ERJU SYSTEM PILLAR

Systems Engineering Management Plan - Annex 03 Requirements Definition Process

Systems Engineering Management Plan - Annex 03 Requirements Definition Process

Author(s)	Dennis Kunz , Jorge Block , EDDOUS Sayfeddine (SNCF RESEAU / Directions Techniques Réseau / DGII DTR GE SF Solutions) , ANTOONS Gilles

Abstract	This document describes all SEMP processes of the System Analysis according to ARCADIA

Config Item	System Engineering Management Plan
Document ID	SEMP Annexes/SEMP process 03 Requirements Definition Process#722555 Systems Engineering Management Plan - Annex 03 Requirements Definition Process
Classification	Public
Status	Released
Version	1.0
Revision	722555
Last Change Date	02.10.2025

Document History

1.1 07.01.2023	Schmidt Steffen (I-NAT-SIBS-ISST)	Reviewed version including Findings from Review X.X
1.2 16.08.2023	ANTOONS Gilles	Reviewed version including Findings from Review 1.1
1.3 25.08.2025	Jorge Block	Reviewed version including Findings from Review 1.2
1.0 19.09.2025	Dennis Kunz	Approved version based on Review 1.3

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

Type of Approval	Document Review
Approvals	EDDOUS Sayfeddine (SNCF RESEAU / Directions Techniques Réseau / DGII DTR GE SF Solutions) : Approved , Jorge Block : Approved , Renard, Marie Pierre (SMO RI MT FR ADC TGMTR3) : Approved , ANTOONS Gilles : Approved
Attachments	Systems_Engineering_Management_Plan_-_Annex_03_Requirements_Definition_Process-comments.xlsx

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) : Waiting , KEFALAS, Georgios : Approved , Schmidt Steffen (I-NAT-GST-ERTM) : Approved , SCHWAN Nico : Approved , Jorge Block : Approved , ANTOONS Gilles : Waiting , Renard, Marie Pierre (SMO RI MT FR ADC TGMTR3) : Approved

Table of contents

1 Preamble

1.1 Purpose

Purpose of the requirements definition process
This requirements definition process is initially executed once during CENELEC phase 2 to create a System Definition and then recursively applied for phases 4 and 5, covering the system and lower-level subsystems. In addition to the system definition the system requirement specification is created.
[SPPR-11707 **]**

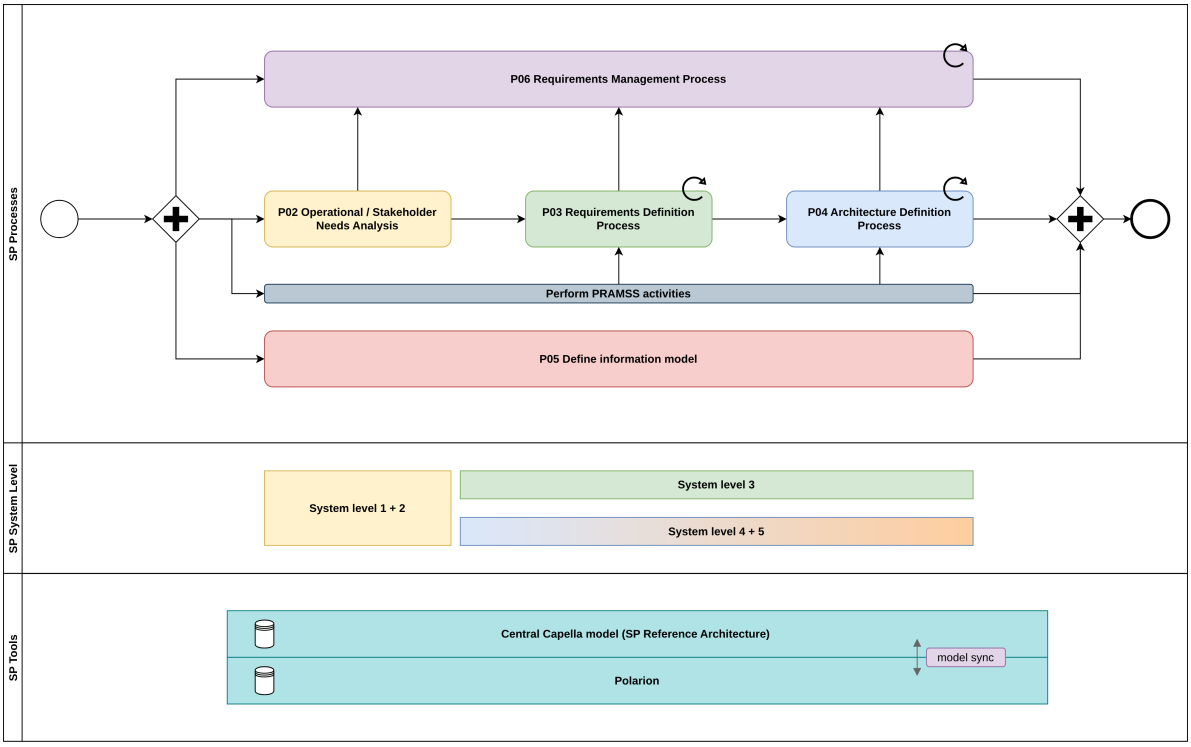
1.2 Intended Audience

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

1.3 Document Context

This document describes the Requirment Process as part of the System Pillar Processes.

System Pillar process overview



[SPPR-4083 **]**

1.4 Glossary

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

2 Content

P03 - Requirement definition process

The requirement definition process consistents of the following tasks:

- SPPR-11475 - P3.0 Prepare for requirements definition
- SPPR-4593 - (Optional) Define system contribution to higher level requirements
- SPPR-5196 - Define functional system boundary
- SPPR-5197 - Define system constraints
- SPPR-5202 - Define required system functions
- SPPR-5209 - Consolidate system definition
- SPPR-5204 - Provide input for risk and hazard analysis
- SPPR-5206 - Define system requirements
- SPPR-11477 - Consolidate system requirements

Inputs	Operational processes and needs Stakeholder requirements
Outputs	System Definition System Requirements Specification
ID	SPPR-2336

Process diagram of "Requirements Definition Process"

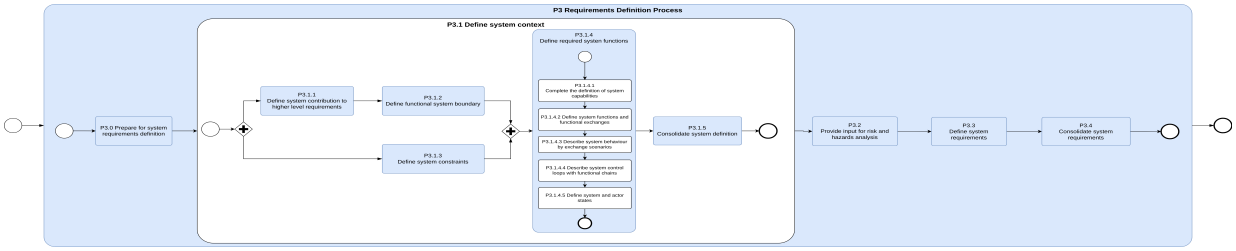


Figure 1 "Requirements Definition Process"

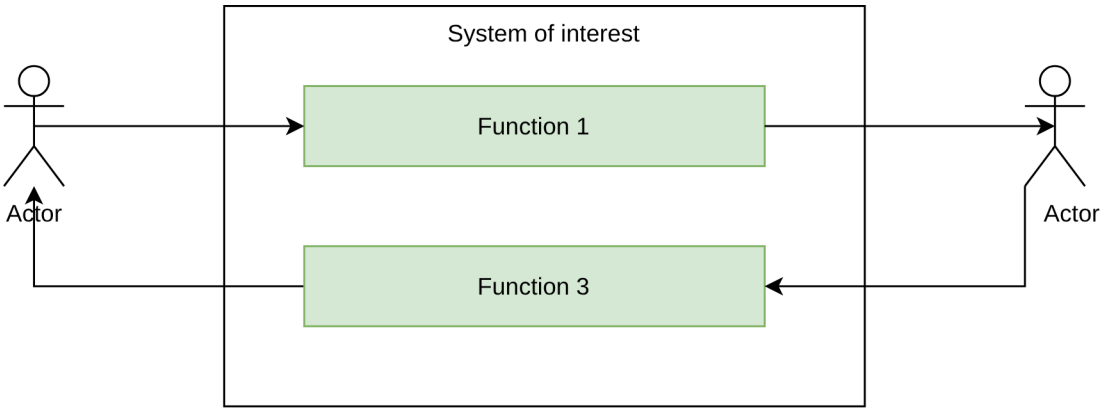
[SPPR-5402]

2.1 P3.0 Prepare for requirements definition

Preparation of the requirements definition
The purpose of the process task is to read and understand the requirement strategy defined in the *SPPROCESS/10 SEMP V 01_01/Requirements Management Plan : 722555* and the Systems Engineering Management Plan - Annex M2 Viewpoint Guidelines to understand the different viewpoints. **[**SPPR-11715 **]**

2.2 P3.1 Define system context

Black-box view of the system under consideration
The analysis is performed with a black-box view of the system. This means no (internal) systems are considered. The result is a set of functions whose inputs and outputs are interconnected to form a network of functions. This network of functions
is derived and allocated to the system and to its actors. By doing so, one defines, which functionality shall be done by the system of interest, and which functions are performed by the actors (which, in turn, implicitly defines the responsibilities and tasks for the actors).



[SPPR-7487 **]**

Solution-agnostic approach
The System Definition should represent the pure business logic needs and should be as generic and as agnostic as possible of technical solution concepts. I.e., functions need to be defined in a way that they can be fulfilled by different technical solution concepts.
Example: Authorsing the motion of a train can be achieved by classical lineside signalling systems + national ATP, or by ETCS L2. The function needs to be defined in a way that both implementations would be a valid solution for that generic function. The refinement into concrete technical solution concepts happens during the Architecture phase.
It must be noted that the SEMP (SPPR-5197 - Define system constraints) allows for the possibility to define system constraints in this phase: constraints on the system solution (which may affect the scope of the system) and/or constraints in the implementation solution (which may affect the technologies, standards, etc. to be used).

That said, it is better to avoid defining specific technical solutions during the System Definition. Doing so allows to devise a description of what the system needs to do that is more stable in time and applicable across a broader range of situations. This, in turn, helps to achieve design solution flexibility and makes it easier to use new technologies without having to revise each time what the system has to provide to its users. Eventually, a specific range of technical solutions must be adopted if we want to develop a practical system, but that can be done at a later stage of the process, during the Architecture pahse.

The System Definition must also describe the information exchanged between the System and the external Actors, be they human actors or other systems. This information exchange is the basis of the (external) system interfaces. In this phase, they should be defined in terms of what information needs to be transmitted (such as time, train speed, point position, etc.). **[**SPPR-7488 **]**

2.2.1 P3.1.1 Define system contribution to higher level requirements

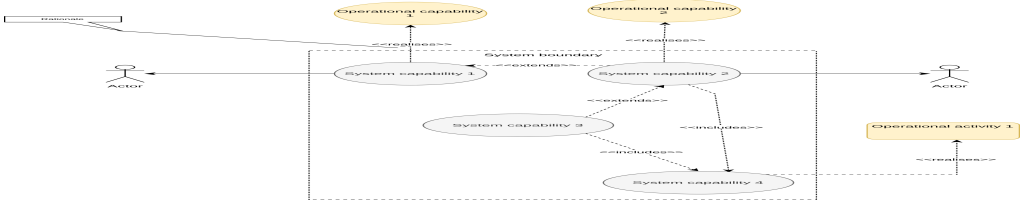
P3.1.1 - (Optional) Define system contribution to higher level requirements
Description:
This task describes the steps to define initial external actors and initial capabilities of the system of interest, based on provided input from higher level requirements (e.g. stakeholder requirements or system requirements of higher system level). In order to refine the understanding of the system scope a set of initial system capabilities are either derived from the higher level requirements or defined based on the existing knowledge of the system. It is important that this task only identifies initial actors and capabilities, which will be refined and justified in the next tasks.

Methods:

- *SPPROCESS/10 SEMP V 01_01/SEMP Annex M1 Capella element rules : 722555*
- *SPPROCESS/10 SEMP V 01_01/SEMP Annex M2 Capella diagram rules : 722555*
- Team **brainstorming** including domain experts and system engineers can be organised to identify and analyse alternatives.

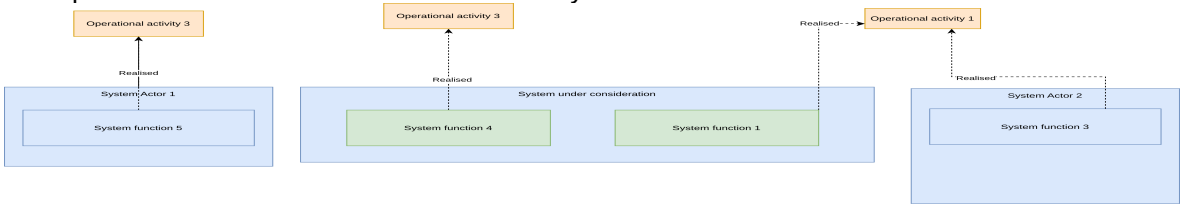
Steps:

- **Define initial system actors**, provided by input from higher level requirements. Determine whether there is a candidate for an external system actor. This is the case where a system actor performs a function which is not to be automated/performed by the system of interest and interacts with the system of interest.
    - *Note: The name and the scope of the system actor might be more specific and its scope need to be defined (e.g. specific technical or human representation).*
- **Define initial system capabilities**, by analysing each higher level requirement. Determine the system capability that yieldes an observable valuable result to one or more system actors.
    - *Note: This includes a first definition of a name, involved actors and pre- and postcondition.*
    - Example: Draw IO can be used to be used for the sketch of system capabilities.



- **Optional: Define initial functions** from higher level requirements**,** by analysing each requirement. Determine whether it is a candidate to be transferred/traced to system analysis as a function. The function captured by the considered requirement is allocated either to the system of interest or to a system actor. In a further case, the function is split into a system function and an actor function.

1. Example: Draw IO to be used for the sketch of system actors and allocation of functions.



Notes:

- This task does not need to result in a complete set of actors, capabilities or functions. For the entire system multiple iterations are recommended.

Inputs	Higher level requirements (i.e. corresponding requirements specification)

Outputs	Record of initial system capabilities, system actors and functions [Polarion draft page to be replaced later by model output]
ID	SPPR-4593

2.2.2 P3.1.2 Define functional system boundary

P3.1.2 - Define functional system boundary
Description:

- Define the boundary and identify functional interfaces in terms of behaviour and properties. Define the system context, (i.e. how the system under consideration fits into the external environment) and system boundary that reflect operational scenarios and expected system behaviour.
- Evaluate the alternative boundaries and select the best boundary most suitable for the problem at hand.
- Ensure the functional interfaces reflect the operational scenarios and expected behaviours
- Consider also other inputs into deciding on the boundary that is often not only influenced by external inputs but also depends on organisational constraints and technical constraints such as technology maturity or strategic documentation.

Methods:

- *SPPROCESS/10 SEMP V 01_01/SEMP Annex M1 Capella element rules : 722555*
- *SPPROCESS/10 SEMP V 01_01/SEMP Annex M2 Capella diagram rules : 722555*
- Systems Engineering Management Plan - Annex T Trade-Off Analysis

Steps:

a. **Define set of system capabilities**, based on the initial system capabilities. Create a consolidated version of the SPPR-6629 - Capabilities definition by following SPPR-10989 - Construction method capability definition.

Note: The system capability definition diagram will also get updates while defining the system actors.

b. **Define a set of system actors** by creating or updating SPPR-6427 - System context description by following SPPR-11002 - Construction method system context description in the model. This is based on a consolidation of the initial system actors, initial capabilities and initial functions (drafted as part of SPPR-4593), also taking system constraints into account.

c. **Define external system interfaces** by updating SPPR-6427 - System context description following SPPR-11002 - Construction method system context description. Furthermore create SPPR-6626 - Interface description following SPPR-10983 - Construction method interface description in the model. This can be non-externally constrained system interface without available external constraints or external constraint interfaces.

d. **Evaluate the available system context** to identify conflicts, i.e. areas where various allocations and hence various system boundaries are possible (also refer the note below). For this purpose the record of potential system capabilities, system actors and functions shall be analysed.

For every system boundary conflict found,

e. **Identify possible system boundary alternatives** involved in the conflict. If it is required, provide additional alternative to the assessment of the conflict, and briefly develop each system-actor interaction and the related functionalities.

- *Note: Some system boundary alternatives may already be available, as part of collaborative projects or given by constraints. Those can be taken as part of the current step.*

f. **Identify the constraints** affecting each of the alternatives, from the constraints currently available.

- *Note: Pay special attention to the constraints that are overlapping or conflicting with each other. Those constraints may be not complied depending on the alternative selected and may require a further discussion with stakeholders in later phases.*

g. If the boundary conflicts found cannot be easily resolved by engineering judgement, **perform a trade-off assessment**, to provide further evidence to the selected boundary alternative.
h. **Define selected system boundary** from the selected boundary conflict solution. Record this in an updated version of SPPR-6427 - System context description and SPPR-6628 - Function allocation by following SPPR-11003 - Construction method function allocation.
i. **Consolidate system constraints** that are relevant for the selected alternative. Keep track of the disregarded constraints that may be not longer applicable.

- *Note: Constraints should only be discarded after the corresponding discussion with involved stakeholders, in order to avoid rejected validations in later phases.*

j. **Create the system boundary trade-off record** with a view according to the System boundary trade-off record template to keep track of the reasoning and relevant information of the decision.

After each actor is defined and consolidated:

k. Align the system context with the collaborative projects, by analysing and comparing the provided content from the collaborative project with the current system context.
l. If conflicts are found with collaborative projects, analyse the conflict between the system context and the relevant content from the collaborative project in order to select the most suitable solution. Update the diagrams accordingly.

- *Note: If based on the outcome of the analysis the decision is NOT adapted to the collaborative project, this decision should be properly discussed with involved stakeholders from the collaborative projects.*
- *Note: If the boundary conflicts found cannot be easily resolved by engineering judgement, remember that a trade-off assessment can be performed.*

Notes:

- *A conflict on the system boundary is defined as a functional allocation that could be either not yet fixed, is unclear or has more than one allocation alternative. Therefore, it is not obvious to decide on the system boundary. Possible naming or terminology mismatches found with the collaborative projects are not part of this definition and should be independently discussed with the collaborative projects.*
- *This task does not need to result in a complete set of actors, capabilities and functions for the entire system in one single try. The actors, capabilities and functions will be completed by further iterations of this task working on the system's contribution to other stakeholder requirements.*

Inputs	- Record of initial system capabilities, actors and functions are defined - System constraints record
Outputs	- SPPR-6427 - System context description - SPPR-6629 - Capabilities definition - SPPR-6628 - Function allocation - SPPR-6626 - Interface description - System boundary trade-off record [Polarion document]
ID	SPPR-5196

2.2.3 P3.1.3 Define system constraints

P3.1.3 - Define system constraints
Description:
This task describes the steps to define the system constraints and documented in a system constraints record. These constraints are further decomposed as necessary, allocated to the system level and implementation proposals are given.
The system constraints will be used as selection criteria when a decision between alternative system boundaries is taken.
Categorise inputs to derive constraints on the system. Each input shall be analysed with respect to the categories below to answer the question **what shall be within the scope of the system** or **how the system shall be implemented**.

Steps:
a. **Identification of constraints** from inputs.

- Consider constraints being flown down from the Stakeholder Requirements, Lifecycle Concept Documents e.g. CONEMP or organisational limitations that impose unavoidable constraints.
- Consider constraints that may come from external, such as interfacing systems in the operating environment, enabling systems or regulatory requirements. This includes existing infrastructure, legacy systems and their interfaces.
- Consider constraints resulting from users, operators, maintainers and disposers.

As inspiration consider the following guiding questions to help identify constraints.

- Does the input have an impact on the scope (e.g. boundary) of the system?
- Does the input have an impact on the implementation of technical solutions to a potential subsystem?
- Does the input express an externally imposed restriction?
- Is the input considered non-negotiable by multiple experts of the subject matter?

b. **Assess constraints** with respect to their applicability to the system development: For each identified constraint, check if the constraint is directly applicable to the development of the system or its subsystems, if further decomposition is needed, or if the constraint might not be applicable at all.
c. Describe constraints in work items including rationale and according to defined rules.
d. **Define implementation proposal for constraints**: For each constraint, define the attribute Implementation Proposal in modelling-tool-specific terms; i.e. specify which model elements could be used to realise the constraint. Populate this information.
e. **Rationalise the result of the analysis**
f. **Validate constraints with stakeholders** to be certain they are fully understood and correct
g. This steps is performed in iterations and in parallel with the architecture definition.

Inputs	- Stakeholder requirements - Input material with potentially applicable constraints
Outputs	- System constraints record
ID	SPPR-5197

2.2.4 P3.1.4 Define required system functions

P3.1.4 - Define required system functions
Description:
The goal of this task is to identify and defined expected function allocated to the system of interest or required functions from the system actors including the associated performance and quality characteristics. This should be done by analysing the system capabilities that describe the system's behaviour under various conditions as it responds to a request from one or more actors.

Methods:

- *SPPROCESS/10 SEMP V 01_01/SEMP Annex M1 Capella element rules : 722555*
- *SPPROCESS/10 SEMP V 01_01/SEMP Annex M2 Capella diagram rules : 722555*

The process tasks steps are described in:

- SPPR-10605 - Refine definition of system capability
- SPPR-10607 - Define required functions and functional exchanges
- SPPR-10606 - Describe behaviour with exchange scenarios
- SPPR-11458 - (Optional) Describe control loops with functional chains
- SPPR-10604 - (Optional) Define system and actor states

Inputs	- Capabilities definition - System context description
Outputs	- Capability context - Function involvement - Functional chain description - Exchange scenario - States definition
ID	SPPR-5202

P3.1.4.1 - Refine definition of system capability
Steps:
This step describes the system capability in detail by defining the relationships and the pre- and postconditions in detail:

1. Define precise **pre- and postconditions of the system capability** of interest.
2. Check and **complete the involved actors of the system capability** in SPPR-6630 - System capability context by following SPPR-10989 - Construction method capability definition
3. Check and complete the **relationships with other system capabilities**.

Notes:
- Completely defined system capabilities of interest (System Capability Context) are further analysed by means of system functions (including functional exchanges and chains), system exchange scenarios and system states.
- Dependencies between the system capability of interest and other system capabilities may help the elaboration of pre- and postconditions for the system capability of interest. This elaboration can be carried out by performing iteratively the steps in this task, as further information on the system capability of interest and its related system capabilities will allow a better construction of pre- and postconditions.
- If there is a relationship with another system capability missing from the diagram, it can be manually added.

Inputs	- SPPR-6629 - Capabilities definition

Outputs	- SPPR-6630 - System capability context
ID	SPPR-10605

P3.1.4.2 - Define required functions and functional exchanges
Steps:

1. **Identify among the already defined functions and functional exchanges** the ones that are needed to transition from the precondition to the post-condition of the system capability of interest. Include them into SPPR-6633 - Functions involvement based on SPPR-11016 - Construction method functional flow definition.
2. If the functions and exchanges have already been created previously, then **reuse** them if they fulfil the need of particular capability.
3. If there are no existing functions and exchanges defined, then **define new functions and exchanges** so that they fulfil the need of the particular capability.
4. **Allocate exchange items to functional exchanges** by following SPPR-10055 - Construction method exchange items
5. Document the rationale for each design decision.

Inputs	- SPPR-6633 - Functions involvement
Outputs	- SPPR-6633 - Functions involvement
ID	SPPR-10607

P3.1.4.3 - Describe behaviour with exchange scenarios
Steps:

a. Define pre- and postcondition see *SPPROCESS/10 SEMP V 01_01/SEMP Annex M ARCADIA Capella Modelling Rules : 722555* for more information.
b. Follow SPPR-7249 - Construction method exchange scenario to create SPPR-8176 - Exchange scenario diagram.

Notes:

- If the person creating the output view knows enough about how the system works with the actors for this capability, they can create the diagram directly from their own knowledge and must only develop a view that complies with all the modelling rules.
  If, on the other hand, it is necessary to gather information from others, a brainstorming workshop should be held.
- Where new functions or functional exchanges are identified, these should be first created on the corresponding view.

Inputs	- Functions - Functional exchanges - Exchange items - Actors
Outputs	- SPPR-8176 - Exchange scenario
ID	SPPR-10606

P3.1.4.4 - (Optional) Describe control loops with functional chains
Steps:

1. Follow SPPR-11021 - Construction method functional chain to create SPPR-8178 - Functional chain description diagram.
2. Follow SPPR-11315 - Construction method control loop to create SPPR-11362 - Control loop diagram.

Inputs	- Functions - Functional exchanges - Actors - Systems
Outputs	- SPPR-11362 - Control loop - SPPR-8178 - Functional chain description
ID	SPPR-11458

P3.1.4.5 - (Optional) Define system and actor states

Description:

Define the states under which the system is to

1. be capable of performing the function,
2. commence performing that function and
3. cease performing that function.

The goal is understand the state-based behaviour of the system and understand the differences in the behaviour of system actors based on states. The task should support the analysis to sketch out and express high-level / broad system states based on inputs and outputs via the system interfaces.

1. Create the state machine diagram for systems or actors.
2. Update states
3. Map system/actors functions to states.
4. Create transitions between the states.
   Note: It is strongly recommended to use the identified incoming system exchange items (via the external system interfaces) as triggers. For the actions, use exchange items which are outgoing. Please use structured natural language to write triggers, guard conditions and actions.
   For each state dimension (that is, each individual state machine or state machine region on the system):
   a. If a system function is only available in certain system states, allocate the function to those states where the function is available.
   b. If a system function is available independent of this state dimension, allocate the function to all states in this dimension.

Notes:

This task shall be performed in parallel to tasks Describe system capability with exchange scenarios.

Inputs	- System - Actors
Outputs	- System states - Actor states
ID	SPPR-10604

2.2.5 P3.1.5 Consolidate system definition

P3.1.5 - Consolidate system definition

Description:
This activity includes the creation, consolidation and review of the system definition so that the risk analysis can be performed and the system requirements can be derived.

Steps:
Consolidate functionality to maximise the reuse of system functions and their functional exchanges in more than one capability.

1. **Perform** an analysis of **duplicated system functions in SPPR-9580 - Functional flow consolidation diagram**, regarding their name, allocation and realised links.
2. **Merge functions** by re-linking/re-allocating the changed functions in the Consolidated functional flow definition in order to disconnect the duplicated functions from the rest of the model - only if the commonalities are clear and if they are identical or can be made identical by redesigning them.
3. **Delete** involvements of the duplicated functions which will be removed in the corresponding functional chain definition and exchange scenario definition.
4. **Replace** the **duplicated functions** with the correct merged functions in the corresponding functional chains, exchange scenarios, functional flows.
5. **Remove duplicates** from the database, if they have been fully disconnected from the rest of the model (no more involvement in capabilities, scenarios, or functional chains and no functional exchanges).
6. **Review** that the **functions** displayed in the Consolidated functional flow definition are all the functions defined in the database, i.e. all **functions** are connected to each other and there are no unused elements are left (e.g. functional exchanges without exchange items).

Consolidate exchange items and data to maximise the reuse of exchange items and create new ones if necessary.

1. **Review** completeness and consistency of defined exchange items in relation to defined system functions and system functional exchanges using system data flow diagram.
2. If the functional exchanges do not have allocated exchange items, **refine** the **data model.**
3. **Allocate** new **exchange items** to the functional exchanges.

Consolidate traceability

1. **Review the relationships** between each model element
    a. Review the correctness of the relationship if it exists
    b. Create a relationship between the element and a corresponding model element at the higher level if it not exists.

Perform review

1. **Transfer content** to Polarion using the Capella2Polarion bridge.
2. **Perform quality review** with EET.
3. **Perform** *SPPROCESS/SEMP Annex D Processes/Work product review process : 722555*.

Notes:

- The System Definition should be created as early as possible to support PRAMS activities.
- Further guidance can be found in Template - System Definition
- Functions and functional exchanges can have duplicates because of the development in silos of capabilities.

- The analysis of the realisation links can be done with a traceability report.
- If no realisation links can be generated automatically for a given model element, establish a new generic trace relationship between them.

Inputs	- Defined functions and functional exchanges - Defined interfaces - Allocation of functions - Draft System Definition
Outputs	- Reviewed System Definition - Traceability report
ID	SPPR-5209

2.3 P3.2 Provide input for risk and hazard analysis

P3.2 - Provide input for risk and hazard analysis
Description:
This process task conducts risk and hazard analysis. Together with PRAMSS team, risks the system will need to handle will be identified and developed. The analysis includes all types of hazards and risks PRAMSS, economic risks, market risks, risks for the stakeholders like migration risks).

Steps:

1. **Transfer** content to Polarion using the Capella2Polarion bridge.
2. The tasks of risks and hazards related to PRAMS (Performance, Reliability, Availability, Maintainability) is generally defined by the *SPPRAMS/Phase 3/ERJU PRAMS Plan : 722555* which needs to be followed.

Inputs	System Definition based on defined model transferred to Polarion
Outputs	Defined by PRAMS Plan
ID	SPPR-5204

2.4 P3.3 Define system requirements

P3.3 - Define system requirements

Description:

This task describes the steps to specify the black-box system requirements what the system of interest shall do, independent of its internal implementation. It defines the external behavior of the system in terms of inputs, outputs, and interactions with external entities. This ensures that higher level requirements are translated into system-level requirements, enabling clear validation criteria and traceability without constraining the design solution. The task considers also the consistency with the functional boundaries, functions, constraints, identified interfaces, critical performance measures and critical qualities.

Steps:

1. Transfer content to Polarion using the Capella2Polarion bridge.
2. Identify model elements from functions, exchange scenarios, exchange items and states of the system that will serve as an input for functional requirements defined during previous process tasks.
3. Identify or receive relevant system requirements that relate to risks, criticality, critical qualities and compliance with standards and regulations from relevant sources.
4. Define black-box requirements and their attributes inline with Requirements Management Plan and *SPPROCESS/SEMP Annex R Requirements/SEMP Annex R3 - Rules for writing textual requirements : 722555*.
    a. Define the rationale and other relevant attributes for each requirement to justify their existence.
    b. Define the source/origin of each requirement either by traceability links to higher level requirements. The key is to ensure consistency and completeness of requirements.
    c. Establish traceability between captured (child) black-box requirements to the (parent) higher level requirements, where appropriate.
5. Link/trace the requirements to the related view/model elements to maintain traceability.
6. Create the system requirement specification based on Template - System Requirements Specification

Notes:

- Whole diagrams (views) cannot be accepted as a requirement when considering *SPPROCESS/SEMP Annex R Requirements/SEMP Annex R3 - Rules for writing textual requirements : 722555*. In addition, SPPR-2681 - [EN 50126-1:2017] demands that each function must be explicitly and individually identified and managed.
- Always ensure those requirements can be tested at your level of abstraction.
- Well written requirements reduce the effort of requirement verification and requirement validation and avoid issues like misinterpretations that may result in a poor system solution.

Inputs	- Functions and exchanges are defined - Exchange scenarios are defined - System/actor states are defined - Interfaces - Actors - Risk analysis results
Outputs	- Set of system requirements - System Requirements Sprecification
ID	SPPR-5206

2.5 P3.4 Consolidate system requirements

P3.4 - Consolidate system requirements
Description:
Consolidate a consistent set of requirements which describes the system of interest in the system requirement specification.
Analyse/check the requirements for characteristics of individual requirements and for the whole set of requirements (e.g. whole document, current increment, build or sprint). In addition, check that each requirement provides consistent traceability.

- Ensure each requirement is necessary, appropriate to their level, unambiguous, complete, singular, feasible, verifiable, correct and conforming.
- Ensure the set of requirements is complete, consistent, feasible, comprehensible, correct and able to be validated.
- Ensure also that especially black-box system requirements are not introducing unnecessary implementation biases (i.e. are really necessary).
- For existing or legacy system being reused, analyse them based on factors such as applicability, feasibility, availability, quality, cost effectiveness, value and currency. While reusing requirements, a careful consistency check of reused requirements with the overall system has to performed in order to assure consistency (see also SPPR-10493 - ISO/IEC/IEEE 29148:2018).

Steps:

1. The requirement author requests one or more experts to review the quality of the requirements, i.e. whether they fulfill the Requirements Management Plan - Annex R1 Requirements Rules.
2. The requirement author requests or or more experts to review the traceablity of the requirements, i.e. whether they fulfill the linking as defined by SPPR-7265 - System Pillar traceability map for requirements and corresponding individual link rules.
3. The review can be performed informally, or performed as part of a formal review process *SPPROCESS/SEMP Annex D Processes/Work product review process : 722555*.
4. The findings have to be resolved with the appropriate stakeholders (i.e. experts).

Note:
Well written requirements reduce the effort of "requirement verification" and "requirement validation" and avoid issues such as misinterpretations that may result in a poor solution/system or insufficient or incorrect performed tests.

Inputs	- Set of requirements
Outputs	- Consolidated set of requirements
ID	SPPR-11477

3 Appendix

3.1 Standards and References

There are currently no specific references used by this document.