




ERJU SYSTEM PILLAR

Systems Engineering Management Plan - 02 MBSE Methodology Handbook



Systems Engineering Management Plan - 02

MBSE Methodology Handbook

Author(s)	Jorge Block , ANTOONS Gilles , EDDOUS Sayfeddine (SNCF RESEAU / Directions Techniques Réseau / DGII DTR GE SF Solutions) , SANGO Marc (SNCF / DIR TECHNOLOGIES INNOVATION ET PROJETS GROUPE / IR DIR RECHERCHE - PSF)
Abstract	The MBSE Methodology Handbook is part of Systems Engineering Management Plan and is intended to explain the MBSE methodology adopted for System Pillar. It stipulates the modelling methods and rules, the processes and modelling tools used for System Pillar.
Config Item	System Engineering Management Plan
Document ID	SEMP Annexes/System Pillar MBSE Methodology Handbook#722568  Systems Engineering Management Plan - 02 MBSE Methodology Handbook
Classification	Public
Status	Released
Version	1.0
Revision	722568
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

0.1 27.09.2024	Jorge Block	Reviewed version (initial)
0.2 14.03.2025	Jorge Block	Reviewed version including Findings from Review 0.1
1 22.09.2025	EDDOUS Sayfeddine (SNCF RESEAU / Directions Techniques Réseau / DGII DTR GE SF Solutions)	Approved version based on Review 0.2
1.0 24.09.2025	EDDOUS Sayfeddine (SNCF RESEAU / Directions Techniques Réseau / DGII DTR GE SF Solutions)	Approved version based on Review 1

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

Type of Approval	 Document Review
Comments	<p>#1 Approval comment by Renard, Marie Pierre (SMO RI MT FR ADC TGMTR3) on 2025-09-24 12:06</p> <p>Approval is granted with limitations. While the overall structure and intent of the Links Roles are aligned with our objectives, the current naming conventions lack clarity and consistency. I recommend refining the role names to better reflect their responsibilities and to ensure alignment with standard terminology across systems. This will improve maintainability and reduce potential confusion for future stakeholders.</p>
Approvals	<p>ANTOONS Gilles : Approved , EDDOUS Sayfeddine (SNCF RESEAU / Directions Techniques Réseau / DGII DTR GE SF Solutions) : Approved , Dennis Kunz : Approved , Renard, Marie Pierre (SMO RI MT FR ADC TGMTR3) : Approved</p>
Attachments	MBSE_Handbook_comments_export.xls

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


Type of Approval	 Document Approval
Approvals	<p>Schmidt Steffen (I-NAT-GST-ERTM) : Approved , SCHWAN Nico : Approved , SANGO Marc (SNCF / DIR TECHNOLOGIES INNOVATION ET PROJETS GROUPE / IR DIR RECHERCHE - PSF) : Approved , KEFALAS, Georgios : Approved</p>

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 Roles	6
3 System Pillar MBSE Methodology Overview	6
3.1 Mapping of terminology to standards and norms	11
4 Processes	11
4.1 Requirements Management Process	11
4.2 Requirements Definition Process	12
4.3 Operational / Stakeholder Needs Analysis Process	12
4.4 Architecture Definition Process	12
4.4.1 Trade-off analysis guideline	12
4.5 Information Model definition process	12
5 MBSE Tools	15
6 Modelling Methods, Rules and Guidelines	15
7 Linking Rules for MBSE - Capella to Polarion work items	16
7.1 Common linking rules	17
7.2 Operational Analysis linking rules	17
7.3 System Definition and Architecture linking rules	18
7.4 Operational Design linking rules	22
7.4.1 Manually created workitems link rules	22
8 Appendix	24
8.1 Standards and References	24
8.2 Open actions and issues	25

1 Preamble

1.1 Purpose

Purpose of the document

This document is intended to explain the MBSE methodology adopted for System Pillar. It stipulates the modelling methods and rules, the processes and modelling tools used for System Pillar. [SPPR-11717]

1.2 Intended Audience

Intended audience

The content of this methodology is valid for all the System Pillar tasks and domains. [SPPR-11718]


1.3 Document Context

Context

This document relates to MBSE methodology applied for System Pillar project. [SPPR-11720]

1.4 Glossary

Glossary

This document primarily outlines overview of methods and processes, while the definitions can be found directly in this document below and additional in  Systems Engineering Management Plan - Annex M2 Viewpoint Guidelines. [SPPR-11719]

Model Based Systems Engineering MBSE

Model-based Systems Engineering (MBSE) is a paradigm that uses formalised representations of systems, known as models, to support and facilitate the performance of Systems Engineering tasks throughout a system's life cycle. These system models, which are generally expressed in a standardised modelling language such as SysML or ARCML (from the ARCADIA method) express key system information in a concise, consistent, correct, and coherent format. MBSE models permit the standardised consolidation and integration of system knowledge across engineering disciplines and subsystems and streamline key systems engineering tasks while also minimising developmental risk. [SPPR-10775]

Purpose of MBSE Methodology

The System Pillar MBSE methodology is a collection of processes, methods and tools to create and maintain a MBSE model for system analysis and architecture, based on the ARCADIA framework.

- It adheres to the practices from EN 50126-1, CSM-RA and ISO 15288.
- It supports the application of product development process in a model-based context.
- It provides inputs to the requirements management.

[SPPR-9985]

No references



2 Roles

Systems Engineer as Modeler

Modeler. Design/validate/check Models [SPPR-10696]

Model Manager

Role in charge of:

- coordinating all contributions to model
- coordinating the updates to the  Systems Engineering Management Plan - 02 MBSE Methodology Handbook
- coordinating the updates the access to the model in <https://capella.rail-research.europa.eu/>
- coordinating the Modelling Functional team meeting (MOD-F). See  MOD-F

[SPPR-11088]

Data Model Integrator

Role in charge of:

- integrate pieces of data model into the unique, reference model used by System Pillar
- review and solve conflict between pieces of the data model
- maintain the JSON file under GIT

[SPPR-11796]

3 System Pillar MBSE Methodology Overview

System Levels of the System Pillar

The system of systems approach is used inside the System Pillar to recursively refine the structure of the architecture down to the level of subsystems.

The following figure shows the decomposition of a system of systems on one consistent example spanning 5 layers of refinement. Level 5 is the actual subsystem layer and is visually integrated into the bottom layer in the following figure to be able to show the relationship to logical components.

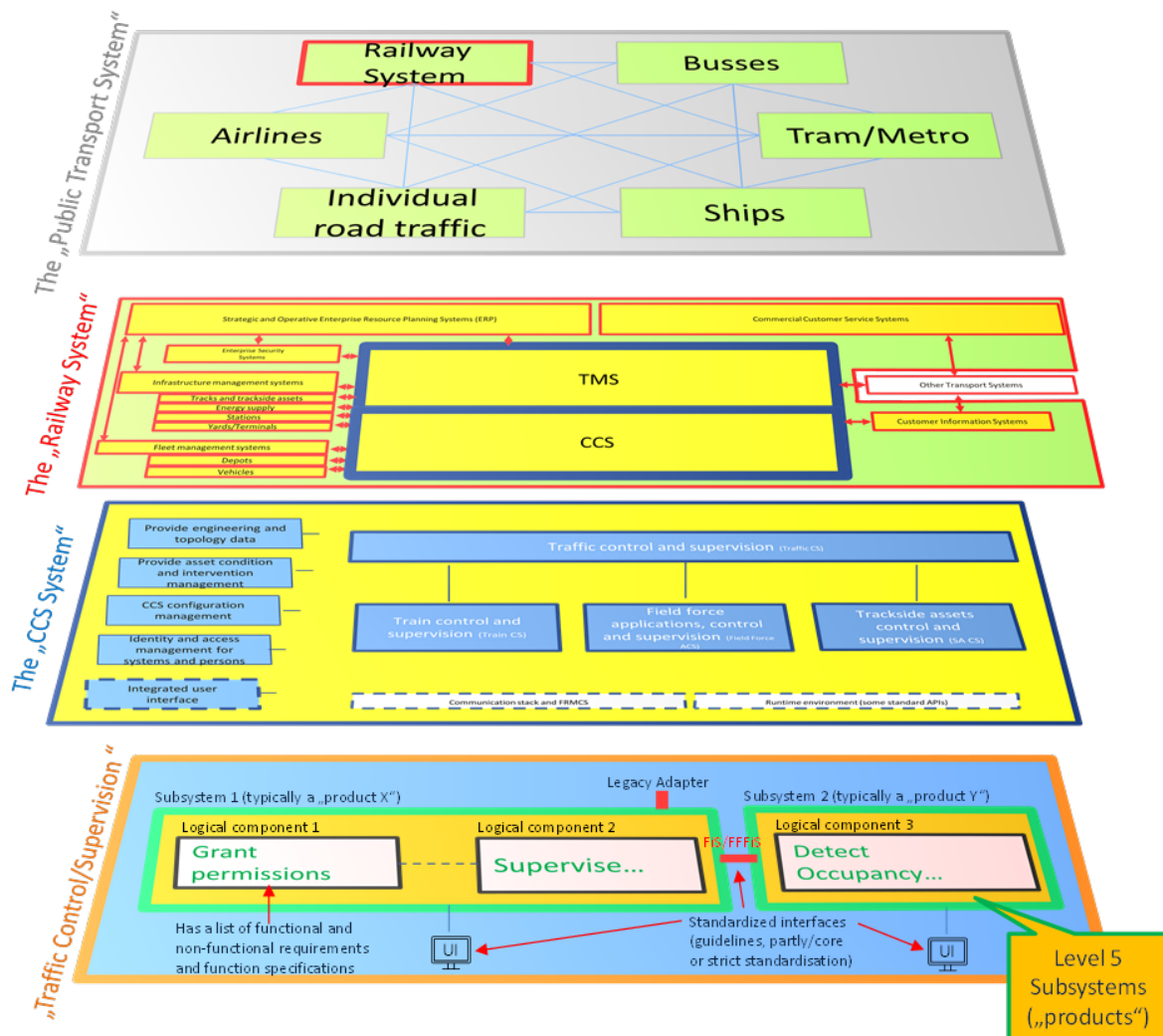


Figure 1: System Level 1-5 combined view

ID	SPPR-2016
----	-----------

Additional explanation to system levels in System Pillar

On the topmost level of scope, the railway system represents the system of interest for the entire System Pillar. It represents the root node in the hierarchy of decomposition layers for systems, that eventually will lead to the identification of the building blocks of the future railway system in Europe. According to the above terminology, the entity under consideration is represented as a system, surrounded by actors in the black box view. These actors represent entities outside the scope of the railway system, that interact with it, either by requiring services from it (e.g. the actor "Passenger" requires to be transported by the "Railway system"), or by imposing restrictions to it (e.g. the actor "Environment" can impose restrictions to the System "Railway System" by weather conditions or obstacles on the line). Actors are connected to the system via interfaces.

The white box view on the same level of abstraction reveals that the "Railway System" itself is a system of systems, where "System CCS/TMS/CMS" is one system under many. The connections to the actors are relayed to the constituent system inside the system of systems, that interacts with the actor on behalf of the entire system of systems. [SPPR-2606]

System

Arrangement of system elements, that together exhibit a stated behaviour or meaning that the individual constituents do not.

According to ISO 15288 a system is “a combination of interacting elements organised to achieve one or more stated purposes. “. In terms of this document, a system in black box view is furthermore defined by:

1. interfaces to actors outside the system, defining the system boundary
2. at least one function allocated to it

A system in white box view can be further refined into (exclusive or):

1. into a more granular systems, hence making itself to a system of systems
2. into subsystems on the lowest level of system of systems refinement

In both cases, a system is a conceptual entity that aggregates the properties of its constituents but is not the element that defines the properties itself. A system is hence subject to the emerging properties of its constituents.

Usage context definitions of term „system“:

- Constituent system: according to ISO 21839, a system that forms part of a system of systems
- System of interest: according to ISO 21839, a system whose life cycle or properties are under consideration in a given context

ID	SPPR-2578
----	-----------

System Of Systems

According to ISO 21839, a system of systems is set of systems that interact to provide a unique capability that none of the constituent systems can accomplish on its own.

A system of systems in our understanding at least comprises two constituent systems.

A system of systems itself can be nested as a constituent system in a larger system of systems, i.e. system of systems can span multiple levels recursively.

ID	SPPR-2593
Abbreviation	SoS

Subsystem

Subsystems are along systems with standard interfaces on System Level 5. Not to be confused with subsystems in the TSI / interoperability directive.

A subsystem is a part of a system, which is not refined any further during the specification task (The term subsystem is used in this document following the referenced architecting standards, it does not correspond to the subsystem definition as in the Interoperability Directive.). It represents a leaf element in the hierarchy of systems-of-systems. A subsystem is defined by the following characteristics:

1. For each subsystem exists a set of specification documents, that allows a supplier to build that subsystem, ideally without the need for further documentation.
2. The level of strictness of the specification can be variable:
3. Interoperable specification: Strict standardisation of all interface aspects that are needed for to systems to fulfil a defined set of operational capabilities together on runtime.
4. Interchangeable specification: Standardisation of all interface aspects that are needed to exchange one of the systems with the lowest reasonable integration effort.
5. Core standardisation specification: An interoperable standardisation that defines a guaranteed minimum of interoperability, but allows and gives room for specific and perhaps incompatible extensions that are only used, when all involved systems have them.
6. Guideline: A recommended specification that can be used as a whole, or can be used partly or changed.
7. A subsystem can be implemented as software only, as hardware only, as a mixture of both depending on the strictness of the specification (the specification can leave that aspect open).
8. Each subsystem can be individually tendered to a supplier
9. Each subsystem can be built individually by a supplier

10. Each subsystem must be integrated into a system, which includes all necessary test, verification, certification and validation activities.

ID	SPPR-2605
----	-----------

MBSE method

The MBSE method chosen for the System Pillar project is based in the ARCADIA method. [SPPR-11721]

(Arc)hitecture (A)nalysis and (D)esign (I)ntegrated (A)pproach

Architecture Analysis & Design Integrated Approach (ARCADIA) is a **system** and **software** architecture engineering method based on architecture-centric and **model-driven engineering** activities. A short description of the method can be found [here](https://www.eclipse.org/capella/arcadia-reference.html). Main resources about ARCADIA can be found on the official website: <https://www.eclipse.org/capella/arcadia-reference.html>

ARCADIA uses four layers to develop and model a system:

- Operational Analysis - what the users of the system need to accomplish
- System Need Analysis - what the system has to accomplish for the user
- Logical Architecture - How the system will work to fulfil expectations
- Physical Architecture - How the system will be developed and built

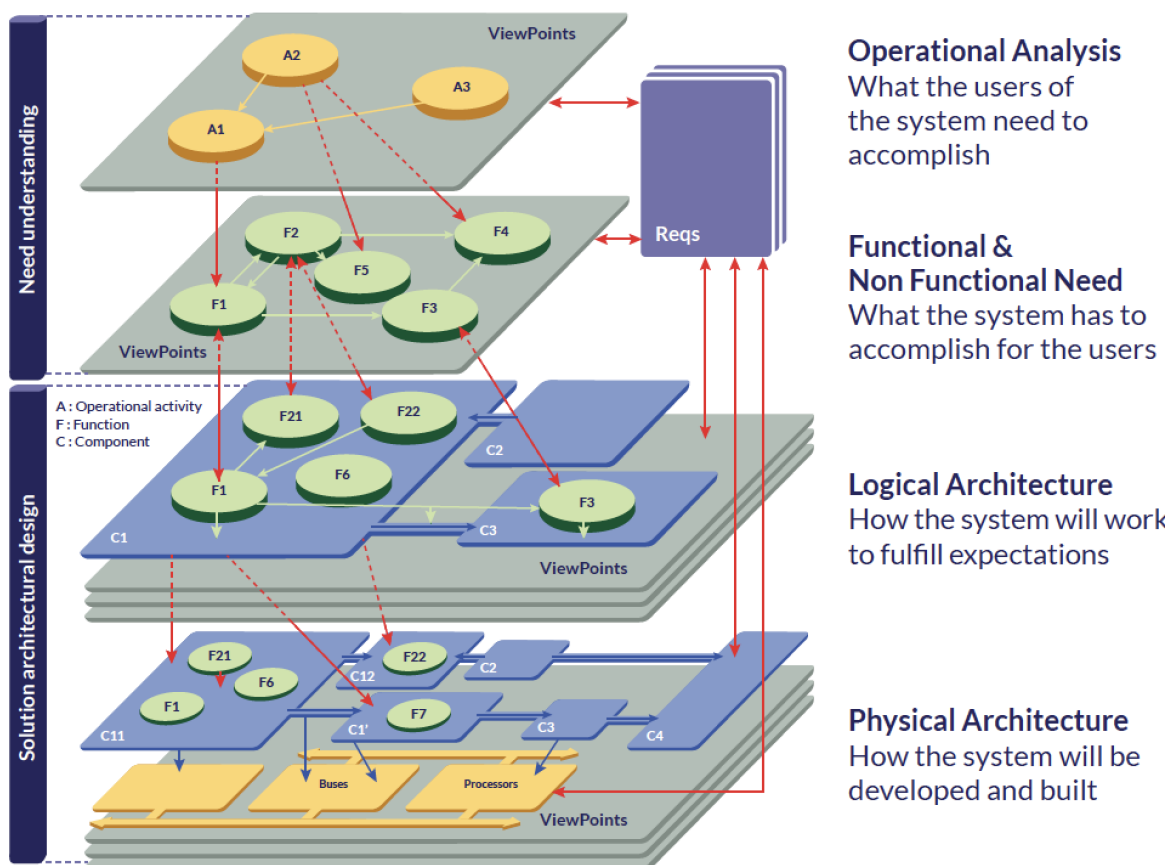


Figure 2 Figure Four perspectives of Arcadia [Arcadia web : <https://www.eclipse.org/capella/arcadia.html>]

ID	SPPR-2290
Abbreviation	ARCADIA

Further guidance to architecture development phases based on ARCADIA

ARCADIA is a model-based Systems Engineering (MBSE) method for system and architecture development that is supported by the open-source modelling tool Capella.

While the Operational Analysis is performed in a system-agnostic way by describing interactions of actors outside of the system of interest, the System Analysis identifies the needs of the users which shall be fulfilled by the system of interest.

Example:

One of the basic things a Railway Undertaking may want to accomplish is to move a train with passengers from a city to another city every morning. The Operational Analysis focuses in trying to describe this need in terms of actions performed by different entities or actors, which interact between them until the objective is reached. This need may be decomposed in simpler chunks of behaviour called Operational Capabilities (e.g., "Move a train from A to B") but, in any case, the system should not be included in this description. Rather, possible actors that may be included could be *railway undertaking*, *signalling entity*. The Operational Capability could include scenarios like "The *signalling entity* commands the *infrastructure elements* from A to B to be in the correct position, check that the *track* is empty, and issue a permission to the *railway undertaking* to move the *train* located in A. The *train* moves until it reaches point B, stops and informs the *signalling entity*."

On the other hand, the System Analysis takes the results of the Operational Analysis and identifies what the system could do to carry out the operational capabilities. It doesn't need to describe the technical solutions that could be used, just what's to be done by the system. In this example, it could be decided that several different System Capabilities could be used to achieve the movement of a train from A to B, one of them being "Set point position." Instead of making the signaller go and move the point with a lever, we can assign functions such as "Control point position", "Observe position of one point", etc. to the CCS System. In a similar fashion, we could decide that is the CCS System who issues the permission to the driver to move the train, or even bypass the driver altogether and move the train automatically. This could lead to assign more functions to the CCS System with inputs from, and outputs to, other actors, with the possibility to use some or all of them depending on the desired level of automation.

An example provided by ARCADIA data sheet is mentioned here https://mbse-capella.org/resources/Datasheet_Arcadia.pdf. [SPPR-7486]

Architectural Concept

The architectural process comprises four steps, each dealing with a separate concern. These steps are described in detail in the following chapters.

1. **Operational analysis (OA):** identify the operational process needs that are to be supported by systems or organisations. This analysis should focus as purely as possible on the processes and ideally does not take any specific technical system architecture into account. The operational analysis is usually performed on an abstraction layer above the topmost system in the systems of systems hierarchy and performed only once.
2. **System analysis (SA):** identify the needs of the system of interest. This step does not design a specific technical solution but captures the needs for the future system. It hence represents a statement of work and not a finished piece of engineering. It is used to rationalize the decision, which operational processes will be performed by the system of interest, and which will be not (these processes then mostly will be either performed by other systems or by human actors and defined as operating rules). System analysis is performed recursively:
 - a. Once for the topmost system of systems, deriving the initial need from the operational analysis
 - b. Multiple times for each system of system decomposition step, deriving the system needs of the lower level of decomposition from the higher level of decomposition
3. **Logical architecture (LA):** design a solution to the system needs based on solution concepts and architectural concepts. Split the system functions based on solution concepts (e.g. absolute positioning vs reference point based localisation, moving blocks, fixed blocks or hybrid) so that it becomes clear, how and by which steps the inputs to a system function are converted to the outputs. This step does not yet define an architecture and does not refer to technical solution concepts like ETCS or ATO. As the system under consideration is still a blackbox, the logical

architecture still leaves the question open, what subsystem structure is the to be used (e.g. very modular subsystems vs. bigger subsystems or combined HW/SW subsystems vs. SW-modules on a common platform). This step is performed once, before the subsystem architecture shall be derived.

4. **Physical architecture (PA):** design the final set of tenderable subsystems and integrate all necessary non-functional requirements. This step integrates all considerations on the intended structure of subsystems and interfaces (down to FFFIS) as well as all open technical aspects into a consistent architectural definition.

[SPPR-2564]

3.1 Mapping of terminology to standards and norms

Terminology mapping

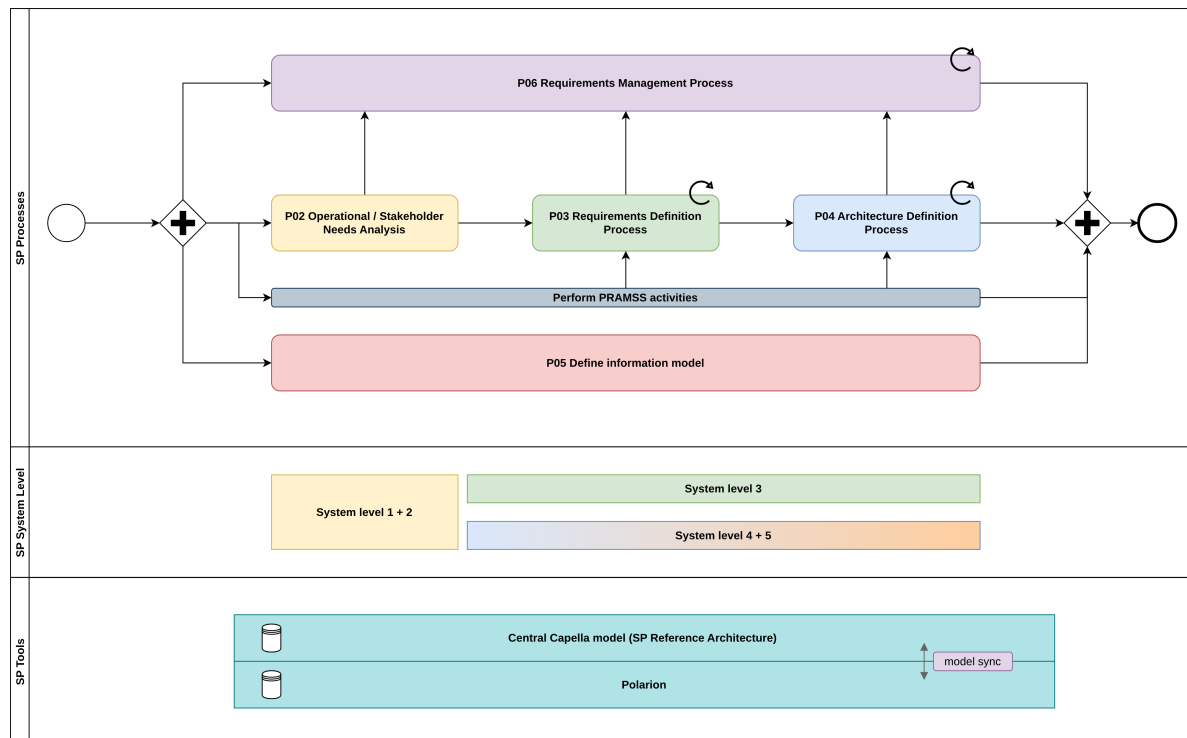
The defined terminology can be mapped to different standard and norms applicable in the railway or engineering domain. Several frameworks (ISO15288, ISO 40020, NATO architecting framework, TOGAF, OPC, INCOSE, etc.) use the terms in a different way. [SPPR-2604]

4 Processes

Purpose and scope of the architectural design process

The main purpose of the System Pillar process is to provide a shared and agreed set of requirements and an architecture that meets stakeholder needs for an integrated European rail system. This is achieved by defining the problem and designing architectural solutions. [SPPR-4082]

System Pillar process overview



[SPPR-4083]


4.1 Requirements Management Process

Requirements management

All requirements management related activities are described in the *SPPROCESS/10 SEMP V 01_01/ Requirements Management Plan : 722568*. [SPPR-11723]

4.2 Requirements Definition Process

Requirements definition process



This process is firstly applicable on System Analysis layer in Arcadia Capella concept. It can be used also recursively at lower levels. Refer to  Systems Engineering Management Plan - Annex 03 Requirements Definition Process. [SPPR-11729]

4.3 Operational / Stakeholder Needs Analysis Process

Operational/Stakeholder Needs Analysis

The Stakeholder Needs and Requirements Process is applied during CENELEC Phase 1 and defines the life cycle concepts and stakeholder requirements, if not already provided as input to this process. The process scope aligns with the Operational Analysis layer of the Arcadia method.

- Identify Stakeholders
- Define the environment and operational environment
- Identify stakeholder needs as capabilities and scenarios / processes
- Analyse interactions
- Analyse and record needs
- Define stakeholder requirements

In the System Pillar the steps are performed by OD domain in order to derive needs for the rulebook based on  Ontology and work item definitions for rulebook and  Rulebook content and documents structure. The rulebook needs are then further used as input for the system requirements. [SPPR-11722]

4.4 Architecture Definition Process

Architecture Definition Process

Abstraction levels are separated in the Capella model between the Logical Architecture and Physical Architecture. In both cases please follow the *SPPROCESS/SEMP Annex D Processes/SEMP process group 04-Logical Architecture : 722568*. [SPPR-11731]


4.4.1 Trade-off analysis guideline

Architecture trade-off analysis

The guideline for trade-off analysis is defined in *SPPROCESS/SEMP Annexes/SEMP Process XX - Trade-off Analysis : 722568* [SPPR-11730]

4.5 Information Model definition process







P5 - Define information model process

The information model definition process is done several times periodically during engineering work (See figure  SPPR-4083 - System Pillar process overview). The creation tasks are identical for each modelling level and allow to enrich step by step the deliverables provided by this. It is a transversal and iterative activity.

High level steps:


- Define terms (For Glossary please see *SPPROCESS/30 SP Metadata Management/Glossary Usage Guidelines : 722568*)
- Define first concepts for data classes and their relationships
- Define properties to refine the data classes

Currently, a test case is running in order to validate the process feasibility. This process will be refined in the future together with the Transversal domain.


-  SPPR-11371 - Draft the data model in Capella
-  SPPR-11327 - Define the data model under JSON format
-  SPPR-11328 - Transfer the JSON data model into Capella
-  SPPR-11329 - Verify the Capella data model
-  SPPR-11334 - Share the Capella data model in Polarion
-  SPPR-11336 - Verify & Validate the data model

This process is intended to guide System Pillar Tasks/Domains in defining a consistent data model for the (sub-)system and interfaces they're in charge of modeling and specifying.

This process shall be applied in parallel to the architectural design activities, as explained in

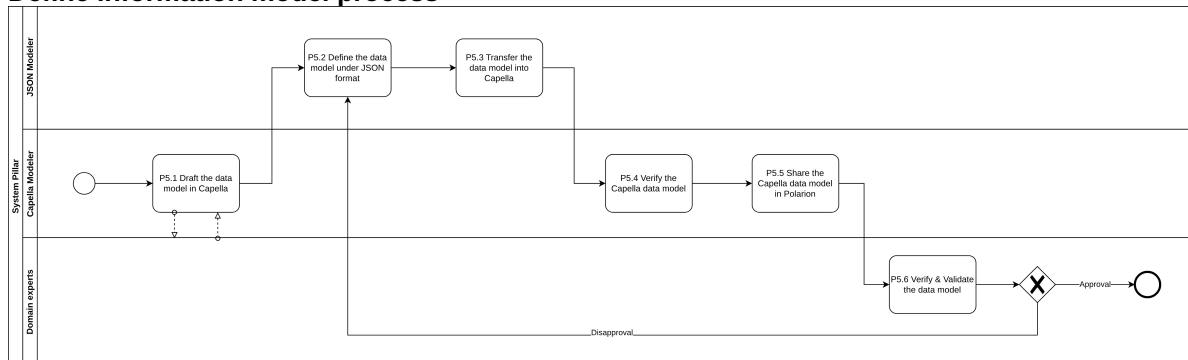
 SPPR-4083 - System Pillar process overview.

Notes:

- Transversal CCS domain's guidelines are applicable to this process. Please refer to  TCCS - Data Model_00_Guide.


ID	SPPR-6083
Inputs	Information model need
Outputs	Information model

Define information model process



[SPPR-11225]

P5.1 - Draft the data model in Capella

The Capella Modeller, together with domain experts, are responsible of defining the preliminary data model. This task consists of drafting the data model under Capella using Information model viewpoint (See  SPPR-8868 - Information model viewpoint) and based on discussions in experts meeting or dedicated "brainstorming" meeting.

ID	SPPR-11371
Inputs	- Capella system model
Outputs	- Capella data model

P5.2 - Define the data model under JSON format

The JSON Modeller is responsible of the modelling activity. This task consists of scripting the data model under JSON format using the selected tool (currently Notepad++) and the inputs from the previous task.

- The JSON Modeller imports the Data model drafted in Capella into JSON modelling tool (Using Capella2JSON tool)
- The JSON Modeller can rely on the ERA Ontology to understand the different railway terms used.

- The JSON Modeller integrates the defined JSON model into the overall CCS-TMS information model. The latter should be already created by Transversal domain and stored in ERJU Git (under JSON format). The data models consistency is ensured by Transversal domain and explained in [TCCS - Data Model_01_Approach](#).

Note :

- In this step, a limitation regarding the association link is identified. All association links will not be correctly synchronised when exporting the Capella draft to JSON format. This limitation is controlled and doesn't present an impact on the final data model.

ID	SPPR-11327
Inputs	- Capella data model - JSON modelling tool - ERA Ontology
Outputs	- JSON data model

P5.3 - Transfer the JSON data model into Capella

The Modeler is responsible of the data model transfer task. This task consists in transferring the JSON data model into Capella.

The JSON data model should be transformed into Capella Compatible format model. This task could be automated via the existing SP Toolchain (Capella Collaboration Manager via Jupiter2Capella bridge). The SP engineering tools are described in [Systems Engineering Management Plan - 03 Engineering Tools](#).

ID	SPPR-11328
Inputs	- JSON data model
Outputs	- Capella data model

P5.4 - Verify the Capella data model

The Capella Modeler is responsible of the verification of the generated Capella data model. This task consists in analysing the Capella data model elements. The [SPPR-11796 - Data Model Integrator](#) could be also involved in this task.

Model Verification

- The Modeller should check if the generated model elements are complete and consistent with the JSON data model elements. A mapping analysis could be conducted to ensure consistency.
- He should also ensure that the predefined model structure in [SPPR-11371 - Draft the data model in Capella](#) has been respected.

Data model Property Values



- The Capella Modeller should at this step add the defined meta information as Property values directly in Capella (Ex. Eulynx SDI-P meta information : Data type, Trigger, Sampling, etc).

ID	SPPR-11329
Inputs	- Capella data model
Outputs	- Verified and enriched Capella data model

P5.5 - Share the Capella data model in Polarion

The Capella Modeler is responsible of the Polarion document generation. This task consists in exporting the Capella data model from the SPRA Capella model into a Polarion document.


The document generation could be completely automated using the C2P bridge. The metamodel information will be included automatically in the properties config of the workitems. Meta information of type "Value" could be considered as non functional requirements on Polarion.

The structure of the Polarion document is defined by SP EET in  Template - System Interface Description. An example of Interface Description document with an integrated data model is provided in  System Interface Description_SDI-P information model_toolchain_demo (SERA Version).


ID	SPPR-11334
Inputs	- Verified Capella data model
Outputs	- Polarion document containing the data model

P5.6 - Verify & Validate the data model

Domain experts are responsible of validating the data model. This task consists of approving the generated polarion document containing the data model.


The V&V is realised based on  Configuration Management Plan - Annex C Document Management Plan, Review and Approval Process


First review step:

- Ensure that the data model is consistent with the CCS-TMS data model and with the modeling methods, structure and policy defined for System Pillar. This task can be conducted by a  SPPR-11796 - Data Model Integrator.

Second review step:

- Check that the information model semantically satisfies the need
- Verify whether the predefined data model structure was respected

If findings conducted to a necessity of modifying the data model (ex. new child data/composed link, change of description, rename, etc), the changes should be realized on the original JSON file (defined in step  SPPR-11327 - Define the data model under JSON format), considered as the source of truth. The process is then reiterated.

If content is under change control the  Configuration Management Plan - Annex B Change Control Management Process needs to be respected.

ID	SPPR-11336
Inputs	- Generated Polarion document of the data model
Outputs	- Validated data model through Polarion document approval OR - Request to modify the JSON data model

5 MBSE Tools

MBSE tool chain

The MBSE Tool chain is described in the *SPPROCESS/10 SEMP V 01_01/System Pillar Engineering Tools : 722568* document. [SPPR-11733]

6 Modelling Methods, Rules and Guidelines

Purpose and scope of modelling rules and guidelines

The purpose of the modelling guidelines and rules is to provide a consistent modelling approach inside an

MBSE model used for requirements engineering and architectural analysis / design for system levels 1-5.


- *SPPROCESS/10 SEMP V 01_01/SEMP Annex M ARCADIA Capella Modelling Rules : 722568* provides the rules for Capella model elements. They contain element and naming convention. It includes existing rules from previous or current initiatives and projects as inputs.
- Viewpoint specific methods and explanations are provided in *SPPROCESS/10 SEMP V 01_01/SEMP Annex M2 Capella diagram rules : 722568* including detailed instructions of how to structure, maintain and use content of the model.

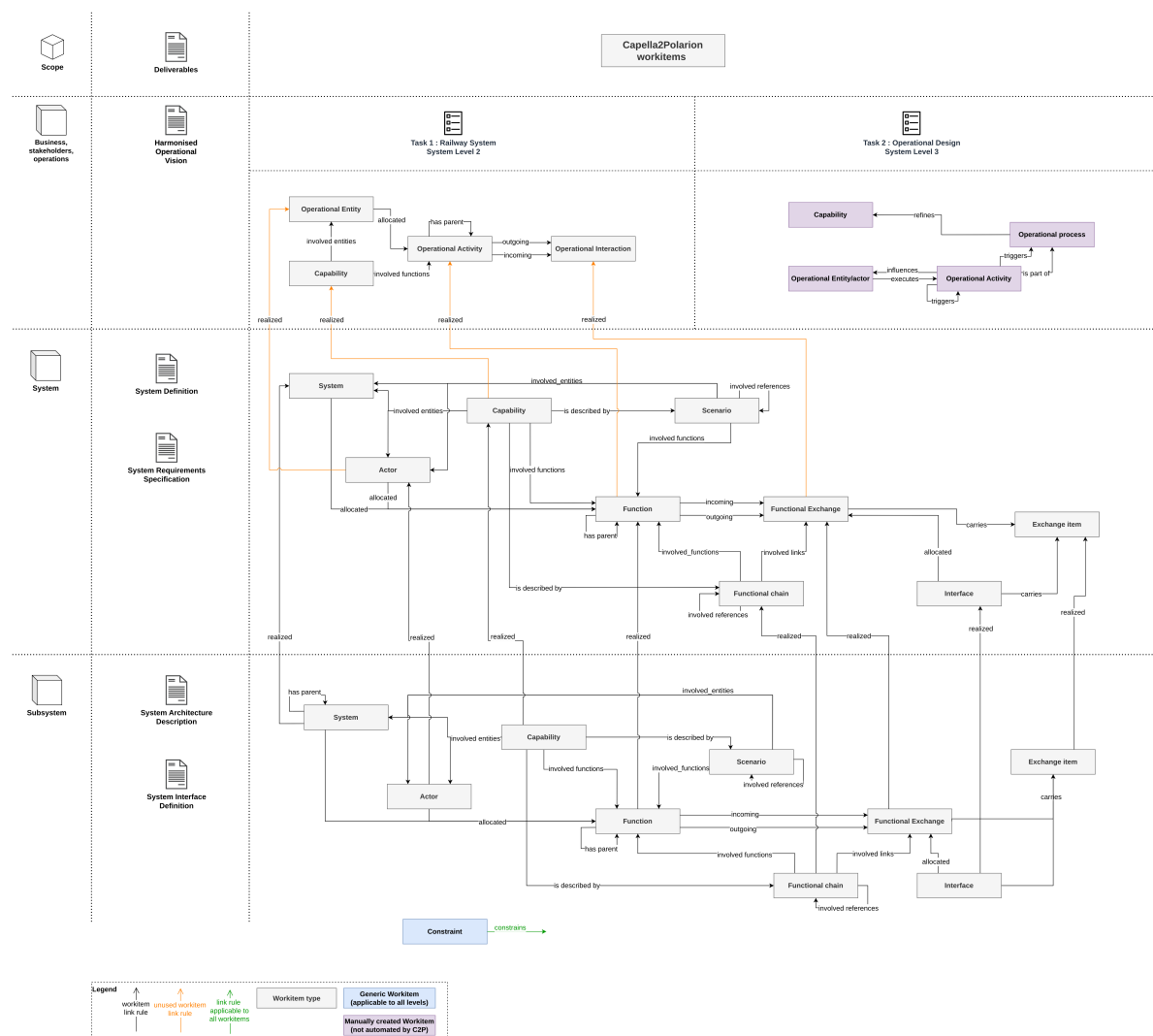
[SPPR-4504]

7 Linking Rules for MBSE - Capella to Polarion work items

System Pillar traceability map - MBSE work items

Note :







- Each workitem link rule which is automated by Capella2Polarion bridge in the context of System Pillar has a naming prefix "C2P-" (Example : C2P-involved_functions).
- Workitems' relationships with Requirements are described in the  Requirements Management Plan.



























[SPPR-11164]

7.1 Common linking rules



A constraint constraints a function, an exchange, an entity or an interface

Source Workitem	{c} Constraint
Link Role	constrains
Target Workitem	 C2P-Function,  C2P-Functional Exchange,  C2P-System,  C2P-Actor,  C2P-Interface,  C2P-Exchange Item
Mandatory type	optional link
ID	SPPR-11096

Each model element can be referenced in description



Source Workitem	 C2P-Operational Activity,  C2P-Operational Entity,  C2P-Operational Interaction,  C2P-Capability,  C2P-Actor,  C2P-System,  C2P-Function,  C2P-Functional Exchange,  C2P-Functional Chain,  C2P-Interface,  C2P-Exchange Item,  C2P-Scenario
Link Role	references in description
Target Workitem	 C2P-Operational Activity,  C2P-Operational Entity,  C2P-Operational Interaction,  C2P-Capability,  C2P-Actor,  C2P-System,  C2P-Function,  C2P-Functional Exchange,  C2P-Functional Chain,  C2P-Interface,  C2P-Exchange Item,  C2P-Scenario
Mandatory type	automated by C2P
ID	SPPR-11100

An Exchange Item realized another Exchange Item from the upper level



Source Workitem	 C2P-Exchange Item
Link Role	C2P-realized
Target Workitem	 C2P-Exchange Item
Mandatory type	automated by C2P
ID	SPPR-11361

7.2 Operational Analysis linking rules



An operational interaction has as a target an operational activity

Source Workitem	 C2P-Operational Activity
Link Role	C2P-incoming
Target Workitem	 C2P-Operational Interaction
Mandatory type	automated by C2P
ID	SPPR-11091

An operational interaction has as a source an operational activity

Source Workitem	 C2P-Operational Activity
Link Role	C2P-outgoing
Target Workitem	 C2P-Operational Interaction
Mandatory type	automated by C2P
ID	SPPR-11092

An operational activity can have a sub operational activity



Source Workitem	 C2P-Operational Activity
Link Role	C2P-has parent
Target Workitem	 C2P-Operational Activity
Mandatory type	automated by C2P
ID	SPPR-11093

7.3 System Definition and Architecture linking rules




A subsystem is part of a system

Source Workitem	 C2P-System
Link Role	C2P-has parent
Target Workitem	 C2P-System
Mandatory type	automated by C2P
ID	SPPR-11097






A system realises a system from the upper architecture layer

Source Workitem	 C2P-System
Link Role	C2P-realized
Target Workitem	 C2P-System
Mandatory type	automated by C2P
ID	SPPR-11133



An actor realized an actor or an operational entity from the upper level

Source Workitem	 C2P-Actor
Link Role	C2P-realized
Target Workitem	 C2P-Actor,  C2P-Operational Entity
Mandatory type	automated by C2P
ID	SPPR-2440




An entity has allocated functions or operational activities

Source Workitem	 C2P-System,  C2P-Actor,  C2P-Operational Entity
Link Role	C2P-allocated
Target Workitem	 C2P-Function,  C2P-Operational Activity
Mandatory type	automated by C2P
ID	SPPR-2436






A Capability realized another Capability from the upper level

Source Workitem	 C2P-Capability
Link Role	C2P-realized
Target Workitem	 C2P-Capability
Mandatory type	automated by C2P
ID	SPPR-7658






A capability is described by a scenario or functional chain

Source Workitem	 C2P-Capability
Link Role	C2P-is described by
Target Workitem	 C2P-Scenario,  C2P-Functional Chain
Mandatory type	automated by C2P
ID	SPPR-11116


A scenario, a functional chain and a capability have involved functions or operational activities

Source Workitem	 C2P-Scenario,  C2P-Functional Chain,  C2P-Capability
Link Role	C2P-involved_functions
Target Workitem	 C2P-Function,  C2P-Operational Activity
Mandatory type	automated by C2P
ID	SPPR-11118




A scenario and a capability have involved entities

Source Workitem	 C2P-Scenario,  C2P-Capability
Link Role	C2P-involved_entities
Target Workitem	 C2P-System,  C2P-Actor,  C2P-Operational Entity
Mandatory type	automated by C2P
ID	SPPR-11323

Function can be split in the sub function

Source Workitem	 C2P-Function
Link Role	C2P-has parent
Target Workitem	 C2P-Function
Mandatory type	automated by C2P
ID	SPPR-2448

A function realized another function or operational activity from the upper level

Source Workitem	 C2P-Function
Link Role	C2P-realized
Target Workitem	 C2P-Function,  C2P-Operational Activity
Mandatory type	automated by C2P
ID	SPPR-2441


A Functional Exchange has as a target a Function

Source Workitem	 C2P-Function
Link Role	C2P-incoming
Target Workitem	 C2P-Functional Exchange
Mandatory type	automated by C2P
ID	SPPR-2394

A Functional Exchange has as a source a Function

Source Workitem	 C2P-Function
Link Role	C2P-outgoing
Target Workitem	 C2P-Functional Exchange
Mandatory type	automated by C2P
ID	SPPR-2393

A functional exchange realized another functional exchange or an operational interaction from the upper level





Source Workitem	 C2P-Functional Exchange
Link Role	C2P-realized
Target Workitem	 C2P-Functional Exchange,  C2P-Operational Interaction
Mandatory type	automated by C2P
ID	SPPR-2442

A functional chain has involved links with functional exchanges

Operational Scenarios are "time ordered" sequences of operational activities

Source Workitem	 C2P-Functional Chain
Link Role	C2P-involved_links
Target Workitem	 C2P-Functional Exchange
Mandatory type	automated by C2P
ID	SPPR-2281

A Functional Chain or a Scenario can, respectively, involve another Functional Chain or another Scenario.

Source Workitem	 C2P-Functional Chain,  C2P-Scenario
Link Role	C2P-involved_references
Target Workitem	 C2P-Functional Chain,  C2P-Scenario
Mandatory type	automated by C2P
ID	SPPR-2385



A Functional Chain realized another Functional Chain from the upper level

Source Workitem	 C2P-Functional Chain
Link Role	C2P-realized
Target Workitem	 C2P-Functional Chain
Mandatory type	automated by C2P
ID	SPPR-11359




Interface has allocated functional exchange

Source Workitem	 C2P-Interface
Link Role	C2P-allocated
Target Workitem	 C2P-Functional Exchange
Mandatory type	automated by C2P
ID	SPPR-2391

An Interface realized another Interface from the upper level

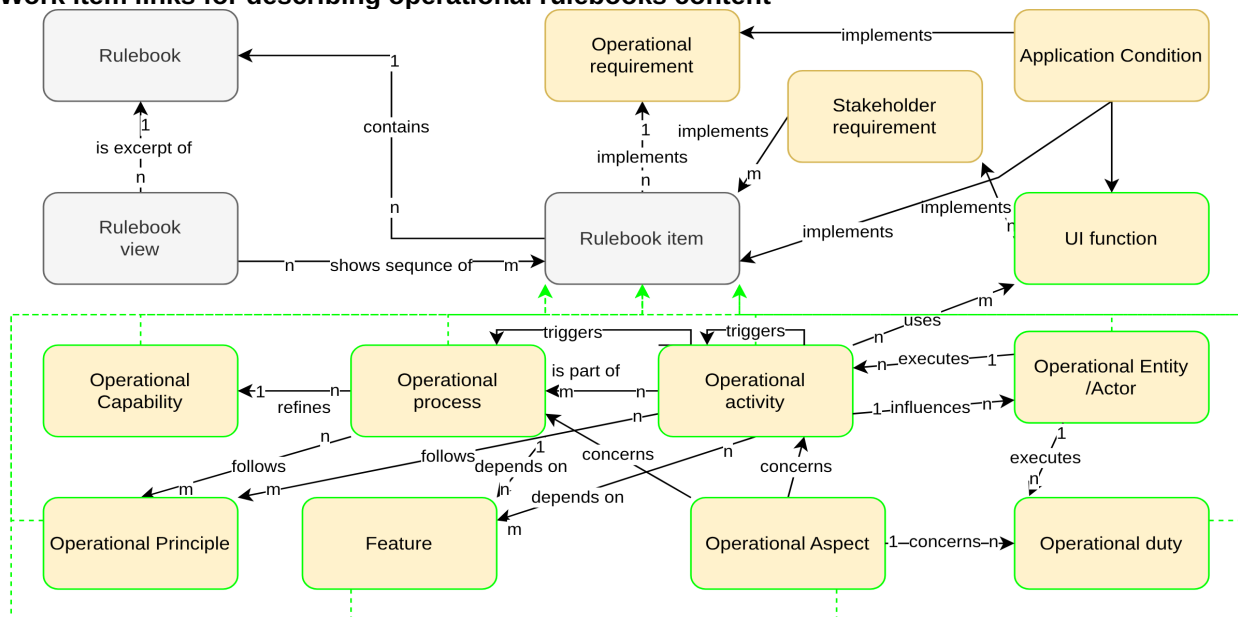
Source Workitem	 C2P-Interface
Link Role	C2P-realized
Target Workitem	 C2P-Interface
Mandatory type	automated by C2P
ID	SPPR-11360

A functional exchange and interface carries an exchange item

Source Workitem	 C2P-Functional Exchange,  C2P-Interface
Link Role	C2P-carries
Target Workitem	 C2P-Exchange Item
Mandatory type	automated by C2P
ID	SPPR-2430

7.4 Operational Design linking rules

Work item links for describing operational rulebooks content



[SPT2OD-8907]



7.4.1 Manually created workitems link rules

An operational process refines an operational capability



Source Workitem	 Operational Process
Link Role	refines
Target Workitem	 Operational Capability

Mandatory type	mandatory link for both
ID	SPPR-11348



A operational activity influences an operational entity/actor

Source Workitem	 Operational Activity
Link Role	influences
Target Workitem	 Operational Entity/Actor
Mandatory type	mandatory link for both
ID	SPPR-11349




An operational entity/actor executes an operational activity

Source Workitem	 Operational Entity/Actor
Link Role	executes
Target Workitem	 Operational Activity
Mandatory type	mandatory link for both
ID	SPPR-11350



An operational activity is part of an operational process

Source Workitem	 Operational Activity
Link Role	is part of
Target Workitem	 Operational Process
Mandatory type	mandatory link for both
ID	SPPR-11351

An operational activity can trigger another operational activity or an operational process

Source Workitem	 Operational Activity
Link Role	triggers
Target Workitem	 Operational Activity,  Operational Process
Mandatory type	optional link
ID	SPPR-11352

An operational entity/actor can be part of another operational entity/actor

Source Workitem	 Operational Entity/Actor
Link Role	has parent
Target Workitem	 Operational Entity/Actor

Mandatory type	optional link
ID	SPPR-11353

8 Appendix

8.1 Standards and References

Systems Architecture Modeling with the Arcadia Method

This book is an illustrative guide for the understanding and implementation of model-based systems and architecture engineering with the Arcadia method, using Capella, a new open-source solution.

More than just another systems modeling tool, Capella is a comprehensive and extensible Eclipse application that has been successfully deployed in a wide variety of industrial contexts. Based on a graphical modeling workbench, it provides systems architects with rich methodological guidance using the Arcadia method and modeling language. Intuitive model editing and advanced viewing capabilities improve modeling quality and productivity, and help engineers focus on the design of the system and its architecture.

This book is the first to help readers discover the richness of the Capella solution.

ID	SPPR-11474
Hyperlinks	external reference - https://shop.elsevier.com/books/systems-architecture-modeling-with-the-arcadia-method/roques/978-1-78548-168-0

8.2 Open actions and issues

Integration of *SPPROCESS/24 WP4 Specification and Model Integration and Quality assurance/ Introduction to ARCADIA* : 722568 into handbook.