound horn)
Decision making about a required state change, instruction or command	Control / Command / Restrict / Acknowledge / Calculate / Generate / Request / Supervise / Authorise (e.g. Control motion)
Conversion of data information into a human-perceptible form such as audible, visual or haptic	Indicate / Display / Illuminate / Generate (e.g. Display the position of one point)
Storage, delivery or change of data	Store / Update / Communicate / Process / Maintain <kind of data> (e.g. Maintain map data) Transmit / Send / Publish / Encode / Produce (e.g. configuration data)
Provision of the original source of data	Provide <kind of data> (e.g. configuration or calibration data, Provide wheel diameter)
Usage of energy	Accumulate / Liberate / Transform
Processing, usage and supply of materials	Store / Procure / Transport / Transform




Rationale	The names are consistent across all functions and are written in the same way.
Compliance	✅ Recommended
ID	SPPR-3936

Linked Work Items	rules :  SPPR-10082 - Function has parent :  SPPR-3698 - Function has an unique name
Example	Control motion of one train Sense+Observe train position Has actual speed





Function has a description

The description has the following further properties:





- follows the pattern: "**This function <verb> ...**"
- is stored in the "Description" field
- optional: why the inputs are required
- optional: how the inputs may be used to produce the outputs

Rationale	The description explains what the function is for, i.e. which outputs the function will produce based on the inputs.
Compliance	 Required
ID	SPPR-4213
Linked Work Items	rules :  SPPR-10082 - Function has parent :  SPPR-3705 - Functions
Example	This function identifies the need to switch the position of one point along the planned movement of the rolling stock and commands, so that the position is correct for the intended plan to the right point in time.

Function is allocated to a Structural entity

Rationale	A function is allocated to either the system, a component or to an actor where some behaviour is expected in order to reach the end conditions of a capability.
Compliance	 Required
ID	SPPR-3665
Linked Work Items	rules :  SPPR-10082 - Function rules :  SPPR-10220 - Operational Activity has parent :  SPPR-3705 - Functions
Example	-

Function has at least one functional exchange to another function

Rationale	There should be no function allocated to a structural element which is unreachable.
Compliance	 Required
ID	SPPR-4201
Linked Work Items	rules :  SPPR-10082 - Function rules :  SPPR-10220 - Operational Activity has parent :  SPPR-3705 - Functions
Example	-

Function has a realisation relationship

Between different levels of details (model layers).

Rationale	To ensure consistency and provide traceability.
-----------	---

Compliance	❗ Required
ID	SPPR-10746
Linked Work Items	rules : 🟢 SPPR-10082 - Function has parent : 📁 SPPR-3705 - Functions
Example	A logical function realises a system function via the realisation relationship. A physical function realises a logical function via the realisation relationship.

4.3 Functional exchanges

Functional exchange has an unique name

The name has the following further properties:

- begins with a **<verb as participle>**
- is written in Sentence case - the first letter of the first word is capitalised and words are separated by spaces

Rationale	The name is clearly distinguishable from the others and follows a consistent pattern.
Compliance	❗ Required
ID	SPPR-3997
Linked Work Items	rules : 🟢 SPPR-10085 - Functional Exchange rules : 🟡 SPPR-10222 - Operational Interaction has parent : 📁 SPPR-11319 - Functional exchanges _ is parent of : ❗ SPPR-10099 - Functional exchange name represents the state
Example	Estimated position state of one train

Functional exchange name represents the state

The name has the following further properties:

- is written in the form **<state of exchange type> of**



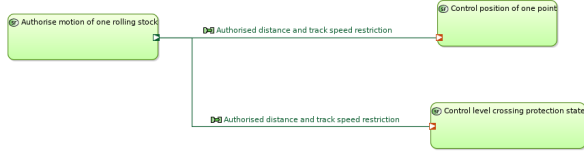
Exception: The adjective "actual" is allowed to express the actual state of the function output. Multiple instances of the same type of functional exchange has identical names but it is permissible to add a unique suffix to them if this improves the descriptiveness of the model.

Rationale	It is recommended to use the naming pattern "<exchange type> of <state>" of functional exchanges between functions.
Compliance	✅ Recommended
ID	SPPR-10099
Linked Work Items	rules : 🟢 SPPR-10085 - Functional Exchange rules : 🟡 SPPR-10222 - Operational Interaction has parent : ❗ SPPR-3997 - Functional exchange has an unique name
Example	Estimated position state of one point

Functional exchange has consistent linking to function ports

Depending on the case, the functional exchange has the following further properties:



- each **different incoming** functional exchange enters the function via a **separate** function input port
- each **different outgoing** functional exchange exits the function via a **separate** function output port
- each **identical incoming** functional exchange enters the function via the **same** function input port
- each **identical outgoing** functional exchange exits the function via the **same** function output port

Rationale	The functional exchanges are linked to function ports in a consistent way.
Compliance	❗ Required
ID	SPPR-3938
Linked Work Items	rules :  SPPR-10085 - Functional Exchange has parent :  SPPR-11319 - Functional exchanges
Example	<p>The following function has two identical outgoing functional exchanges which are sharing the same function output port.</p> 

Functional exchange has at least one allocated exchange item

Exception 1: If the associated exchange item represents real-world interaction, then it does not have to have a fully-specified data model behind to avoid re-modelling the full complexity of the real world for no benefit.



Exception 2: Functional exchanges between two actor functions do not necessarily need exchange items.

Rationale	Each instance of a functional exchange of a given type only conveys those exchange items that are used by the sink function. This means that multiple instances of the same type of functional exchange are allowed to carry different sets of exchange items, provided that all the exchange items are legitimate for that functional exchange type.
Compliance	❗ Required
ID	SPPR-4205
Linked Work Items	rules :  SPPR-10085 - Functional Exchange has parent :  SPPR-11319 - Functional exchanges
Example	-

Functional exchange is allocated to a component exchange

The functional exchanges has the following further properties:

- Case 1: Functional exchanges which are existing between different system components or between the system and an actor are allocated to a component exchange.
- Case 2: Functional exchanges which are existing inside one system component are not allocated to a component exchange.




Rationale	The usage of functional exchanges are consistent, i.e. there is no unallocated functional exchange between internal system components or between the system and its actors.
Compliance	❗ Required
ID	SPPR-3723
Linked Work Items	rules :  SPPR-10085 - Functional Exchange has parent :  SPPR-11319 - Functional exchanges
Example	-

4.4 Scenarios

Scenario has an unique name

The name has the following further properties:




- is written in the form **<capability name> (<optional information>)**
- the optional information describes an additional purpose of the scenario
- is written in Sentence case - the first letter of the first word is capitalised and words are separated by spaces

Rationale	The name gives a detail on the individual nature of the particular scenario (that is, what differentiates this scenario from other scenarios).
Compliance	 Required
ID	SPPR-3695
Linked Work Items	rules :  SPPR-10098 - Scenario has parent :  SPPR-6058 - Scenarios
Example	Set point position (Signaller request position change)

Scenario has a textual description

The description has the following further properties:

- is stored in the "Description" field

Rationale	The description gives a detail on the individual nature of the particular scenario (that is, what differentiates this scenario from other scenarios).
Compliance	 Recommended
ID	SPPR-3696
Linked Work Items	rules :  SPPR-10098 - Scenario has parent :  SPPR-6058 - Scenarios
Example	This scenario describes the point movement from left end position to right end position request by the Signaller.


Scenario has a precondition



The precondition has the following further properties:

- If the precondition of the owning capability fully applies to the considered scenario, then the precondition is taken from the owning capability.
- If the precondition of the owning capability do not (fully) apply to the considered scenario, then the precondition is defined suitable for the considered scope of the scenario.

Preconditions **only reflect system states** which are relevant in the context of the scenario. **Triggers of system events are not part** of preconditions, but can mentioned in the description because they help to understand the scenario's scope.

Note: Additional preconditions provide specific context for a scenario and represent a refinement or more concrete conditions that apply to this specific scenario.

Rationale	The precondition defines the prerequisites that must be true for scenario to begin.
Compliance	 Required
ID	SPPR-3697

Linked Work Items	rules :  SPPR-10098 - Scenario has parent :  SPPR-6058 - Scenarios
Example	One planned rolling stock movement extends over the Point AND Estimated point position is in right end position AND Required point position is left end position.




Scenario has a postcondition

The postcondition has the following further properties:

- If the postcondition of the owning capability fully applies to the considered scenario, then the postcondition is taken from the owning capability.
- If the postcondition of the owning capability do not (fully) apply to the considered scenario, then the postcondition is defined suitable for the considered scope of the scenario.

Postconditions **only reflect system states** which are relevant in the context of the scenario. **Triggers of system events are not part** of postconditions, but can mentioned in the description because they help to understand the scenario's scope.

Note: Additional postconditions provide specific context for a scenario and represent a refinement or more concrete conditions that apply to this specific scenario.




Rationale	The postcondition defines the desired outcome that must be true at the end of the scenario.
Compliance	 Required
ID	SPPR-10173
Linked Work Items	rules :  SPPR-10098 - Scenario has parent :  SPPR-6058 - Scenarios
Example	Estimated position of the point is left as the required end position to execute the movement AND Estimated position of the point is indicated to the Signaller.

4.5 Functional chains

Functional chain has an unique name

The name has the following further properties:

- is written in the form: **<capability name> (<optional information>)**
- the optional information describes an additional purpose of the functional chain
- is written in Sentence case - the first letter of the first word is capitalised and words are separated by spaces

Rationale	The name is clearly distinguishable from the others and follows a consistent pattern.
Compliance	 Required
ID	SPPR-3718
Linked Work Items	rules :  SPPR-10097 - Functional Chain has parent :  SPPR-4025 - Functional chains
Example	Set point to position

4.6 State machines

State machine has a unique name

Rationale	The name is clearly distinguishable from the others.
Compliance	❗ Required
ID	SPPR-10111
Linked Work Items	rules : 📄 SPPR-10110 - State has parent : 📄 SPPR-10210 - State machines
Example	-

State has a unique name

The name has the following further properties:

- is written in Sentence case - the first letter of the first word is capitalised and words are separated by spaces
- is written using adjectives (-ed), referring to a quality of the owning entity/actor OR
- is written using verbs in the present participle form (-ing) OR
- is using verbatim copies of states defined externally to the project (that is, when reusing state definitions from elsewhere, they should be reproduced precisely and not changed)

Rationale	The name is clearly distinguishable from the others and follows a consistent pattern.
Compliance	❗ Required
ID	SPPR-3713
Linked Work Items	rules : 📄 SPPR-10110 - State has parent : 📄 SPPR-10210 - State machines
Example	Operating, Initialising, Disconnected, Suspended

State has a textual description

The description has the following further properties:

- is stored in the "Description" field

Rationale	The description has a brief description of the qualities that define this state as distinct from the others. This may include a summary of the distinct behaviour of the entity that is exposed in this state.
Compliance	✅ Recommended
ID	SPPR-3716
Linked Work Items	rules : 📄 SPPR-10110 - State has parent : 📄 SPPR-10210 - State machines
Example	Examples will be added at a later stage.

State has at least one incoming transition and at least one outgoing transition

Note: In case of an initial state, the state has only one outgoing transition. In case of a final state, the state has only at least one incoming transition.




Rationale	Every state is reachable and has no dead ends.
Compliance	❗ Required
ID	SPPR-3717
Linked Work Items	rules : 📄 SPPR-10110 - State has parent : 📄 SPPR-10210 - State machines

Example	-
---------	---

Transition has a trigger

A state transition has a definition of its conditions:

- The trigger definition refers to exchange items or change events.
- The boolean expression for change events are textual and can be written in natural language.
- The guard condition qualifies and indicates the condition to perform the change of state

Rationale	Each transition has a distinct trigger.
Compliance	 Required
ID	SPPR-10109
Linked Work Items	rules :  SPPR-10110 - State has parent :  SPPR-10210 - State machines
Example	Examples will be added at a later stage.

5 Rules for exchange items

Exchange item has an unique name

The name has the following further properties:

- is written in snake_case - all letters are lower case and words separated by underscores
- use nouns at the end

Rationale	The name is clearly distinguishable from the others and follows a consistent pattern.
Compliance	❗ Required
ID	SPPR-4195
Linked Work Items	rules : ➡ SPPR-10105 - Exchange Item has parent : 📖 SPPR-5701 - Rules for exchange items
Example	point_position_report movement_authority_request

Exchange item has an exchange mechanism

The mechanism has one of the following values:

- **UNSET**: in the first stages of the engineering process, the communication mechanism applied to a given exchange Item might not be known or decided upon;
- **FLOW**: flow of continuously available data, matter, energy or light;
- **EVENT**: asynchronous mechanism where an event is sent by an element and received by one or several other.

Rationale	The exchange mechanism is selected in a consistent way based on the intended characteristics of the exchange item.
Compliance	❗ Required
ID	SPPR-4196
Linked Work Items	rules : ➡ SPPR-10105 - Exchange Item has parent : 📖 SPPR-5701 - Rules for exchange items
Example	"movement_authority_request" has the exchange mechanism "EVENT".



Exchange item is allocated to a functional exchange

Rationale	Each functional exchange conveys exchange items that are used by the target function.
Compliance	❗ Required
ID	SPPR-3937
Linked Work Items	rules : ➡ SPPR-10105 - Exchange Item has parent : 📖 SPPR-5701 - Rules for exchange items
Example	-

Exchange item has at least one exchange item element or a textual description

Note: If the associated exchange item represents real-world interaction or represents exchanged information on a higher level of abstraction, then it does not have to have a fully-specified data model.




Rationale	The exchange item has a clear purpose.
Compliance	❗ Required

ID	SPPR-10213
Linked Work Items	rules :  SPPR-10105 - Exchange Item has parent :  SPPR-5701 - Rules for exchange items
Example	"movement_authority_request" has the exchange item element called "requiredPermission".




Exchange item element has a unique name

The name has the following further properties:




- is written in camelCase - name starts with a lower case letter, each further proper word in the name is capitalised and words are written together

Rationale	The name is clearly distinguishable from the others and follows a consistent pattern.
Compliance	 Required
ID	SPPR-4207
Linked Work Items	rules :  SPPR-10105 - Exchange Item has parent :  SPPR-5701 - Rules for exchange items
Example	"requiredPermission" is an exchange item element of "movement_authority_request".

Exchange item element has a defined cardinality

Rationale	Each exchange item element has a defined cardinality, i.e. how many instances of that element are used by the exchange item.
Compliance	 Required
ID	SPPR-4208
Linked Work Items	rules :  SPPR-10105 - Exchange Item has parent :  SPPR-5701 - Rules for exchange items
Example	By default, the cardinality is set to 1; that is, an exchange Item has one field of the specified type. Some exchange items need to be defined such that the cardinality is variable depending on the number of classes being transmitted.

Exchange item has a realisation relationship



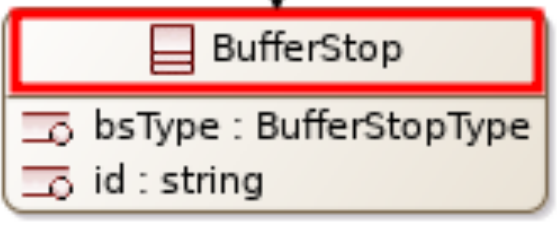
Rationale	Where an exchange item implements or realises a higher level exchange item, there exists a realisation relationship between realising exchange item and realised exchange item.
Compliance	 Recommended
ID	SPPR-10107
Linked Work Items	rules :  SPPR-10105 - Exchange Item has parent :  SPPR-5701 - Rules for exchange items
Example	-

6 Rules for data types and classes

Data object class has an unique name

The name has the following further properties:



- is singular
- is written in PascalCase - all words are capitalised and written together and the first letter is capitalised

Rationale	The name is clearly distinguishable from the others.
Compliance	❗ Required
ID	SPPR-3662
Linked Work Items	rules :  SPPR-10104 - Class has parent :  SPPR-3715 - Rules for data types and classes
Example	

Data object class has a textual description

The description has the following further properties:



- is stored in the "Description" field






Rationale	The data object class has an appropriate description of the purpose of the data object.
Compliance	✓ Suggested
ID	SPPR-3584
Linked Work Items	rules :  SPPR-10104 - Class has parent :  SPPR-3715 - Rules for data types and classes
Example	Data object for buffer stop on the track.

Data object class property has an unique name

The name has the following further properties:

- is written in camelCase - name starts with a lower case letter, each further proper word in the name is capitalised and words are written together
- is also applicable to roles and attributes




Rationale	The data object class has a property which is defined in a consistent way. The names should be chosen in a way that they are understandable. The name is clearly distinguishable from the others.
Compliance	❗ Required
ID	SPPR-4011
Linked Work Items	rules :  SPPR-10104 - Class has parent :  SPPR-3715 - Rules for data types and classes

Example	<div style="background-color: #4a7ebb; color: white; padding: 2px; margin-bottom: 5px;">  BufferStop </div> <ul style="list-style-type: none"> ▶  [1..*] linearElementSections : LinearElementSection ▶  [*] dangerPoints : DangerPoint ▶  bsType : BufferStopType ▶  id : string
---------	--

Data object class property has a textual description

The description has the following further properties:



- is stored in the "Description" field

Rationale	Each property has an adequate description of the purpose of the data object class property.
Compliance	 Suggested
ID	SPPR-4197
Linked Work Items	rules :  SPPR-10104 - Class has parent :  SPPR-3715 - Rules for data types and classes
Example	bsType: Defines buffer stop type.

Simple type has a unique name

The following further naming conventions are applied for the different simple types:

Type	Rule
BooleanType NumericType StringType PhysicalQuantity Enumeration	PascalCase - all words are capitalised and written together and the first letter is capitalised
Enumeration Literal NumericLiteral BooleanLiteral	Use one of the following two writing styles: <ul style="list-style-type: none"> • PascalCase - all words are capitalised and written together and the first letter is capitalised • camelCase - starts with a lower case letter, each further proper word in the name is capitalised and words are written together

Rationale	The name is clearly distinguishable from the others and follows a consistent pattern.
Compliance	 Required
ID	SPPR-4215
Linked Work Items	has parent :  SPPR-3715 - Rules for data types and classes
Example	Basic computing types are String, Numeric or Boolean.

Simple type name uses SI units

The Capella elements of type "Unit" get the unit's symbol as the name and the written-out name as the summary. Choose between units for the same physical quantity:

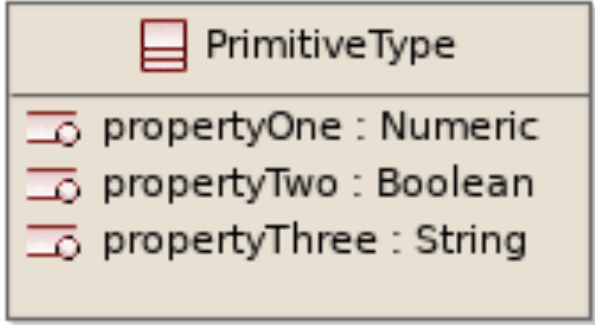
- Case 1: The SI-coherent unit, if the field of application is new or mostly closed.
- Case 2: A non-coherent unit, if an external standard or common practise prefers this unit against the coherent unit.

Rationale	Use only units that are part or accepted by the SI which can be found here: https://www.bipm.org/en/publications/si-brochure
Compliance	❗ Required
ID	SPPR-10103
Linked Work Items	has parent : 📖 SPPR-3715 - Rules for data types and classes
Example	m/s (unit's symbol as the element name) metre/second (unit's written-out name as the element summary) km/h (unit's symbol used in railway common practise and ETCS language)

Structured type has a unique name

The name has the following further properties:

- is singular
- is written in PascalCase - all words are capitalised and written together and the first letter is capitalised

Rationale	The name is clearly distinguishable from the others and follows a consistent pattern.
Compliance	❗ Required
ID	SPPR-4214
Linked Work Items	has parent : 📖 SPPR-3715 - Rules for data types and classes
Example	

Structured type has a at least one property

The name has the following further properties:

- the chosen class which represents the structured type is specified as "Primitive"
- structured types and primitive classes can have properties and can be used as type of a property in another class or in another structured type

Rationale	Concept of primitive class corresponds to that of DataType in UML, whose instances have no identity, unlike for classes.
Compliance	❗ Required
ID	SPPR-3664
Linked Work Items	rules : 📖 SPPR-10104 - Class has parent : 📖 SPPR-3715 - Rules for data types and classes
Example	-

7 Common rules

7.1 Language conventions

British English is used with correct grammar and spelling

Refer to the Cambridge Dictionary or the Oxford Dictionary for further information.

<https://dictionary.cambridge.org/>

<https://www.oed.com/>

<https://www.oxfordlearnersdictionaries.com/>

Rationale	British English should be used at all times, in order to use correct grammar and ensure consistency within the project.
Compliance	❗ Required
ID	SPPR-3945

Abbreviations are avoided

If there is special need to use an abbreviation in a requirement though, then a link or glossary reference to its definition must be provided. Please consider the *SPPROCESS/30 SP Metadata Management/Glossary Usage Guidelines : 722475* for the details.

Note: When there are abbreviations which have different or multiple meanings, it is a best practice to avoid ambiguity by spelling out the abbreviation.

Rationale	Abbreviations should be avoided because of unambiguity in elements used in requirements.
Compliance	✅ Recommended
ID	SPPR-3958

7.2 Rationales and assumptions

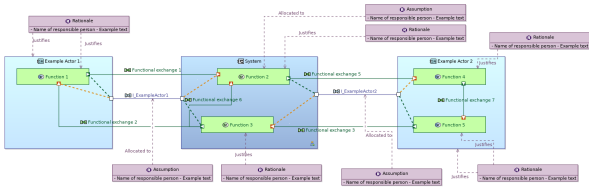
Rationales are justifying design decisions for model elements


The rationale has the following further properties:

- is defined using the requirement model element with selected type "Rationale"
- has at least one outgoing link of type "Justifies" to the model element which it pertains
- includes the name of the responsible person inserted in the field "Name"
- has a description which is inserted in the field "Text" describing only design decisions and not the functionality of the model element being justified (if a rationale expands a decision that is documented outside of the model, e.g. a trade-off analysis record, the rationale description should include a reference to the documentation, including (where possible) its document number and a hyperlink to the current version)

Rationale	Design decisions are clearly documented and comprehensible.
Compliance	❗ Required
ID	SPPR-3438
Linked Work Items	has parent : 📁 SPPR-8972 - Rationales and assumptions _ is referred by : 📄 SPPR-10053 - How to create rationale and assumptions in Capella

Example







Further example see:  SPPR-10053 - How to create rationale and assumptions in Capella

Assumptions are allocated to model elements

The assumption has the following further properties:



- is defined using the requirement model element with selected type "Assumption"
- includes the name of the responsible person inserted in the field "Name"
- has a description which is inserted in the field "Text"
- has at least one outgoing link of type "allocated to" to the architecture element which it pertains

Rationale	Assumptions are clearly documented and comprehensible.
Compliance	 Required
ID	SPPR-10052
Linked Work Items	has parent :  SPPR-8972 - Rationales and assumptions _ is referred by :  SPPR-10053 - How to create rationale and assumptions in Capella
Example	See example in  SPPR-3438 - Rationales are justifying design decisions for model elements.

7.3 Tool-related considerations

Auto links are used in element descriptions



These model element description links are used in all descriptions within the model.

Rationale	If a model element description refers to another model element, the name of this model element is referred to in form of some interactive link automatically updated or maintained by the tool. The auto links are then translated into work item links and ease the traceability between model elements as work items.
Compliance	 Recommended
ID	SPPR-3435
Linked Work Items	has parent :  SPPR-9480 - Tool-related considerations
Example	The <actor name> need the <system name> to <system capability name>. Each element in "<>" is represented there as an auto link.


7.4 Conditions


Conditions are written as logical expressions

Conditions are represented consistently by logical expressions which have the following conventions:

- consists of single expression or multiple expressions **separated by logical operators**
- uses the logical operators **NOT, AND, OR, XOR** written in upper case
- are used explicitly in preconditions and triggers for  SPPR-3519 - Functional requirement pattern is used for system behaviour and  SPPR-4332 - Non-functional requirement pattern is used for system characteristics


There are three different cases how to write conditions:


Detailed case	Specific rule
single expression with no logical operator	<p>Do not use square brackets for the expression.</p> <p><expressionA></p> <p>A single expression is written as following  SPPR-3946 - Values are expressed precisely .</p>
multiple expressions with the same logical operator	<p>Put square brackets around the whole expression including all logical operators. Do not put every condition part in a particular square bracket pair connected by the logical operators. This is for the situations where i.e., only AND or only OR is used.</p> <p>[<expressionA> <logical operator 1> <expressionB> <logical operator 1> ...]</p> <p><u>Exception:</u> When the condition is written inside a <u>Capella2Polarion</u> work item, the outer square brackets can be omitted for the whole condition.</p>
multiple expressions with different logical operators	<p>Put square brackets around every expression part that you want to connect with the logical operator. Furthermore, put a blank between subsequent nested square brackets for readability. This is for the situations where i.e., combinations of AND, OR are used.</p> <p>[[<expressionA>] <logical operator 1> [<expressionB>] <logical operator 2> <expressionC> ...]</p> <p><u>Note:</u> Please do not rely on the logical operator precedence known from first-order logic (e.g., a OR b AND c = [a OR [b AND c]]), because other people have to understand the requirements and might have other interpretations. Therefore, place the brackets in the way the conditions within brackets shall be evaluated. This means that the most inner brackets are evaluated first.</p> <p><u>Exception:</u> When the condition is written inside a <u>Capella2Polarion</u> work item, the outer square brackets can be omitted for the whole condition.</p>

Rationale	Conditions are described in a consistent way to control the scope using brackets and logical operators.
Compliance	 Required
ID	SPPR-3952

Conditions are reusable


Use work items of type {c} SPPR-11193 - Condition to write reusable conditions.

Insert the {c} SPPR-11193 - Condition work item as an active link in the requirement sentence. See  SPPR-3952 - Conditions are written as logical expressions for the details how to write conditions (i.e. for preconditions, triggers).

Rationale	Conditions can be written and managed separately using Constraint work items for the purpose of reusability in multiple requirements.
Compliance	 Recommended
ID	SPPR-11195

8 Appendix

8.1 Construction and interpretation methods

The construction and interpretation methods are defined in  Systems Engineering Management Plan - Annex M2 Viewpoint Guidelines.

8.2 Compliance report

The following rules are checked currently by the compliance report available in the Capella model explorer. In a later version of this document it is planned to merge these rules with already defined ones earlier in this document.

Summary of common rules which are checked automatically by the current compliance report

This is the collection of rules that apply to more than one object type across multiple modeling layers.

- Object has a description or a summary
 - checked for: Operational entity, Operational capability, Operational activity, System capability, System actor, Logical component, Behaviour physical component, Function, State
- Behavior name follows verb-noun pattern
 - checked for: Operational capability, Operational activity, System function
- Capability involves an entity / actor
 - checked for: Operational capability, System capability
- All Functional exchanges shall be allocated to at least one component exchange
 - checked for Functional exchange
- Each component exchange shall have at least one functional exchange allocated to it
 - checked for: Component exchange
- A Behaviour element shall be allocated to a structure element
 - checked for: Operational activity, System function
- Capability should represent a specific behavior described by scenarios or chains
 - checked for: Operational capability, System capability

[SPPR-9337]

Summary of system analysis rules which are checked automatically by the current compliance report

This is the collection of rules that apply to the system analysis layer.

- System capability
 - involves at least one Actor Function and one System function
 - each Actor contributes at least one Function
 - each Function is allocated to an appropriate Actor or System
- System function
 - is connected to another Entity's function (System, Actor)
 - has an input port and an output port

[SPPR-9345]