GfSE SAF System Architecture Framework © 2021 GfSE	**SAF Profile for Sparx Enterprise Architect** SAF Viewpoint One Pager	Parkan, Eric Täubrich, Jan Weiland, Kalle

Introduction and purpose of this document:

The document provides an overview of SAF viewpoints and the profile implementation in Sparx Enterprise Architect. The viewpoint one pagers support modeling activities in the project. For each viewpoint listed the document describes used stereotypes, their interdependencies, an example, and a workflow how to create a high-quality view in the model.
This document is NOT …

- … an introduction to MBSE nor Sparx Enterprise Architect - basic knowledge is assumed
- … an explanation of architecture frameworks in general - basic knowledge is assumed
- … an introduction of viewpoint-based architecture documentation as per ISO 42010
- … a documentation of the SAF specification – is documented on GitHub
- … a guideline to assemble engineering artefacts such as a System Requirement Specification.

Configuration matrix:

Document version	1.0 / last change 11/2/2022
Sparx EA tool version	15.2
MDG Technology EA SAF Profile version	1.0 / Initial-Release
SysML version	1.5 (please note that SysML1.5 implementation in EA reuses frequently SysML1.4 stereotypes which are used in the SAF viewpoints)
Used MDG Technologies	• Basic UML© 2 Technology • Core Extensions • MDG Technology Builder • SysML 1.5
SAF Concept Model version	Initial-Release: https://github.com/GfSE/SAF-Specification/releases/tag/Initial-Release

Content

The SAF Grid

The following overview shows the implemented SAF viewpoints as EA profile. Each viewpoint is detailed in a dedicated one pager which highlights the purpose, recommends a workflow, and shows an example. Furthermore, required input and output for each viewpoint is explained as the SAF grid provides navigational support for your modeling activities by structure the separate viewpoints.

class Overview

Glossary	1 STORY CONTEXT & EXCHANGE	2 STRUCTURE	3 PROCESS & BEHAVIOR	4 INTERACTION & COLLABORATION	5 INTERFACE	6 REQUIREMENT	7 SAFETY & SECURITY	8 MAPPING & CROSS REFERENCE
COMMON DOMAIN SCV								
OPERATIONAL DOMAIN SOV	SAF_OperationalDomain : SAF_OperationalStoryProfile SAF_OperationalDomain : SAF_OperationalContextDefinitionProfile SAF_OperationalDomain : SAF_OperationalContextExchangeProfile	SAF_OperationalDomain : SAF_OperationalDomainItemKindProfile SAF_OperationalDomain : SAF_OperationalPerformerProfile SAF_OperationalDomain : SAF_OperationalCapabilityProfile SAF_OperationalDomain : SAF_StakeholderIdentificationProfile	SAF_OperationalDomain : SAF_OperationalProcessProfile	SAF_OperationalDomain : SAF_OperationalInteractionProfile		SAF_OperationalDomain : SAF_StakeholderRequirementProfile		SAF_OperationalDomain : SAF_OperationalProcessTraceabilityProfile SAF_OperationalDomain : SAF_OperationalCapabilityTraceabilityProfile
FUNCTIONAL DOMAIN SFV	SAF_FunctionalDomain : SAF_SystemStoryProfile SAF_FunctionalDomain : SAF_SystemContextDefinitionProfile SAF_FunctionalDomain : SAF_SystemContextExchangeProfile	SAF_FunctionalDomain : SAF_SystemDomainItemKindProfile SAF_FunctionalDomain : SAF_SystemCapabilityProfile SAF_FunctionalDomain : SAF_SystemFunctionalBreakdownProfile	SAF_FunctionalDomain : SAF_SystemProcessProfile SAF_FunctionalDomain : SAF_SystemStateProfile	SAF_FunctionalDomain : SAF_SystemContextInteractionProfile		SAF_FunctionalDomain : SAF_SystemRequirementProfile		SAF_FunctionalDomain : SAF_SystemFunctionBlackBoxAllocationProfile
LOGICAL DOMAIN SLV								
PHYSICAL DOMAIN SPV								

Please note that the diagram describes the SAF viewpoint structure in grid form. It is not a suitable navigation tool to include all project's diagram.

SAF SOV01a Operational Story Viewpoint

This viewpoint documents the Operational Story(s) and their relation(s) to Operational Performer(s).

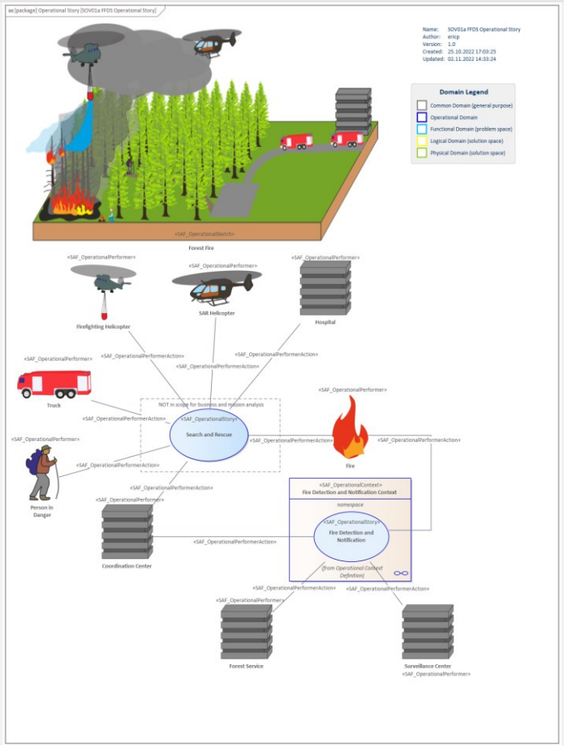
These represent the operational background from a Stakeholder's perspective. The Operational Story(s) themselves are usually provided by product experts, who represent the customer and/or user perspective within the company or organization.

Through the Operational Story Viewpoint, relevant Operational Performer are embedded in a larger operational background. Thus, the Operational Story Viewpoint is strictly located in the problem space of the system description, connects the Systems of Interest to their Operational Context and serves as a starting point to identify interfaces to stakeholders and/or context element(s).

General Recommendations and Known Pitfalls:

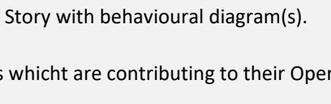
1. In addition, an illustration (drawing, sketch, etc.) and a description in free text provides a comprehensive understanding of the operational background. Ensure that the illustration and the story description are well understood by external stakeholders.
2. It is recommended to assign all Operational Stories that are to be further analyzed to an Operational Context. Operational Stories that are not assigned can thus be deemed "out of scope" for the project.
3. It is strongly recommended to document proposed Operational Story(s), even if some might not affect the System of Interest at first glance.
4. The Operational Story(s) should be prioritized so that it is clearly defined, which aspects are the most important to realize.
5. Depending on the project's development approach and the required target state of the operational concept it might be helpful to describe the ultimate intended concept or multiple incremental intermediate versions of it.

Example



Workflow:

1. Establish a dedicated package in your model for Operational Stories
2. Create a SAF Operational Story Diagram as specialized SysML1.5 Use Case Diagram for your operational stories in your package with Add Diagram > SysML 1.5 > UseCase > SAF_SOV01a_OperationalStoryDiagram.



3. Design a meaningful illustration of the operational events, performer und processes as a picture file and add the file as image asset (specialized artifact) to your model (please note that you don't need the image asset on your diagram)
4. Add an Operational Sketch on your diagram and change its appearance by applying the image asset. Make sure the name is shown under the image (all available in Appearance context menu)
5. Create Operational Story on your diagram with exemplary and relevant stories in prose. The text is inserted in the note field of the Operational Story stereotype.
6. Drag and drop already created Operational Performer from their dedicated package on the diagram who are involved in the operational stories.
   In case additional Operational Performer are identified here, add the Operational Performer(s) to the diagram and move them to their dedicated package afterwards.
7. Link Operational Stories with involved Operational Performers with the stereotype SAF Operational Performer Acting with the QuickLinker or the diagram's toolbox.
8. Move these Operational Stories which will remain in scope for the business and/or mission analysis below the Operational Context element in the Project Browser. Drag and drop the Operational Story onto the Operational Context element in the diagram.

Viewpoint Definition:

The viewpoint depicts the following stereotypes:

Stereotype	Description
Operational Sketch	• Illustration which summarizes graphically relevant operational performers, events and processes. The derivation of the required SOI should be possible for non-techncal stakeholders in the project • Can be created only in SOV01a • SAF stereotype is implemented as an extension of EA artefact
Operational Performer	• Involved participant in Operational Stories. Operational Performer can represent relevant persons, situations, or organizations and/or their units. • Can be created in SOV01a, SOV01b or in SOV02b • SAF stereotype is implemented as specialized SysML block
Operational Story	• Exemplary story which describes in prose a (non-complete) sucession of events with involved performers. The Operational Story highlights the necessity of the SOI • Can be created only in SOV01a • SAF stereotype is implemented as an extension of UML UseCase
Operational Performer Acting	• Associates an Operational Story with an Operational Performer who is acting in the story. • Can be created only in SOV01a • SAF stereotype is implemented as an extension of UML Association
Operational Context	• Defines the scope of the business and/or mission analysis in the operational domain • Owner of Operational Roles and Operational Stories • SAF stereotype is implemented as specialized SysML block

Viewpoint Input:

- **SAF SOV02b Operational Performer**:
  (De-)composes existing relevant Operational Performer. The Operational Performer can be used in this viewpoint.
- **SAF External:**
  The Operational Story contents and the foundations for the sketches and the performer illustrations are created with the assistance of product experts

Viewpoint Output:

- **SAF SOV02b Operational Performer**:
  Identifies relevant Operational Performer. The Operational Performer can be used in this viewpoint.
- **SAF SOV03a Operational Process**:
  Formalizes an Operational Story with behavioural diagram(s).
- **SAF SOV04a Operational Interaction**:
  Formalizes an Operational Story with behavioural diagram(s).
- **SAF SFV01a System Story**:
  Derives SOI specific stories whicht are contributing to their Operational Story

SAF SOV01b Operational Context Definition Viewpoint

The Operational Context Definition Viewpoints defines the scope for relevant business and/or mission analysis in the Operational Domain.

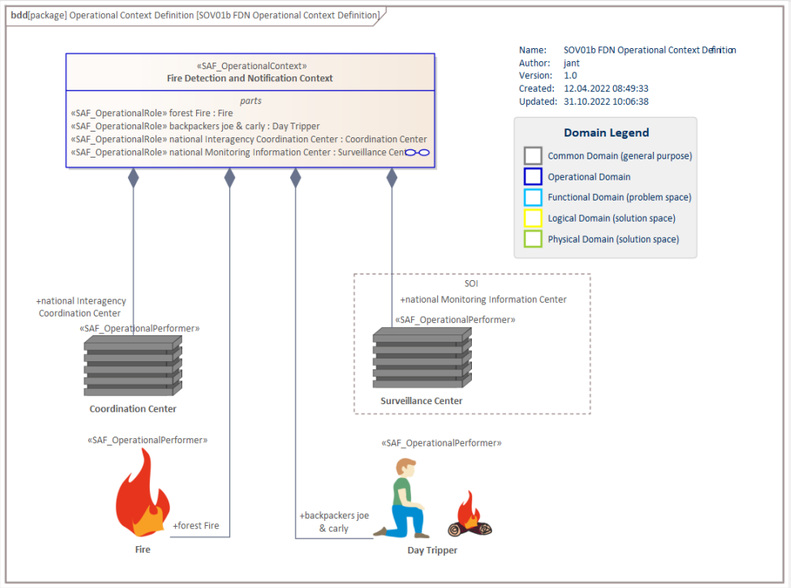
It defines the Operational Context that is required to achieve one or more Operational Stories by connecting the required Operational Performer(s) and resources to the context.

The Operational Context Definition Viewpoint's main purpose is to provide an anchor for the Operational Process(es) and Operational Interaction(s) and thus provide a coherence and consistency within the model. It is not a viewpoint that provides essential information for external stakeholders.

General Recommendations and Known Pitfalls:

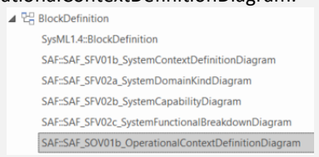
1. At least one Operational Context is required in model for the Operational Domain. It is possible however, to create several Operational Contexts with different scopes.
2. Different Operational Contexts should be defined if the scopes of these differ largely. This could be e.g., due to operational ambiguities between different Operational Stories or different operational scopes (e.g., different life phases).
3. While it is possible to create Operational Performer(s) in this viewpoint, it is recommended to manage and maintain them in the SAF SOV02b Operational Performer Viewpoint.

Example

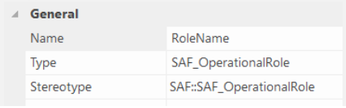


Workflow:

1. Create a new Operational Context Definition diagram as specialized SysML1.5 Block Definition Diagram in the dedicated package in your model with Add Diagram > SysML 1.5 > BlockDefinition > SAF_SOV01b_OperationalContextDefinitionDiagram.



2. Add an Operational Context to the diagram.
3. Drag and drop already available Operational Performer or create new Operational Performer in the diagram. New created Operational Performer must be moved to the dedicated package.
4. Create Operational Roles by using the composition relationship. Please note that the composition must be set from the Operational Context to the Operational Performer.
5. Give the Operational Role a meaningful name by selecting the part property in the Operational Context and amending the name on the property element page for traceability purposes in matrix specifications.



Viewpoint Definition:

The viewpoint depicts the following stereotypes:

Stereotype	Description
Operational Context	• Defines the scope of the business and/or mission analysis in the operational domain • Owner of Operational Roles • SAF stereotype is implemented as specialized SysML block
Operational Performer	• Involved participant in Operational Stories. Operational Performer can represent relevant persons, situations, or organizations and/or their units. • Can be created in SOV01a, SOV01b or in SOV02b • SAF stereotype is implemented as specialized SysML block
Composition	• Defines the part-of relationship between the Operational Context (parent) and Operational Perfomer (child) • Creates the usage of Operational Performer in the Operational Conext as Operational Roles
Operational Role	• Usage of Operational Performer in the scope on an Operational Context • SAF stereotype is implemented as extension of part property

Viewpoint Input:

- **SAF SOV02b Operational Performer**:
  Connects existing Operational Performers to Operational Context(s).
- **SAF SOV01a Operationa Story:**
  The Operational Stories are required to define the scope of an Operational Context

Viewpoint Output:

- **SAF SOV02b Operational Performer:**
  Identifies relevant Operational Performer. The Operational Performer can be used in this viewpoint.
- **SAF SOV01c Operational Context Exchange:**
  Defines what is being exchanged between the Operational Roles within the Operational Context.
- **SAF SOV03a Operational Process:**
  Refines Operational Stories that are happening within the defined Operational Context.
- **SAF SOV04a Operational Interaction:**
  Refines Operational Stories that are happening within the defined Operational Context.

SAF SOV01c Operational Context Exchange Viewpoint

This viewpoint documents and aides the definition of the connectivity and exchange between Operational Roles (which represent Operational Performers) within an Operational Context.

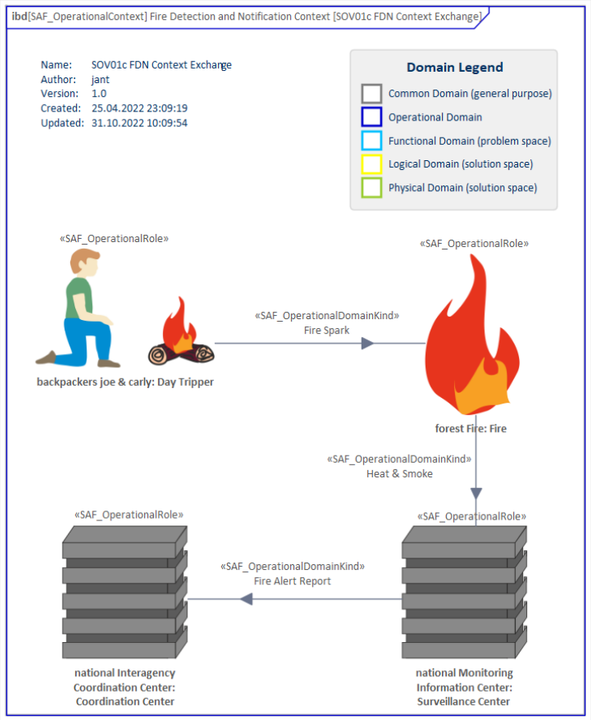
The exchanges can be anything (such as: information, material, energy…) and are documented with a direction. An Operational Context Exchange Viewpoint is always connected to only one Operational Context.

The Operational Context Exchange Viewpoint shows which Operational Roles interact with which other Operational Roles and what the exchange, but it does not provide details on how they interact.

General Recommendations and Known Pitfalls:

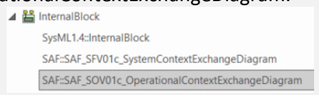
1. It is recommended to agree on which details are exposed on one diagram: only essentials, all known exchanges, intended exchanges and/or unwanted disturbances.
2. It is possible to create several Operational Context Exchange views for one Operational Context. This can be helpful to focus the Operational Context Exchange Viewpoint on specific issues or situations within the Operational Context.
3. The Exchanges should incorporate everything that is actually exchanged between Operational Performers within the given Operational Context and the scope of the Operational Context Exchange Viewpoint. If the Viewpoint gets too confusing, it can help to color code different types of exchange Item Flows (e.g. material flow, energy flow, information flow).
4. When there is a specific Operational Domain Kind missing to type an exchange, it should be created in SAF SOV02a Operational Domain Item Kind Viewpoint before being used in this viewpoint.

Example

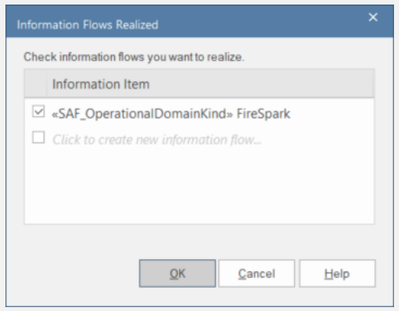


Workflow:

1. Select an Operation Context and add a new Operational Context Exchange Diagram as a specialized SysML1.5 Internal Block Diagram with Add Diagram > SysML 1.5 > InternalBlock > SAF_SOV01c_OperationalContextExchangeDiagram.



2. Drag and drop the involved Operational Roles from the Operational Context onto the Operational Context Exchange Diagram.
3. If the Operational Role exchange information, energy, or material establish an Item Flow and select the according Operational Domain Kind.
4. Create a Connector between the Operational Roles.
5. Select the Connector > right-click on the Connector > Advanced > Information Flow Realized > Select the available Operational Domain Kind(s).



Viewpoint Definition:

The viewpoint depicts the following stereotypes:

Stereotype	Description
Operational Role	• Usage of Operational Performer in the scope of an Operational Context • Should be created in SOV01b • Could be depicted as picture to increase project's stakeholder acceptance • SAF stereotype is implemented as extension of part property
Operational Domain Kind	• Represents material, energy or information specific for your domain • Using value types and enumerations it can include a quantative statement with values and units • Should be created in SOV02a • SAF stereotype is implemented as specialized SysML block
Item Flow	• Defines the flow of an Operational Domain Kind between two Operational Roles
Connector	• Hosts one or more Item Flows between two Operational Roles • A connector can host multiple Item Flows. It is highly recommended only to host Item Flows with the same direction.

Viewpoint Input:

- **SAF SOV01b Operational Context**:
  The Operational Context defines the scope in which the Operational Roles interact.
- **SAF Operational Role:**
  The Operational Roles have to be defined in order to define the exchanges between them.
- **SAF SOV02a Operational Domain Item Kind:**
  The Operational Domain Kinds define what is being exchanged between the different Operational Roles within one or more Operational Context(s).
- **SAF SOV01a Operational Story:**
  The Operational Stories provide input on the exchanges between the different Operational Roles involved in an Operational Context.

Viewpoint Output:

- **SAF SOV03a Operational Process:**
  Refines the exchanges between Operational Roles within given processes.
- **SAF SOV04a Operational Interaction:**
  Refines the communication between Operational Roles within given processes.
- **SAF SOV02d Stakeholder Identification:**
  The Operational Roles and their exchanges are considered for the Stakeholder and Stakeholder Concern identification.
- **SAF SOV02a Operational Domain Item Kind:**
  Defines Operational Domain Kinds when there is none yet existing for an Item Flow between two Operational Roles identified in this viewpoint.

SAF SOV02a Operational Domain Item Kind Viewpoint

This viewpoint provides an overview of all relevant and needed exchange types between Operational Performer in the Operation Domain.

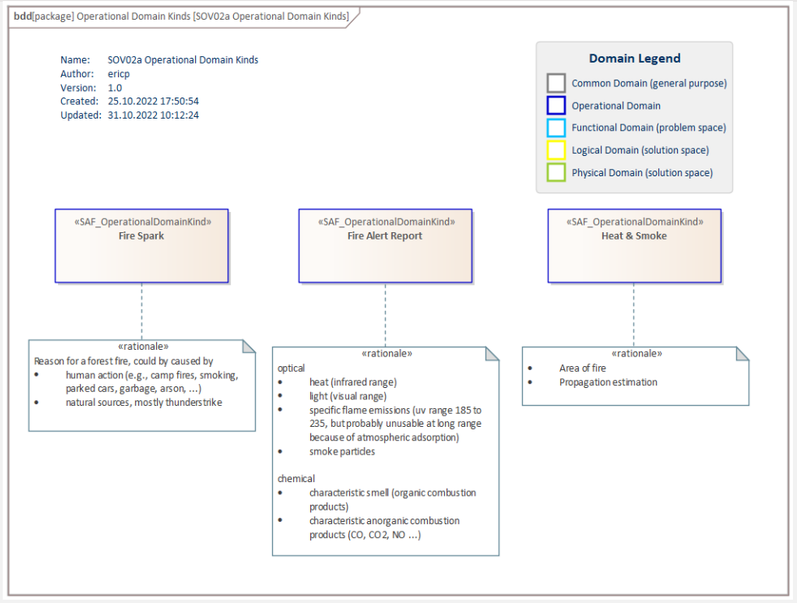
If required, it is also possible to decompose Operational Domain Kinds in this viewpoint, but other than that no relations are documented here. If additional information can aide a common understanding of an Operational Domain Kind, it is best documented in this viewpoint.

The Operational Domain Item Kind Viewpoint provides information for all modelers working on the project. It is not necessarily a viewpoint for communication with external stakeholders, though.

General Recommendations and Known Pitfalls:

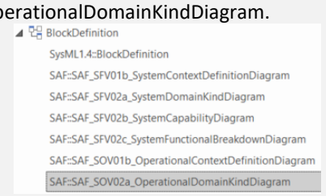
1. It is recommended to define Operational Domain Kinds only in this viewpoint.
2. For a common understanding of the different Operational Domain Kinds, it is helpful to provide written information on it. This is best done through a rationale for each Operational Domain Kind.
3. There should only be one Operational Domain Item Kind Viewpoint for the overall project.

Example



Workflow:

1. Create a new Operational Domain Kind diagram as specialized SysML1.5 Block Definition Diagram with Add Diagram > SysML 1.5 > BlockDefinition > SAF_SOV02a_OperationalDomainKindDiagram.



2. Create new Operational Domain Kind which is required to describe an exchange of information, energy, or material between Operational Roles.
3. If needed, you can formalize the Domain Kind attributes with value types or enumerations.
4. If needed, you can establish an Operational Domain Kind Composition relationship between two Operational Domain Kinds. Please note that the composition must be set from parent to child.

Viewpoint Definition:

The viewpoint depicts the following stereotypes:

Stereotype	Description
Operational Domain Kind	• Represents material, energy or information specific for your domain • Using value types and enumerations it can include a quantative statement with values and units • SAF stereotype is implemented as specialized SysML block
Composition	• Defines the part-of relationship between two Operational Dopmain Kinds if needed
Rationale	• Explains the reason why the Operational Domain Kind is needed and lists the essential details of the Operational Domain Kind

Viewpoint Input:

- **SAF SOV01a Operational Story:**
  The Operational Stories' descriptions provide information on the Operational Domain Kinds.
- **SAF SOV01c Operational Context Exchange:**
  The Operational Context Exchanges identify flows between Operational Roles that are typed by Operational Domain Kinds.

Viewpoint Output:

- **SAF SOV01c Operational Context Exchange:**
  Operational Domain Kinds are used to type Item Flows between Operational Performers.
- **SAF SOV03a Operational Process Viewpoint:**
  Operational Domain Kinds can be used to type action pins which refine exchanges between Process Actions.

SAF SOV02b Operational Performer Viewpoint

This viewpoint provides an overview of all relevant and (de-)composed Operational Performers and their relationships. In addition, it shows which Operational Performers are represented by which Stakeholders.

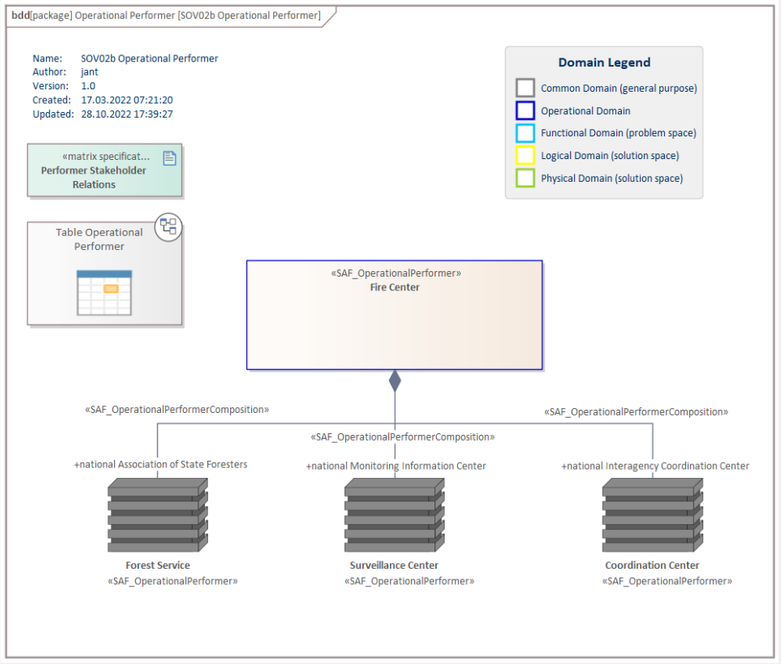
The Operational Performer Viewpoint provides information for all modelers working on the project. It is not necessarily a viewpoint for communication with external stakeholder, though.

Operational Performers can also be identified in other viewpoints (such as SAF SOV01b Operational Context Definition Viewpoint), but this viewpoint should display all of them to have one consistent overview.

General Recommendations and Known Pitfalls:

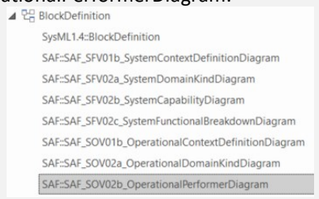
1. If Operational Performers are added to the model while working on other viewpoints, they are not automatically added to the Operational Performer diagram. In such a case, they must be added manually (see workflow description).
2. For better understanding for everyone involved in modeling, it is recommended to add descriptions to the Operational Performer "Text" attribute for each Operational Performer.

Example



Workflow:

1. Create a new Operational Performer Diagram in the dedicated package in your model as specialized SysML1.5 Block Definition Diagram with Add Diagram > SysML 1.5 > BlockDefinition > SAF_SOV02b_OperationalPerformerDiagram.



2. Drag and drop already identified Operational Performer or create new items on the diagram.
3. If needed, you can establish an Operational Performer Composition relationship between two Operational Performer. Please note that the composition must be set from parent to child.
4. Drag and drop already identified Operational Performer or create new Stakeholder on the diagram
5. Link the Stakeholder(s) to the Operational Performer which it represents with the Operational Stakeholder Representation stereotype.

Viewpoint Definition:

The viewpoint depicts the following stereotypes:

Stereotype	Description
Operational Performer	• Involved participant in Operational Stories. Operational Performer can represent relevant persons, situations, or organizations and/or their units. • Can be created in SOV01a, SOV01b or in SOV02b • SAF stereotype is implemented as specialized SysML block
Operational Performer Composition	• Specialized composition relationship to (de-) compose Operational Performer, if needed • SAF stereotype is implemented as extension of EA composition
Stakeholder	• Person, group, organization or one of its unit who represents an Operational Performer • Can be created in SOV02b or SOV02d • SAF stereotype is implemented as specialized SysML stakeholder
Operational Stakeholder Representation	• Links Stakeholder to Operational Performer to define a Representation relationship • SAF stereotype is implemented as extension of UML Association

Viewpoint Input:

- **SAF SOV01a Operational Story**:
  Operational Stories do involve Operational Performers. This is where all Operational Performers should originate.
- **SAF SOV01b Operational Context:**
  Operational Performers can be created in the Operational Context Definition View.

Viewpoint Output:

- **SAF SOV01a Operational Story**:
  Operational Stories do involve Operational Performers.
- **SAF SOV01b Operational Context:**
  Operational Performers are part of an Operational Context.
- **SAF SOV02d Stakeholder Identification:**
  Stakeholders represent Operational Performers with their Concerns.

SAF SOV02c Operational Capability Viewpoint

The Operational Capability Viewpoint documents and aides the definition of relevant Operational Capabilities and their relationships.

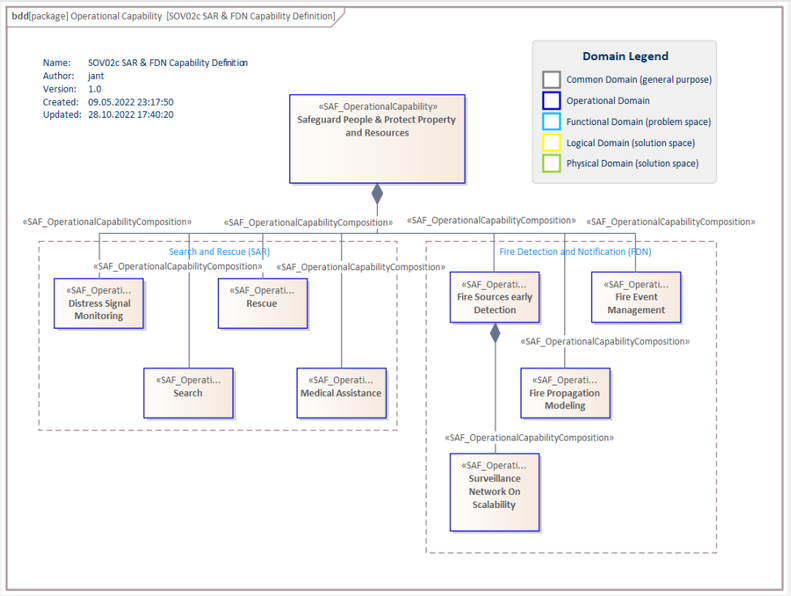
Operational Capabilities must be derived from Operational Stories by the modeler and are documented in this viewpoint. This process is not supported by the tool nor the SAF framework. An Operational Story is decomposed into Operational Capabilities that can be realized by Operational Performers.

The identified Operational Capabilities can be related to each other. They can be decomposed further, can be dependent on or can generalize other capabilities.

General Recommendations and Known Pitfalls:

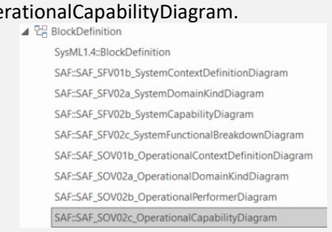
1. It is not necessarily required to define Operational Capabilities for a project. However, there are other frameworks (such as UAF) that integrate this step and build further development on these capabilities. If Operational Capabilities are used in the development process, they should be defined as per this viewpoint.
2. It is important to link Operational Capabilities to Operational Stories, Operational Interactions and/or Operational Processes to aide ensuring that the sum of the linked Operational Capabilities fully realize these. This process is not automized and not supported by the tool but must be ensured through reviews. Please note also SOV08b.

Example



Workflow:

1. Create a new Operational Capability Diagram in the dedicated package in your model as specialized SysML1.5 Block Definition Diagram with Add Diagram > SysML 1.5 > BlockDefinition > SAF_SOV02c_OperationalCapabilityDiagram.



2. Add new Operational Capabilities on the diagram.
3. Define Operational Capability Composition, Generalization and Dependencies between them. Please note that the composition must be set from parent to child.

Viewpoint Definition:

The viewpoint depicts the following stereotypes:

Stereotype	Description
Operational Capability	• Ability to perform a planned operational action. • SAF stereotype is implemented as specialized SysML block
Operational Capability Composition	• Relationship to (de-)compose an Operational Capability into detailed Operational Capabilities, if needed • SAF stereotype is implemented as extension of EA composition
Operational Capability Generalization	• Inheritence relationship to generalize/specialize an Operational Capability, if needed • SAF stereotype is implemented as extension of UML Generalization
Operational Capability Dependency	• Loose relationship between two Operational Capabilities which defines a dependency between them. • SAF stereotype is implemented as extension of UML Dependency

Viewpoint Input:

- **SAF SOV01a Operational Story**:
  Operational Capabilities perform a planned action within an Operational Story.
- **SAF SOV03a Operational Process:**
  Operational Processes can provide information on required Operational Capabilities.

Viewpoint Output:

- **SAF SOV06a Stakeholder Requirements:**
  Operational Capabilities are on input for the definition of Stakeholder Requirements.
- **SAF SFV02b System Capability:**
  System Capabilities contribute to Operational Capabilities covering a specific and defined System of Interest.

SAF SOV02d Stakeholder Identification Viewpoint

This viewpoint identifies, characterizes, and documents stakeholders for the system of interest and system context elements.

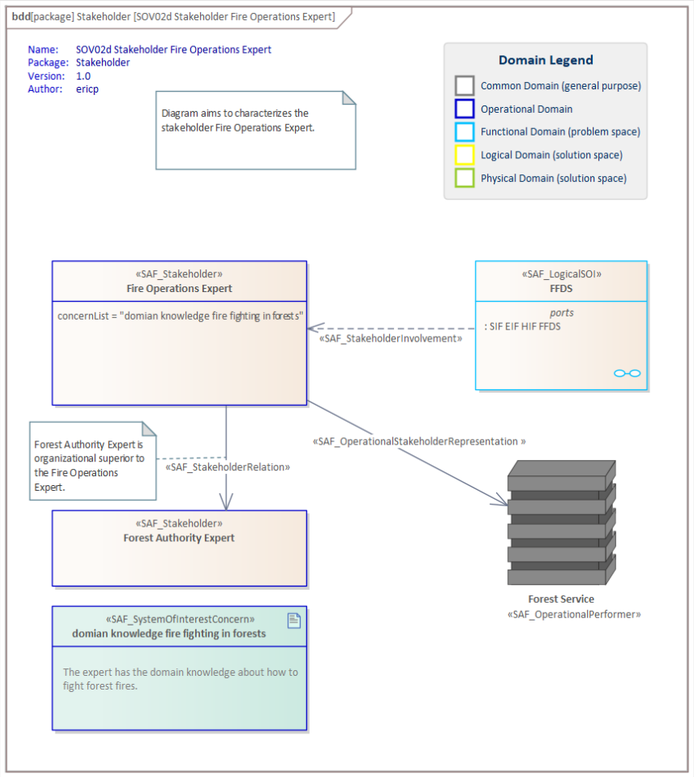
This means, that this viewpoint sits on the transition between the ConOps/OpsCon and the System of Interest definition. Therefore, it is important to first define the System of Interest, before starting to identify its stakeholders.

Stakeholders can represent one or more Operational Performers, but they can also represent concerns that cannot be derived from the ConOps/OpsCon.

General Recommendations and Known Pitfalls:

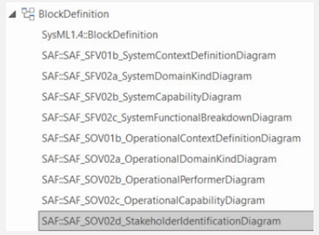
1. It is highly recommended to consider this viewpoint, even if no OpsCon/ConOps is documented in the project.
2. Stakeholders represent Operational Performers. However, other Stakeholders, that never occurred in the OpsCon/ConOps, and their concerns must also be respected. These may include, but are not limited to, company managers who make strategic product design or marketing decisions, authorities that grant releases for their respective markets or dealers who actually sell the product to customers.
3. Stakeholders and Stakeholder Concerns should be categorized in order to support a prioritization for the Stakeholder Requirements (see SAF SOV06a Stakeholder Requirement Viewpoint). The categories have to be defined by the project team.

Example

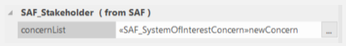


Workflow:

1. Create a new Stakeholder Identification Diagram in the dedicated package in your model as specialized SysML1.5 Block Definition Diagram with Add Diagram > SysML 1.5 > BlockDefinition > SAF_SOV02d_Stakeholder-IdentificationDiagram.



2. Create a Stakeholder on the diagram.
   If the Stakeholder has an interest in the System of Interest, create the System of Interest Concern on the diagram and select the System of Interest Concern in the Stakeholder's concern list in the property page. If the Stakeholder has multiple concerns press ctrl while selecting the elements.



3. If the Stakeholder is directly involved in the System of Interest (as an end user or representative) connect the Stakeholder and the System of Interest with the Stakeholder Involvement stereotype.
4. If the Stakeholder represents or has an interest in a System Context Element (such as the Logical Environment, Logical Context Element or as a Logical User) drag and drop the System Context element onto the diagram (or create a new one) and connect it with the stakeholder with the Context Element Characterization stereotype.
5. If the Stakeholder is related to one or more other stakeholders, drag and drop them onto the diagram (or create a new one) and connect them with the Stakeholder Relation stereotype.

Viewpoint Definition:

The viewpoint depicts the following stereotypes:

Stereotype	Description
Stakeholder	• Person, organizational representant or unit who represents an Operational Performer • SAF stereotype is implemented as specialized SysML stakeholder
Logical SOI	• Logical representation of the System of Interest. • Should be created in SFV01b. • SAF stereotype is implemented as specialized SysML block
Stakeholder Involvement	• Relationship to define a dependency between a Stakeholder and the Logical SOI • SAF stereotype is implemented as extension of UML dependency
System of Interest Concern	• Documents a Stakeholder's concern towards the SOI • SAF stereotype is implemented as extension of EA artefact
Stakeholder Relation	• Dependency relationship among Stakeholders • SAF stereotype is implemented as extension of UML dependency
Logical Context Element (Logical User, Logical Environment, Logical External System)	• Logical representation of a SOI's user, environment or external systems • Should be created in SFV01b. • SAF stereotype is implemented as specialized SysML block
Context Element Characterization	• Relationship which associates Logical Context Elements to Stakeholder • SAF stereotype is implemented as extension of UML association

Viewpoint Input:

- **SAF SOV02b Operational Performer**:
  Operational Performers are represented by Stakeholders to elicit requirements.
- **SAF SOV01a Operational Story:**
  Operational Stories provide input on System of Interest Concerns Stakeholders might have.
- **SAF SOV01c Operational Context Exchange:**
  Operational Context Exchanges provide input on System of Interest Concerns Stakeholders might have.
- **SAF External:**
  External domain experts (e.g. legal, standardization, quality…) need to be represented by Stakeholders.

Viewpoint Output:

- **SAF SOV06a Stakeholder Requirements:**
  Stakeholder Requirements are elicited from the identified Stakeholders.
- **SAF SFV01b System Context:**
  Logical Context Elements represent Stakeholders in the System Functional Domain.

SAF SOV03a Operational Process Viewpoint

The viewpoint formalizes an Operation Story (see SAF SOV01a Operational Story Viewpoint) by modeling a flow-based behavior of involved Operational Roles.

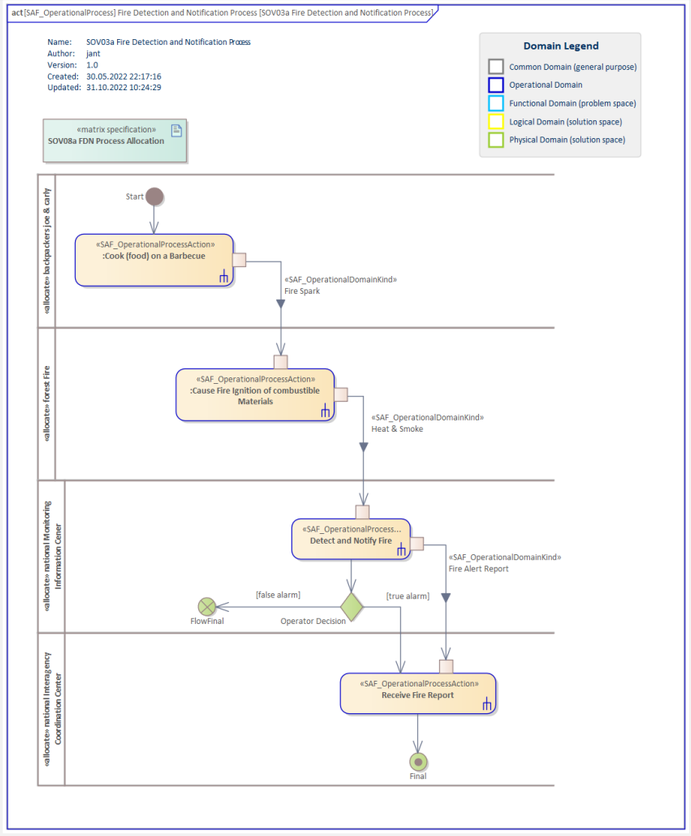
Also, the Operational Activities are assigned to the Operational Roles that are connected to the Operational Story that is being described. The breakdown of any Operational Story should be performed in cooperation with the stakeholders that pitched the Operational Story itself.

In the Operational Process diagram the Operational Activities are represented by Operational Process Actions and the Operational Performers by their Operational Roles.

General Recommendations and Known Pitfalls:

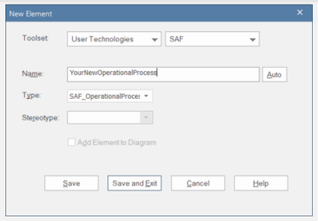
1. Due to tool restrictions, the Operational Process Actions are not automatically allocated to Operational Roles by assigning them to partitions in the diagram. Therefore, it is mandatory to maintain the allocation matrix as per SAF SOV08a Operational Process Traceability Viewpoint
2. Operational Process Actions have to be broken down until they can be unambiguously assigned to single Operational Roles.
3. If the focus of the modeling activity is to describe the interaction between the different performers and not on the flow of activities, it is recommended to use an Operational Interaction diagram instead.

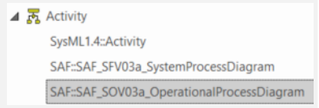
Example

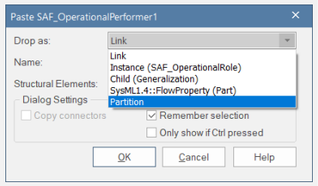


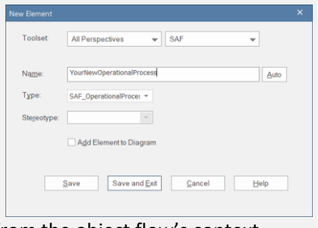
Workflow:

1. Create an Operational Process item in the dedicated package in your model to formalize an Operational Story or to expose a scenario.
2. Create a new SAF Operational Process diagram for the new Operational Process - right-click on the Operational Process, select New Child Diagram > Add Diagram > SysML 1.5 > Activity > SAF_SOV03a_Operational-ProcessDiagram
3. Drag and drop an involved Operational Performer onto the diagram and create an Operational Role by selecting partition in the dialog
4. Add Operational Process Actions to the according partition by drag and drop of existing Operational Processes and selecting instance.
5. For new Operational Process Actions create an Operational Process in the dedicated package with Add Element > Toolset: All Perspectives / SAF > Type: Operational Process. Proceed with step 4.
6. Model control and object flows by using the Quick Linker. For object flows you can set the information flow from the object flow's context menu with Advanced > Information Flows Realized. Click on the empty field to browse or search for the Operational Domain Kind that has been created previously.









Viewpoint Definition:

The viewpoint depicts the following stereotypes:

Stereotype	Description
Operational Role	• Usage of Operational Performer in the scope of an Operational Context • Should be created in SOV01b • SAF stereotype is implemented as extension of part property
Operational Process	• Definition of an Operational Flow • Does not appear visually on the diagram • SAF stereotype is implemented as extension of UML activity
Operational Process Action	• Usage of Operational Process in the scope of an Operational Context • SAF stereotype is implemented as extension of UML Call Behavior Actions
Allocate Activity Partition	• Partition-wise representation of an Operational Role • Any Operational Process Action located in the Allocate Activity Partition must have an allocate relationship to the Operational Role (allocate by usage pattern) • However, please note that EA does not set the allocate relationship automatically
Item Flow	• Defines the flow of an Operational Domain Kind between two Operational Process Actions
Matrix Specification	• Matrix-oriented traceability representation to set and maintain allocate relationship between Operational Process Actions and Operational Role (allocate by usage pattern)

For other notation elements which are exposed in the viewpoint, consider UML/SysML standard reference literature.

Viewpoint Input:

- **SAF SOV01a Operational Story:**
  Operational Stories are broken down into Operational Processes.
- **SAF SOV02b Operational Performer:**
  Operational Performers perform Operational Processes within an Operational Context.
- **SAF SOV01c Operational Context Exchange:**
  Operational Processes exchange items that are identified in Operational Context Exchange Views.

Viewpoint Output:

- **SAF SFV03a System Process:**
  System Processes contribute to Operational Processes but only cover one specific and defined System of Interest.
- **SAF SOV02c Operational Capability:**
  A consistency between Operational Processes and Operational Capabilities has to be ensured if both are defined in a project.
- **SAF SOV06a Stakeholder Requirements:**
  Operational Processes lead to Stakeholder Requirements.

SAF SOV04a Operational Interaction Viewpoint

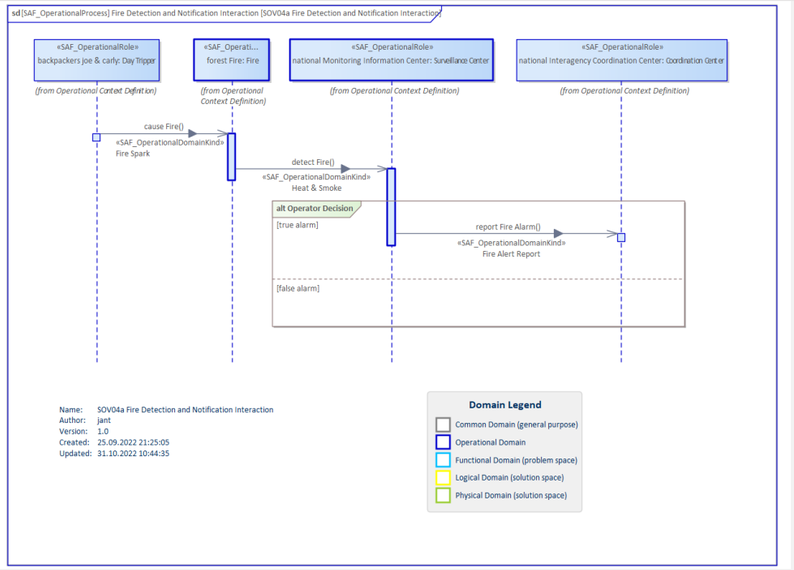
The Operational Interaction Viewpoint is an alternative method to SAF SOV03a Operational Process Viewpoint for formalizing an Operational Story and specifying the tasks of the Operational Performers involved in that story.

While the SAF SOV03a Operational Process Viewpoint details the flow of events and their operational exchange that contribute to an Operational Story, the Operational Interaction Viewpoint describes the sequence in which Operational Performers are interacting and the messages between them.

General Recommendations and Known Pitfalls:

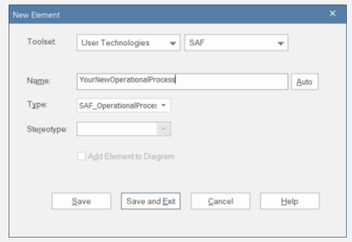
1. This viewpoint is an alternative viewpoint to SAF SOV03a Operational Process Viewpoint in that it serves the same purpose to detail Operational Stories. However, it does so differently by concentrating on the interactions between Operational Performers, whereas the Operational Process shows “internal” behavior of these performers as well. It should be decided, which viewpoint to use within a project and only use the alternative when necessary.

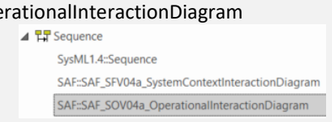
Example



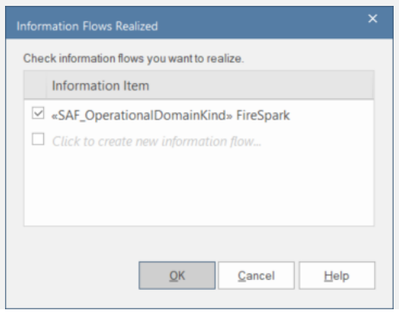
Workflow:

1. Create an Operational Process item in the dedicated package in your model to formalize an Operational Story or to expose a scenario.
2. Create a new SAF Operational Interaction diagram for the new Operational Process - right-click on the Operational Process, select New Child Diagram > Add Diagram > SysML 1.5 > Sequence > SAF_SOV04a_OperationalInteractionDiagram





3. Drag and drop Operational Roles onto the diagram which are involved in the interaction.
4. Define the sequence in the interaction with messages and fragments.
5. Select the message > right-click on the Connector > Advanced > Information Flow Realized > Select the available Operational Domain Kind(s).



Viewpoint Definition:

The viewpoint depicts the following stereotypes:

Stereotype	Description
Operational Role	• Usage of Operational Performer in the scope of an Operational Context • Should be created in SOV01b • SAF stereotype is implemented as extension of part property
Interaction Fragment	• Operator in interaction sequence
Message	• Exchange of material, energy or information between Operational Roles.
Item Flow	• Defines the flow of an Operational Domain Kind between two Operational Roles

Viewpoint Input:

- **SAF SOV01a Operational Story**:
  Operational Stories are broken down into Operational Interactions.
- **SAF SOV02b Operational Performer:**
  Operational Performers interact with each other within an Operational Context.
- **SAF SOV01c Operational Context Exchange:**
  Within Operational Interactions, Operational Performers exchange information and items identified in Operational Context Exchanges.

Viewpoint Output:

- **SAF SFV03a System Process:**
  System Processes contribute to Operational Interactions but only cover one specific and defined System of Interest.
- **SAF SOV02c Operational Capability:**
  A consistency between Operational Interactions and Operational Capabilities has to be ensured if both are defined in a project.
- **SAF SOV06a Stakeholder Requirements:**
  Operational Interactions lead to Stakeholder Requirements.

SAF SOV06a Stakeholder Requirement Viewpoint

The Stakeholder Requirement Viewpoint documents and manages Stakeholder Requirements for particular stakeholders which are relevant for the project.

Thus, the requirements are sorted by stakeholder in that every stakeholder has its own requirement package.

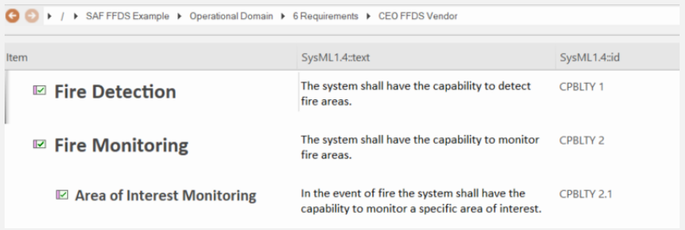
The Stakeholder Requirement Viewpoint considers different kinds of requirements, such as capabilities, functions, non-functional properties, and constraints.

General Recommendations and Known Pitfalls:

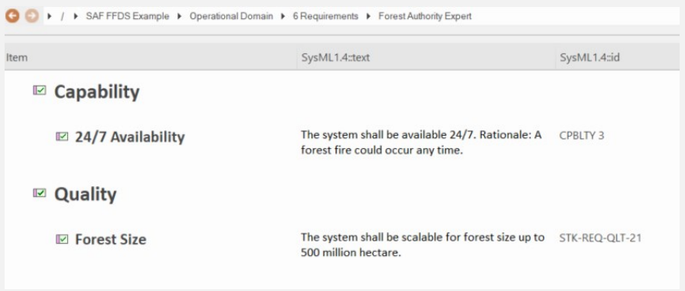
1. It is required to create one requirement package per stakeholder (as described in the workflow section).
2. If different stakeholders have similar or even the same requirement, these should still be described separately for every stakeholder and should not be shared between them. The consolidation of all requirements is subject of the system requirement analysis.
3. Stakeholder Requirements should be prioritized, so that it is clearly defined, which requirements are the most important to realize. Priorities can be based on Operational Story prioritization and/or Stakeholder categorization.
4. In case the project considers a given Customer Requirement Specification, the according Stakeholder Requirements should be managed in one dedicated stakeholder package.
5. The viewpoint does not stipulate a consolidation of contradicting Stakeholder Requirements. A careful consolidation has to be agreed within the project, depending on Stakeholders' priorities.

Example

FFDS Vendor:

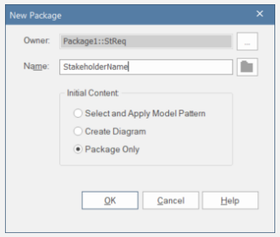


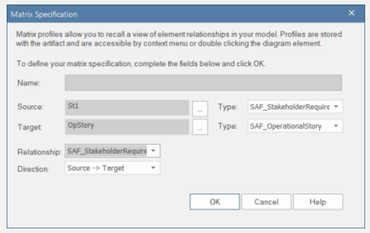
Forest Authority Expert:

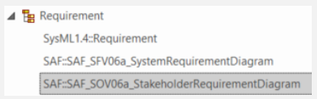


Workflow:

1. Create one package per identified stakeholder within the dedicated stakeholder requirement package.
2. Open the global Specification Manager (and not the Specification View of a specific diagram).
3. Select the package for a specific stakeholder.
4. Document your identified requirements with the Stakeholder Requirements stereotype.
6. If needed, create (and maintain) matrix specifications to relate Stakeholder Requirements to Stakeholder (using the Stakeholder Requirement Imposition stereotype), System of Interest Concerns (using the Stakeholder Requirement Refinement stereotype) or Operational Stories (using the Stakeholder Requirement Refinement stereotype).
7. If needed, create a new SAF Stakeholder Requirement diagram to analyze single stakeholder requirements - right-click on the dedicated stakeholder package, select New Child Diagram > Add Diagram > SysML 1.5 > Requirement > SAF_SOV06a_StakeholderRequirementDiagram. Create new Stakeholder Requirements or drag and drop existing Stakeholder Requirements onto the diagram. Link the Stakeholder Requirements to Operational Stories, System of Interest Concerns or Operational Stakeholder using the Quick Linker.







Viewpoint Input:

- **SAF SOV02d Stakeholder Identification**:
  Stakeholder Requirements are organized by the identified Stakeholders.
- **SAF SOV02c Operational Capability:**
  Operational Capabilities result in Stakeholder Requirements.
- **SAF SOV03a Operational Process:**
  Operational Processes lead to Stakeholder Requirements.
- **SAF SOV04a Operational Interaction:**
  Operational Interactions lead to Stakeholder Requirements.
- **SAF External:**
  Standards, internal processes or strategic decisions can lead to Stakeholder Requirements.

Viewpoint Output:

- **SAF System Requirements:**
  System Requirements are derived from Stakeholder Requirements.

Viewpoint Definition:

The viewpoint depicts the following stereotypes:

Stereotype	Description
Stakeholder	• Person, organizational representant or unit who represents an Operational Performer • Stakeholder are identified in OV02d • SAF stereotype is implemented as specialized SysML stakeholder
Stakeholder Requirements	• Requirement which is specific for one Stakeholder. • SAF stereotype is implemented as specialized SysML extendedRequirement
Stakeholder Requirement Imposition	• Relationship between Stakeholders and Stakeholder Requirements, if needed. • SAF stereotype is implemented as extension of UML Dependency
System of Interest Concern	• Documents a Stakeholder's concern towards the SOI • SAF stereotype is implemented as extension of EA artefact
Operational Story	• Exemplary story which describes in prose a (non-complete) sucession of events with involved performers. The Operational Story highlights the necessity of the SOI • Can be created only in SOV01a • SAF stereotype is implemented as an extension of UML UseCase
Stakeholder Requirement Refinement	• Relationship between an System of Interest Concern and Stakeholder Requirements, or an Operational Story and Stakeholder Requirements • SAF stereotype is implemented as extension of UML Abstraction

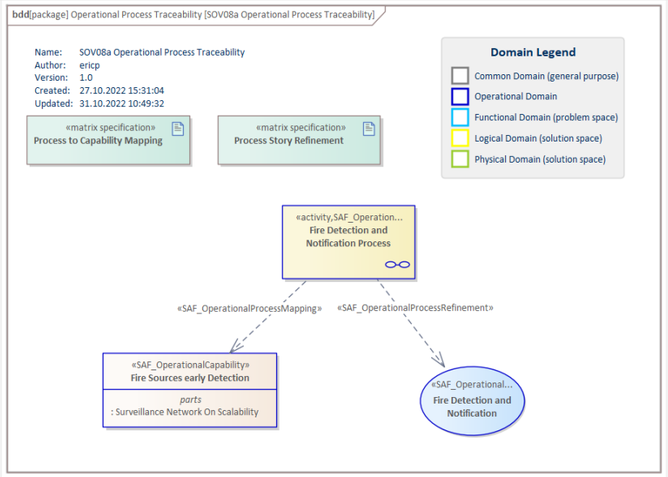
SAF SOV08a Operational Process Traceability Viewpoint

This viewpoint documents the dependencies between Operational Processes and Operational Performer as well as Operational Stories and Operational Capabilities. Thus, a traceability between these elements is established.

General Recommendations and Known Pitfalls:

1. There is not automatic tool support to ensure that the sum of all Operational Processes linked to one Operational Story fully cover this story. Thorough reviews are required to ensure this coverage.
2. If the Viewpoint SAF SOV02c Operational Capability is not utilized in a project, then no dependency matrix between Operational Capabilities and Operational Processes has to be established.

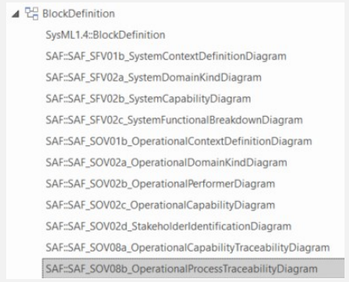
Example



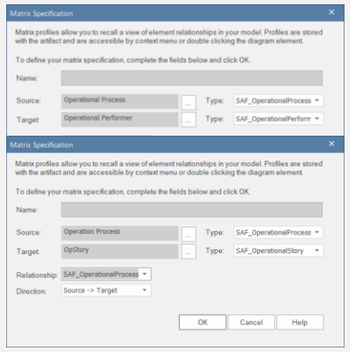
Source \ Target	Distress Signal Monitoring	Fire Event Management	Fire Propagation Modeling	Fire Sources early Detection	Medical Assistance	Rescue	Safeguard People & Protect Property and Resources	Search	Surveillance Network On Scalability
Cause Fire Ignition of combustible Materials									
Cook (food) on a Barbecue									
Detect and Notify Fire									
Fire Detection and Notification Process	↑	↑	↑	↑					
Receive Fire Report									

Workflow:

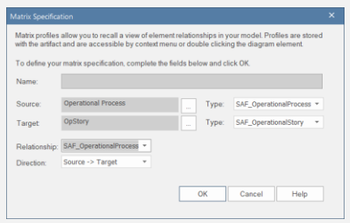
1. If needed, create an Operation Process Traceability diagram as specialized SysML 1.5 Block Definition Diagram with Add Diagram > SysML 1.5 > BlockDefinition >



   SAF_SOV08a_OperationalProcessTraceabilityDiagram.
2. Create (and maintain) a matrix specification to link Operational Processes to Operational Performer with the following configuration using the SysML allocation:
   Set the allocation link according to the involvement of the Operational Performer in Operational Processes.



3. Create (and maintain) a matrix specification to link Operational Processes to Operational Stories with the following configuration using the Operational Process Refinement stereotype:
   Set the dependency links according to which Operational Process refines which Operational Story.
4. Create (and maintain) a matrix specification to link Operational Processes to Operational Capabilities with the following configuration using the Operational Process Mapping stereotype:
   Set the dependency links according to which Operational Process can be mapped to Operational Capabilities.



Viewpoint Definition:

The viewpoint depicts the following stereotypes:

Stereotype	Description
Operational Process	• Definition of an Operational Flow • SAF stereotype is implemented as extension of UML activity
Operational Story	• Exemplary story which describes in prose a (non-complete) sucession of events with involved performers. The Operational Story highlights the necessity of the SOI • Can be created only in SOV01a • SAF stereotype is implemented as an extension of UML UseCase
Operational Process Refinement	• Relationship between an Operational Stories and required Operational Process(es) • SAF stereotype is implemented as an extension of UML dependency
Operational Capability	• Ability to perform a planned operational action. • Can only be created in SOV02c • SAF stereotype is implemented as specialized SysML block
Operational Process Mapping	• Relationship between Operational Process(es) and Operational Capabilities • Explains the reason to exist of Operational Capabilities based on Operational Process(es) • SAF stereotype is implemented as an extension of UML dependency
Matrix Specification	• Matrix-oriented traceability representation to set and maintain the refinement between Operational Stories and Operational Processes. Set and maintain the mapping between Operational Stories and Operational Capabilities.

Viewpoint Input:

- **SAF SOV03a Operational Process**:
  Operational Processes are identified and a traceability is created in this view.
- **SAF SOV01a Operational Story:**
  Operational Stories are identified and a traceability is created in this view.
- **SAF SOV02c Operational Capability:**
  Operational Capabilities are identified and a traceability is created in this view.

Viewpoint Output:

- **SAF External:**
  Traceability supports the quality control of the model, showing “lose ends” or inconsistencies. Also, it supports impact analyses for change management and assessments.

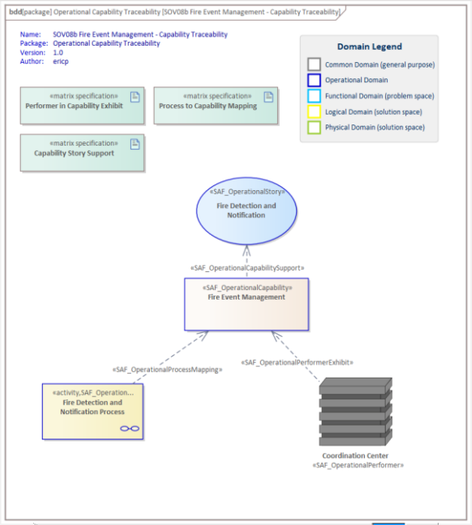
SAF SOV08b Operational Capability Traceability Viewpoint

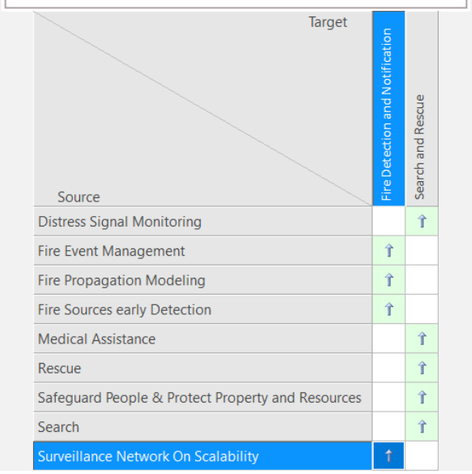
This viewpoint documents the dependencies between Operational Capabilities and Operational Stories as well as Operational Processes and Operational Performer. Thus, a traceability between these elements is established.

General Recommendations and Known Pitfalls:

1. There is not automatic tool support to ensure that the sum of all Operational Capabilities linked to one Operational Story fully cover this story. Thorough reviews are required to ensure this coverage.
2. If the Viewpoint SAF SOV02c Operational Capability is not utilized in a project, then no dependency matrix between Operational Capabilities and Operational Stories has to be established.
3. The traceability between Operational Capabilities and Operational Performer is not an allocation. Thus, it is not required to decompose an Operational Capabilities to map the sub-capability to an Operational Performer.

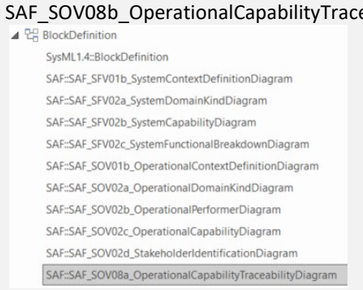
Example



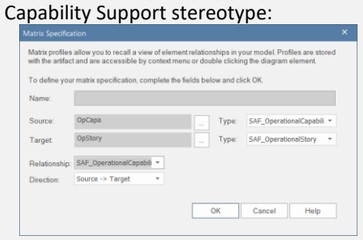


Workflow:

1. Create an Operational Capability Traceability diagram as specialized SysML 1.5 Block Definition Diagram with Add Diagram > SysML 1.5 > BlockDefinition > SAF_SOV08b_OperationalCapabilityTraceabilityDiagram.



2. Create (and maintain) a matrix specification to link Operational Capabilities to Operational Stories with the following configuration using the Operational Capability Support stereotype:



3. Create (and maintain) a matrix specification to link Operational Performer to Operational Capabilities with the following configuration using the Operational Performer Exhibit stereotype:

Viewpoint Definition:

The viewpoint depicts the following stereotypes:

Stereotype	Description
Operational Capability	• Ability to perform a planned operational action. • Can only be created in SOV02c • SAF stereotype is implemented as specialized SysML block
Operational Story	• Exemplary story which describes in prose a (non-complete) sucession of events with involved performers. The Operational Story highlights the necessity of the SOI • Can be created only in SOV01a • SAF stereotype is implemented as an extension of UML UseCase
Operational Capability Support	• Relationship between an Operational Story and required Operational Capabilities. • SAF stereotype is implemented as an extension of UML dependency
Operational Process	• Definition of an Operational Flow • SAF stereotype is implemented as extension of UML activity
Operational Process Mapping	• Relationship between Operational Process(es) and Operational Capabilities • Explains the reason to exist of Operational Capabilities based on Operational Process(es) • SAF stereotype is implemented as an extension of UML dependency
Operational Performer	• Involved participant in Operational Stories. Operational Performer can represent relevant persons, situations, or organizations and/or their units. • Can be created in SOV01a, SOV01b or in SOV02b • SAF stereotype is implemented as specialized SysML block
Operational Performer Exhibit	• Relationship between Operational Performer and Operational Capabilities • SAF stereotype is implemented as an extension of UML dependency

Viewpoint Input:

- **SAF SOV01a Operational Story:**
  Operational Stories are identified and a traceability is created in this view.
- **SAF SOV02c Operational Capability:**
  Operational Capabilities are identified and a traceability is created in this view.
- **SAF SOV02b Operational Performer:**
  Operational Performers are identified and a traceability is created in this view.

Viewpoint Output:

- **SAF External:**
  Traceability supports the quality control of the model, showing “lose ends” or inconsistencies. Also, it supports impact analyses for change management and assessments.

SAF SFV01a System Story Viewpoint

This viewpoint documents and aides the documentation of textual scenarios as System Stories involving the SOI. Thus, the benefit of the SOI is highlighted.

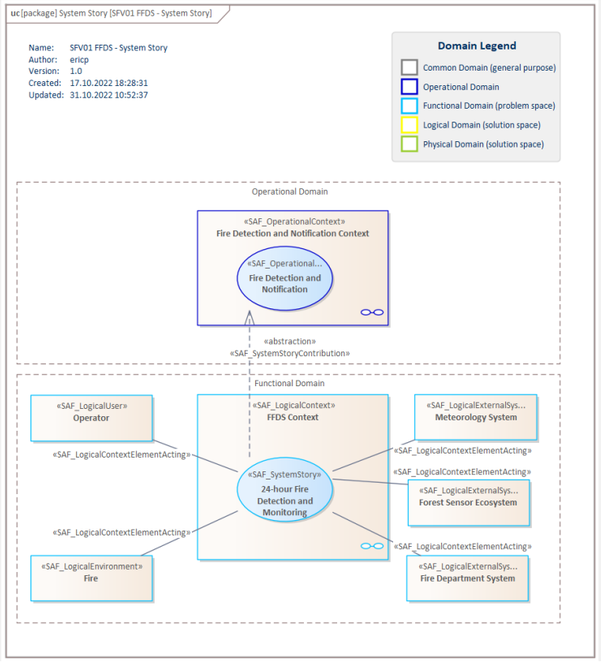
System Stories break down the Operational Stories with the focus on the System of Interest and thus contribute to Operational Stories. These contribution relationships can also be documented in this viewpoint. System Stories can be related to each other as well, such as one System Story being included in another System Story.

The actors involved in a System Story are represented as Logical Context Elements.

General Recommendations and Known Pitfalls:

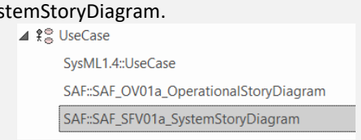
1. While the System Story is more specific than the Operational Story, it is still a black box description and should reside in the problem space. It should be avoided to go into technical details and solutions, but rather keep an outside perspective.
2. In the likely event that the System Story cannot fully cover an Operational Story it contributes to, the System Stories of the other contributing Systems do not need to be modeled. They are represented through the linked Logical Context Elements.

Example



Workflow:

1. Create a System Story Diagram as specialized SysML 1.5 Use Case Diagram with Add Diagram > SysML 1.5 > UseCase > SAF_SFV01a_SystemStoryDiagram.



2. Drag and drop already created Logical Context Elements (e.g., a Logical User, Logical External System, or the Logical Environment) from their dedicated package onto the diagram who are involved in the system stories.
   In case additional Logical Context Elements are identified here, add the Logical Context Elements onto the diagram and move them to their dedicated package afterwards.
3. Link System Stories with involved Logical Context Elements with the stereotype Logical Context Element Acting with the QuickLinker or the diagram's toolbox.
4. If needed, establish an include relationship between two System Stories Acting with the QuickLinker or the diagram's toolbox.
5. If needed, add existing Operational Stories to the diagram. Connect a System Story to these Operational Stories where it contributes to using the System Story Contribution stereotype.

Viewpoint Definition:

The viewpoint depicts the following stereotypes:

Stereotype	Description
System Story	• Exemplary story of the System of Interest which describes in prose a (non-complete) sucession of events with involved Logical Context Elements. • Can be created only in SFV01a • SAF stereotype is implemented as an extension of UML UseCase
Logical Context Element: • Logical User • Logical External System or • Logical Environment	• Logical context element which is a representation of an actual person, adjacent technical system or the environment the SOI is embedded in • SAF stereotypes are implemented as specialized SysML blocks
Logical Context Element Acting	• Relationship between a System Story and involved logical context element(s) • SAF stereotype is implemented as an extension of UML Association
Operational Story	• Exemplary story which describes in prose a (non-complete) sucession of events with involved performers. The Operational Story highlights the necessity of the SOI • Can be created only in SOV01a • SAF stereotype is implemented as an extension of UML UseCase
System Story Contribution	• Relationship between the contributing System Story and an Operational Story • SAF stereotype is implemented as an extension of UML abstraction
System Context	• Defines the scope of the functional analysis in functional domain. • Owner of Logical Roles and Logical Stories • SAF stereotype is implemented as specialized SysML block

Viewpoint Input:

- **SAF SOV01a Operational Story**:
  Operational Stories provide the Operational benefit that shall be achieved and the context in which the System Stories act.
- **SAF SOV02b Operational Performer**:
  Logical Context Elements that are used in this viewpoint can be based off of Operational Performers in the Operational Domain.
- **SAF SOV02c Operational Capability:**
  Operational Capabilities provide the Operational benefit that the System Stories contribute to.
- **SAF SOV03a Operational Process:**
  System Stories contribute to Operational Processes.
- **SAF SOV04a Operational Interaction:**
  System Stories contribute to Operational Interactions.
- **SAF SOV06a Stakeholder Requirements:**
  Stakeholders can contain specific requirements relevant for the elements defined in this viewpoint.

Viewpoint Output:

- **SAF SFV01b System Context:**
  System Context are defined based on the System Stories and the involved Logical Context Elements.
- **SAF SFV03a System Process:**
  System Processes formalize System Stories with a behavioral diagram(s).
- **SAF SFV04a System Context Interaction:**
  System Context Interactions formalize System Stories with a behavioral diagram(s).

SAF SFV01b System Context Definition Viewpoint

The System Context Viewpoint documents the System Context(s) and the involved Logical Context Element(s). Thus, it defines the SOI and the SOI's relevant system context of the SOI.

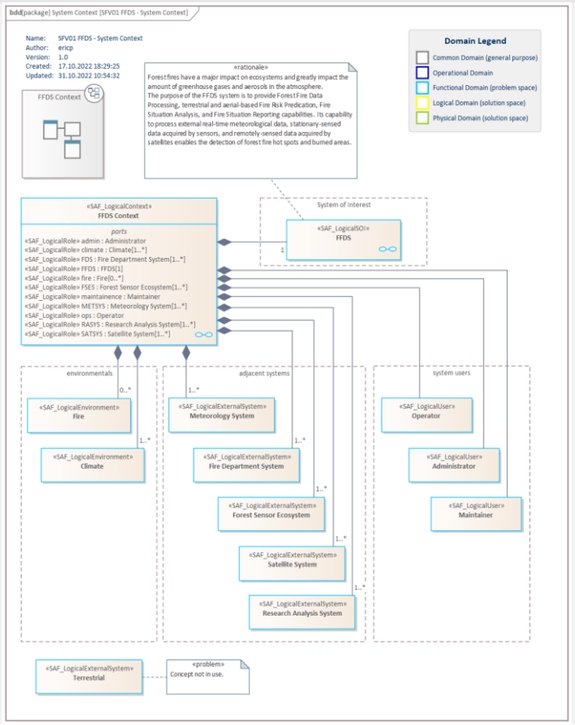
It defines the System Context that is required to achieve one or more System Story(s) by connecting Logical Context Elements (e.g., users, external systems, or environment) to the Context.

The System Context Definition Viewpoint's main purpose is to provide an anchor for the System Process(es) and System Interaction(s) and thus provide coherence and consistency within the model. It is not a viewpoint that provides essential information for external stakeholders.

General Recommendations and Known Pitfalls:

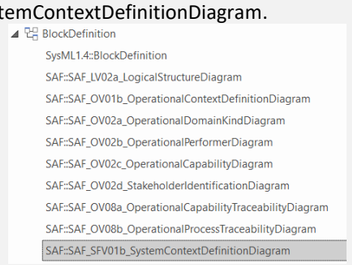
1. At least one System Context is required for a model. It is possible however, to create several System Contexts with different scopes. However, a holistic system context view has to be created.
2. Different System Contexts should be defined if the scopes of these differ largely. This could be e.g. due to functional ambiguities between different System Stories or different System scopes (e.g. different life phases).
3. All System Stories should be covered by a System Context.

Example

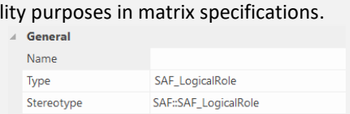


Workflow:

1. Create a new System Context Definition diagram as specialized SysML1.5 Block Definition Diagram in the dedicated package in your model with Add Diagram > SysML 1.5 > BlockDefinition > SAF_SFV01b_SystemContextDefinitionDiagram.



2. Create a Logical System Context to the diagram.
3. Create a Logical SOI and Logical Context Elements (e.g., a Logical User, Logical External System, or the Logical Environment).
4. Create Logical Roles by using the composition relationship. Please note that the composition must be set from the System Context to the Logical SOI and Logical Context Elements.
5. Give the Logical Role a meaningful name by selecting the part property in the Operational Context and amending the name on the property element page for traceability purposes in matrix specifications.



6. If needed, define the multiplicity for the Logical Roles.

Viewpoint Input:

- **SAF FV01a System Story**:
  The System Stories are required to define the scope of a System Context and provide Logical Context Elements that act within System Context(s).
- **SAF SOV01b Operational Context:**
  An Operational Context definition influences the scope of a System Context.
- **SAF SOV02b Operational Performer:**
  Logical Context Elements that are used in this viewpoint can be based off of Operational Performers in the Operational Domain.
- **SAF SOV06a Stakeholder Requirements:**
  Stakeholders can contain specific requirements relevant for the elements defined in this viewpoint.

Viewpoint Output:

- **SAF FV01c System Context Exchange:**
  Defines what is being exchanged between the Logical Context Elements and the SoI within the System Context.
- **SAF FV03a System Process:**
  Refines System Stories happening within a defined System Context.
- **SAF FV04a System Context Interaction:**
  Refines System Stories happening within a defined System Context.

Viewpoint Definition:

The viewpoint depicts the following stereotypes:

Stereotype	Description
System Context	• Defines the scope of the functional analysis in functional domain. • Owner of Logical Roles and Logical Stories • SAF stereotype is implemented as specialized SysML block
Logical SOI	• Logical representation of the System of Interest • SAF stereotype is implemented as specialized SysML block
Logical User	• Logical context element which is a representation of an actual person. • SAF stereotype is implemented as specialized SysML block
Logical External System	• Logical context element which is a representation of a technical adjacent system the SOI has to inteact with. • SAF stereotype is implemented as specialized SysML block
Logical Environment	• Logical context element which is a representation of the environment the SOI is embedded in. • SAF stereotype is implemented as specialized SysML block
Composition	• Defines the part-of relationship between the System Context (parent) and Logical Context Elements and the Logical SOI (child) • Creates the usage of Logical Context Elements and the Logical SOI in the System Context as Logical Roles
Logical Role	• Usage of Logical Context Elements or the Logical SOI in the scope on a System Context • SAF stereotype is implemented as extension of part property

SAF SFV01c System Context Exchange Viewpoint

This viewpoint documents and aides the definition of the connectivity and exchange between Logical Roles (which represent Logical Context Elements) within a System Context.

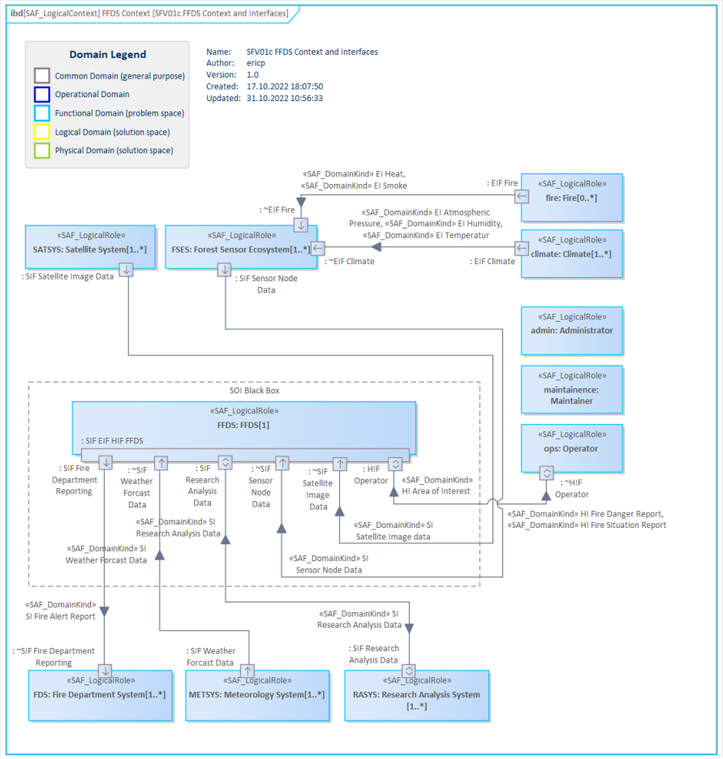
The exchanges can be anything (such as: information, material, energy…) and are documented with a direction.

The System Context Exchange Viewpoint shows which Logical Roles interact with which other Logical Roles and what they exchange, but it does not provide details on how they interact.

General Recommendations and Known Pitfalls:

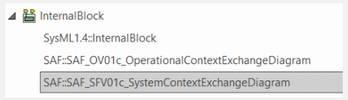
1. If interfaces are required by technical necessity of the System of Interest, but were not required by a Stakeholder, they should still be added to the System Context Exchange Viewpoint. Interfaces can also be derived in the design phase.
2. For interfaces which are derived and were not required initially by a Stakeholder, assign a rationale to the according Proxy Port.
3. For definition of (external) interfaces create a separate bdd for InterfaceBlocks, ProxyPorts and FlowProperties.

Example

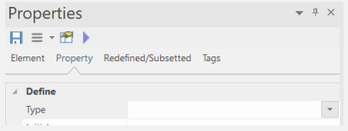


Workflow:

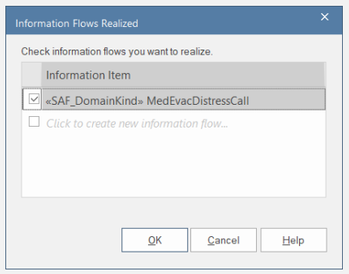
1. Select a System Context and add a new System Context Exchange Diagram as a specialized SysML1.5 Internal Block Diagram with Add Diagram > SysML 1.5 > InternalBlock > SAF_SFV04a_SystemContextExchangeDiagram.



2. Drag and drop the involved Logical Roles from the System Context onto the System Context Exchange Diagram.
3. Add new required interfaces as Proxy Ports to the Logical Role or select the needed required Proxy Ports from the property's Feature Page > Interaction Points.
4. For new Proxy Ports select or create an Interface Block by selecting the Proxy Port > Properties > Property > Define > Type > Select Type ...



5. If needed add FlowProperties to the InterfaceBlocks to refine the interface or visualize a direction.
6. Create a connector between the Logical Role's Proxy Port of the SOI and the Logical Role of a context element. If required, connect to a Proxy Port of the Logical Role of a context element.
7. If the Logical Roles exchange information, energy, or material establish an Item Flow and select the according Domain Kind.
   Select the Connector > right-click on the Connector > Advanced > Information Flow Realized > Select the Domain Kind(s)



Viewpoint Definition:

The viewpoint depicts the following stereotypes:

Stereotype	Description
Logical Role	• Usage of Logical Context Elements or the Logical SOI in the scope of a System Context • SAF stereotype is implemented as extension of part property
Proxy Port	• Logical interface of the Logical SOI or a given/required interface of a System Context Element.
Interface Block	• Type definition of proxy ports to strcuture the logical interface.
ItemFlow	• Defines the flow of a Domain Kind between two Logical Roles
Connector	• Hosts one or more Item Flows between two Logical Roles • A connector can host multiple Item Flows. It is highly recommended only to host Item Flows with the same direction.
Rational	• For derived interfaces which were not required initially.

Viewpoint Input:

- **SAF SFV01b System Context**:
  The System Context defines the scop in which the Logical Roles interact.
- **SAF SFV02a System Domain Kind:**
  The System Domain Kinds define what is being exchanged between the differen Logical Roles within one or more System Context(s).
- **SAF SFV01a System Story:**
  The System Stories provide input on the exchanges between the differen Logical Roles involved in a System Context.
- **SAF Logical Role:**
  The Logical Roles have to be defined in order to define the exchanges between them.
- **SAF SOV06a Stakeholder Requirements:**
  Stakeholders have requirements relevant for elements defined in this viewpoint.

Viewpoint Output:

- **SAF SFV03a System Process:**
  Refines the exchanges between Logical Roles within given processes.
- **SAF SFV04a System Context Interaction:**
  Refines the communication between Logical Roles within given processes.
- **SAF SFV02a System Domain Kind:**
  Defines System Domain Kinds when there is none yet existing for an Item Flow between two Operational Roles identified in this viewpoint.

SAF SFV02a System Domain Item Kind Viewpoint

This viewpoint provides an overview of all relevant and needed exchange types between the SOI and System Context Elements.

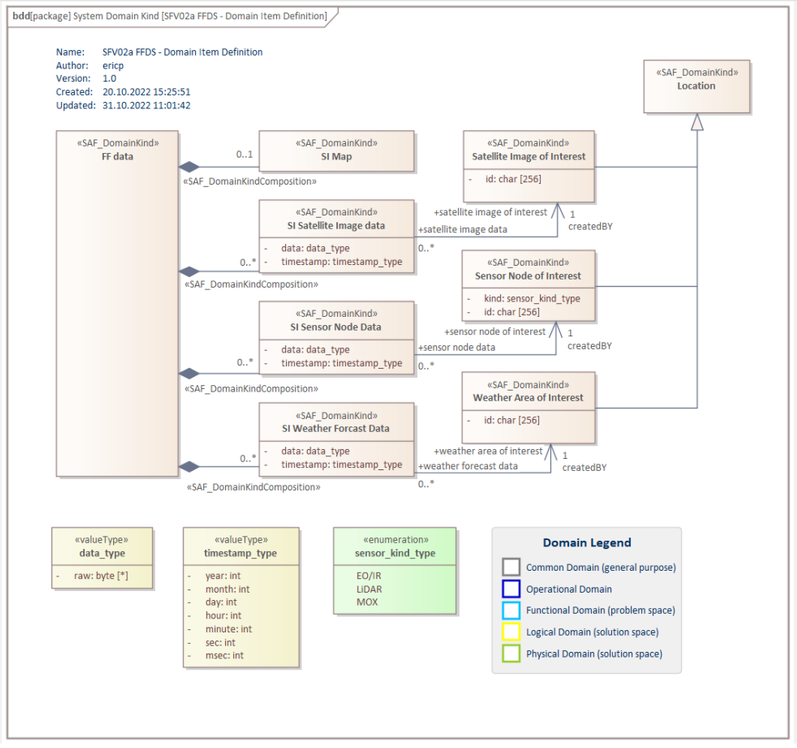
The System Domain Kinds type item flows and thus describe what is being exchanged between Logical Roles in the Context Exchange Viewpoint.

Different System Domain Kinds can be related to each other, which is documented in this viewpoint, but other than that no relations are documented here. If information on System Domain Kinds is needed to understand them, it should be documented in this viewpoint.

General Recommendations and Known Pitfalls:

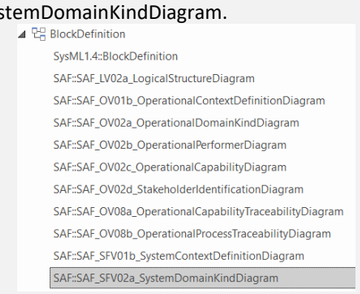
1. It is strongly recommended to add information to each System Domain Kind for a better understanding and in order to avoid creating the same System Domain Kind twice with different names. This information should be added as rationales in this viewpoint.
2. If System Domain Kinds were created in other viewpoints, they will not be added to this viewpoint automatically. They have to be added manually to keep this viewpoint up to date.
3. While there is no technical limitation to the number of System Domain Item Kind viewpoints per project, it is strongly recommended to only have one per project.

Example



Workflow:

1. Create a new System Domain Kind diagram as specialized SysML1.5 Block Definition Diagram with Add Diagram > SysML 1.5 > BlockDefinition > SAF_SFV02a_SystemDomainKindDiagram.



2. Create new Domain Kind which is required e.g., to describe an exchange of information, energy, or material between Logical Roles or to type a flow property.
3. Drag and Drop Operational Domain Kind(s) onto the diagram and establish a relationship with the Domain Kind using the Domain Kind Derivation stereotype.
4. If needed, you can formalize the Domain Kind attributes with value types or enumerations.
5. If needed, you can establish a Domain Kind Composition, an association, or a generalization relationship between two Domain Kinds. Please note that the composition must be set from parent to child.

Viewpoint Definition:

The viewpoint depicts the following stereotypes:

Stereotype	Description
Domain Kind	• Represents material, energy or information relevant for the SOI • Using value types and enumerations it can include a quantative statement with values and units • SAF stereotype is implemented as specialized SysML block
Domain Kind Composition	• Defines the part-of relationship between two Dopmain Kinds if needed
Operational Domain Kind	• Represents material, energy or information specific for the Operational Context • SAF stereotype is implemented as specialized SysML block
Domain Kind Derivation	• Relationship between the derived Domain Kind and Operational Domain Kind • SAF stereotype is implemented as an extension of UML abstraction
Value Type	• Definition for value properties
Enumeration	• Definition for enumeration based types
Rational	• Explains the reason why the Domain Kind is needed and lists the essential details of the Operational Domain Kind

Viewpoint Input:

- **SAF SFV01a System Story**:
  The System Story descriptions provide information
- **SAF SFV01c System Context Exchange:**
  The System Context Exchange(s) identify flows between Logical Roles that are typed by System Domain Kinds.
- **SAF SOV06a Stakeholder Requirements:**
  Stakeholders can contain specific requirements relevant for the elements defined in this viewpoint.

Viewpoint Output:

- **SAF SFV01c System Context Exchange:**
  System Domain Kinds are used to type Item Flows between Logical Roles.
- **SAF SFV03a System Process:**
  System Domain Kinds can be used to type action pins which refine exchanges between Function Actions.

SAF SFV02b System Capability Viewpoint

This viewpoint documents and aides the definition of the SOI's System Capabilities which are contributing to overlying Operational Capabilities.

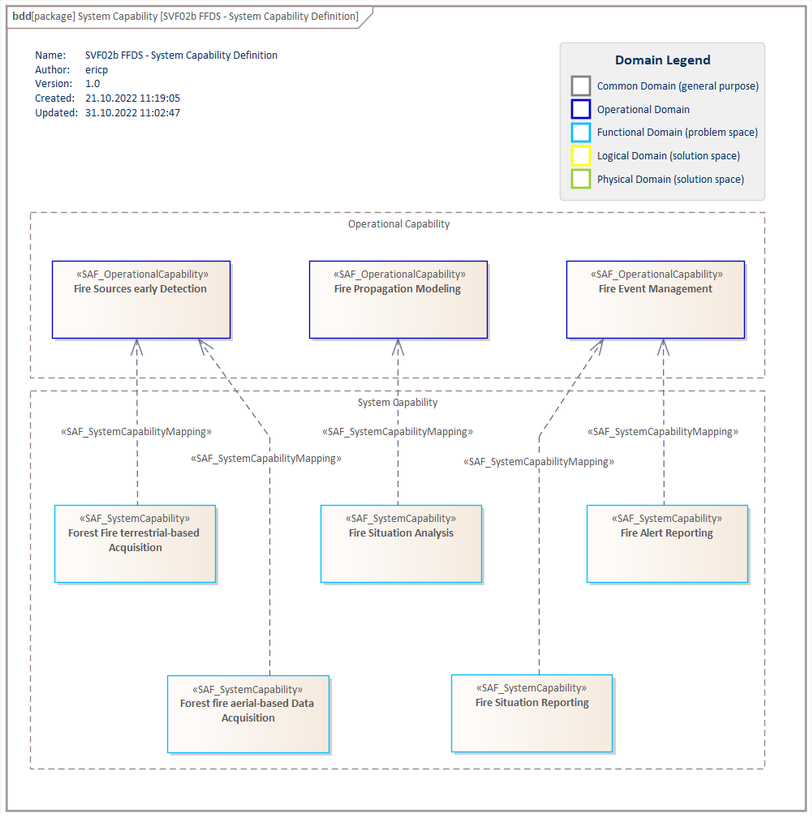
These are only created in this viewpoint and are derived from Operational Capabilities (see SAF SOV02c Operational Capability Viewpoint) and can be the origin for System Functions (SAF SFV02c System Functional Breakdown Viewpoint). These traces are also documented in this viewpoint.

System Capabilities can be related to each other, which is also documented in this viewpoint.

General Recommendations and Known Pitfalls:

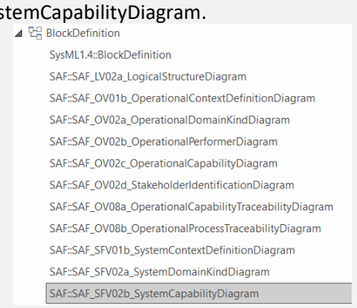
1. System Capabilities should always be documented in combination with Operational Capabilities.
2. When deriving System Capabilities from Operational Capabilities, it has to be ensured that the System Capabilities fit to the System of Interest definition. Also, System Capabilities should fit to the defined System Stories. Both these relations are not documented in any viewpoint but have to be respected to ensure a consistent definition of the System of Interest.

Example



Workflow:

1. Create a new System Capability Diagram in the dedicated package in your model as specialized SysML1.5 Block Definition Diagram with Add Diagram > SysML 1.5 > BlockDefinition > SAF_SFV02b_SystemCapabilityDiagram.



2. Drag and drop Operational Capabilities onto the diagram.
3. Derive and add new System Capabilities on the diagram.
4. Establish a relationship between Operational Capabilities and System Capabilities using the System Capability Mapping stereotype.
5. If needed define System Capability Composition, Generalization and Dependencies between System Capabilities. Please note that the composition must be set from parent to child.

Viewpoint Definition:

The viewpoint depicts the following stereotypes:

Stereotype	Description
System Capability	• SOI's ability to perform a planned action. • Can only be created in SFV02b • SAF stereotype is implemented as specialized SysML block
System Capability Composition	• Relationship to (de-)compose a System Capability into detailed System Capabilities, if needed • SAF stereotype is implemented as extension of EA composition
System Capability Generalization	• Inheritence relationship to generalize/specialize an System Capability, if needed • SAF stereotype is implemented as extension of UML generalization
System Capability Dependency	• Loose relationship between two System Capabilities which defines a dependency between them. • SAF stereotype is implemented as extension of UML dependency
Operational Capability	• Ability to perform a planned operational action. • SAF stereotype is implemented as specialized SysML block
System Capability Mapping	• Relationship to document the contribution of a System Capabilities and an Operational Capability • SAF stereotype is implemented as an extension of UML dependency

Viewpoint Input:

- **SAF SFV01a System Story**:
  System Capabilities perform a planned action within a System Story.
- **SAF SOV02c Operational Capabilities**:
  System Capabilities contribute to Operational Capabilities.
- **SAF SOV06a Stakeholder Requirements:**
  Stakeholders can contain specific requirements relevant for the elements defined in this viewpoint.

Viewpoint Output:

- **SAF SFV06a System Requirement:**
  System Requirements refine System Capabilities.
- **SAF SFV03a System Process:**
  System Processes refine System Capabilities.

SAF SFV02c System Functional Breakdown Viewpoint

This viewpoint documents and aides the decomposition of System Processes into required System Functions which the System of Interest has to do in order to perform the System Process.

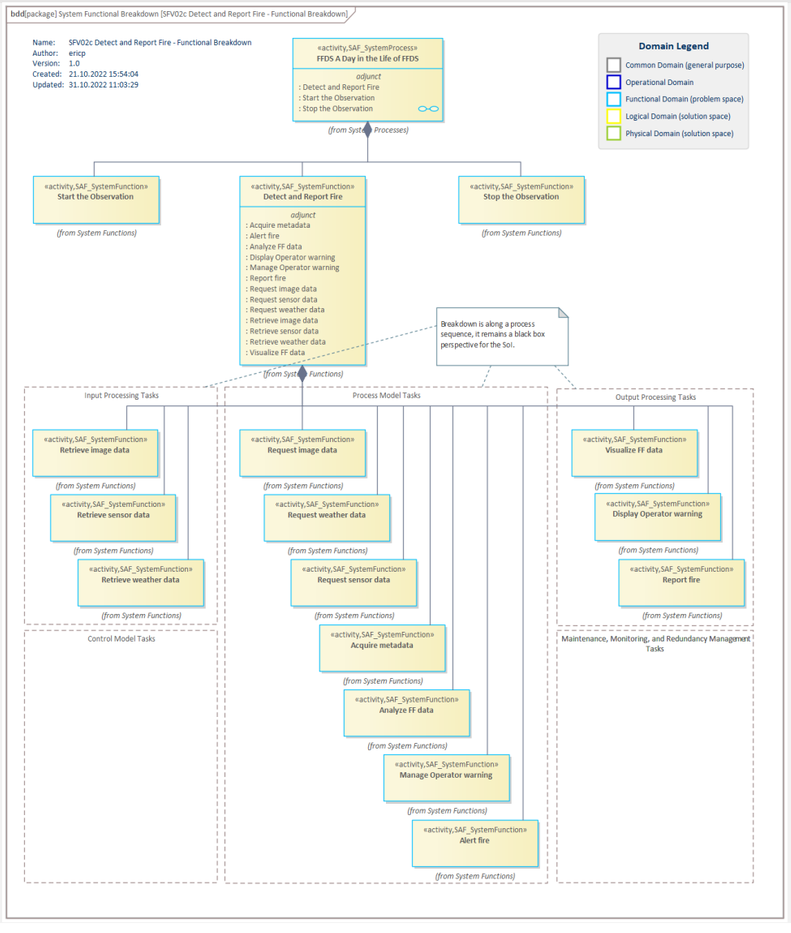
In this viewpoint no interactions or interfaces between the System Functions themselves or between them and Context Elements are defined. However, a good understanding of the system processes that are broken down and their embedment in their respective context is required, to ensure a consistency between the System Functions and the System Process.

While the System of Interest is broken down functionally here, it is still a Blackbox view on it and does not document, how the System Functions are to be realized.

General Recommendations and Known Pitfalls:

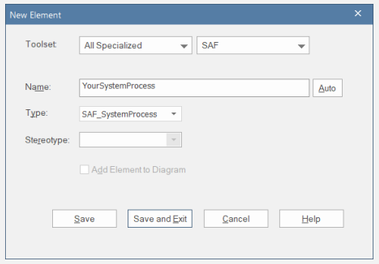
1. If boundaries are given for the design of the System of Interest (e.g., certain system components are pre-determined by a stakeholder), these should be respected in the definition of the System Functions. While this viewpoint is still a Blackbox viewpoint, it is the basis for the further functional breakdown in the Whitebox design.
2. It is important to ensure a consistency between this viewpoint and SAF SFV03a System Process Viewpoint. Therefore, the Functional Breakdown of each System Process should be validated against its Process View, ensuring that the breakdown covers all in- and outputs of the process and the functions required to fulfill its purpose.

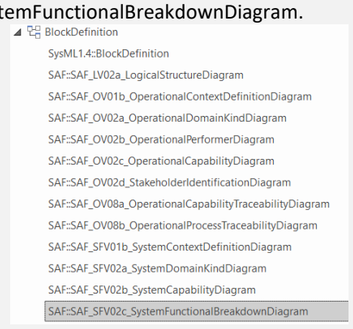
Example



Workflow:

1. Create a System Process item in the dedicated package in your model to formalize a System Story or analyze further the SOI's contribution in an Operational Process.
2. Create a new System Functional Breakdown Diagram in the dedicated package in your model as specialized SysML1.5 Block Definition Diagram with Add Diagram > SysML 1.5 > BlockDefinition > SAF_SFV02c_SystemFunctionalBreakdownDiagram.





3. Drag and drop an existing System Process onto the diagram.
4. Define and create System Function(s) which are needed to perform the System Process.
5. Create a composition between the System Process and System Function. Please note that the composition must be set from System Process as parent to System Function as child.

Viewpoint Definition:

The viewpoint depicts the following stereotypes:

Stereotype	Description
System Process	• Representation of a scenario in the System Context, which involves the SOI and its system context elements • SAF stereotype is implemented as specialized SysML activity
System Function	• Single black-box function • SAF stereotype is implemented as specialized SysML activity
Composition	• Relationship to (de-)compose a System Process into detailed black-box System Functions

Viewpoint Input:

- **SAF SFV03a System Process**:
  System Processes are defined that are broken down into System Functions.
- **SAF SOV06a Stakeholder Requirements:**
  Stakeholders can contain specific requirements relevant for the elements defined in this viewpoint.

Viewpoint Output:

- **SAF SFV03a System Process:**
  Identified System Functions are used in System Processes by Function Actions.
- **SAF SFV06a System Requirements:**
  System Functions are refined by System Requirements.

SAF SFV03a System Process Viewpoint

This viewpoint documents and aides the definition of behavioral System Processes and how they are embedded in their respective System Contexts. This is documented in the form of an Activity Diagram, where the contributing Context Elements are represented as Logical Roles.

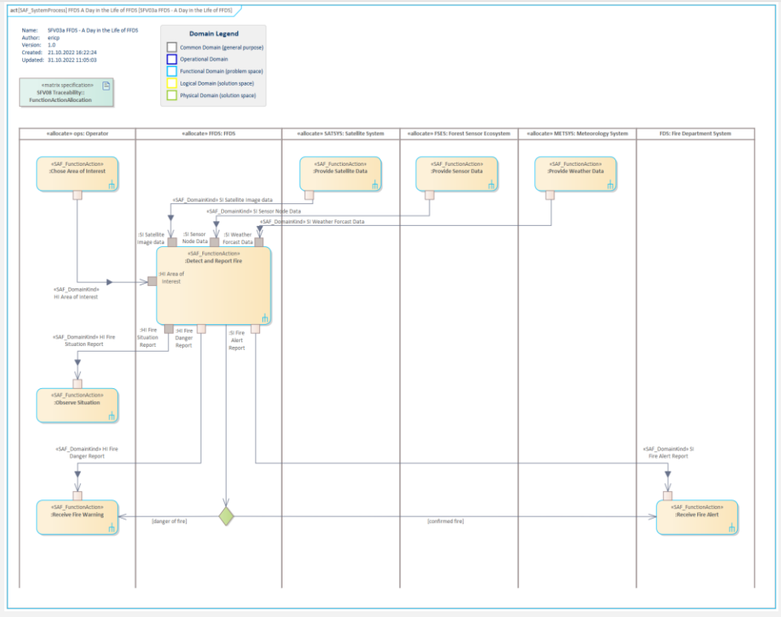
The main focus of this viewpoint is to document how the System of Interest shall interact with its System Context. Thus, this viewpoint aides the SAF SFV02c System Functional Breakdown Viewpoint and also the definition of requirements for the System of Interest towards its Context and the Context Elements.

Furthermore, the viewpoint might aide to analyze the required integrity, reliability, and latencies of the System Process and required System Functions.

General Recommendations and Known Pitfalls:

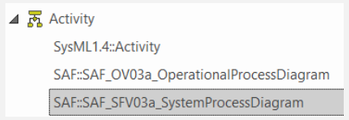
1. If design constraints are given for the System of Interest, its interfaces, or Context Elements, it is important to respect these in this viewpoint, even if the details are not visible directly.
2. It is important to ensure a consistency between this viewpoint and SAF SFV02c System Functional Breakdown Viewpoint. Therefore, the System Process Viewpoint should be validated against its Functional breakdown, ensuring that the breakdown covers all in- and outputs of the process and the functions required to fulfill its purpose.
3. Neither for the System of Interest nor for the System Context Roles should internal functionalities be described.
4. As the Enterprise Architect does not allocate the Function Actions automatically when they are moved into a partition, it is very important to maintain the allocation matrix in SAF SFV08a System Functional Black Box Allocation Viewpoint.

Example

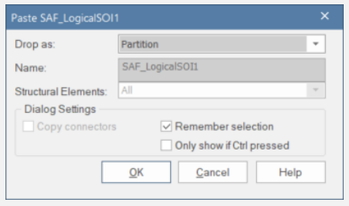


Workflow:

1. Create a new SAF System Process diagram below a System Process in its dedicated package - right-click on the System Process, select New Child Diagram > Add Diagram > SysML 1.5 > Activity > SAF_SFV03a_SystemProcessDiagram



2. Drag and drop an involved Logical SOI or Logical Context Elements onto the diagram and create a Logical Role by selecting partition in the dialog. Name the Logical Role according to the Logical Role in the System Context.
3. Add Function Actions to the according AllocateActivity-Partition.



4. Set the behavior classifier to the created System Function by right-click on the action > Advanced > Set Behavior Classifier.
5. If no corresponding System Function is available, create a new element in SAF SFV02c System Functional Breakdown Viewpoint). Proceed with step 4.
6. Model control and object flows by using the Quick Linker. For object flows you can set the type of the action pins to a Domain Kind in the pin properties.
7. Establish and maintain an allocation between System Function Actions and System Context's Logical Roles in SAF SFV08a System Functional Black Box Allocation Viewpoint.

Viewpoint Definition:

The viewpoint depicts the following stereotypes:

Stereotype	Description
Logical Role	• Usage of the Logical SOI or Logical Context Element in the scope of an Operational Context • Should be created in SFV01b • SAF stereotype is implemented as extension of part property
Function Action	• Usage of System Function in the scope of a System Context • SAF stereotype is implemented as extension of UML Call Behavior Actions
Trigger	• Indication that an event is taken place • Might cause a Transition in a state machine
Allocate Activity Partition	• Partition-wise representation of a Logical Role • Any Function Action located in the Allocate Activity Partition must have an allocate relationship to the Logical Role (allocate by usage pattern) • However, please note that EA does not set the allocate relationship automatically
ItemFlow	• Defines the flow of a Domain Kind between two Function Actions
Matrix Specification	• Matrix-oriented traceability representation to set and maintain allocate relationship between Function Actions and Logical Role (allocate by usage pattern)

For other notation elements which are exposed in the viewpoint, consider UML/SysML standard reference literature.

Viewpoint Input:

- **SAF SFV02c System Functional Breakdown:**
  System Functions broken down from Processes are used as Function Actions.
- **SAF SOV03a Operational Process:**
  Operational Processes are refined by System Processes
- **SAF SFV01b System Context Definition:**
  Defines Logical Context Elements that are represented by Logical Roles in System Processes.
- **SAF SFV01a System Story:**
  System Stories are refined by System Processes. Also, Logical Context Elements are defined which are represented by Logical Roles in System Processes.
- **SAF SOV06a Stakeholder Requirements:**
  Stakeholders can contain specific requirements relevant for the elements defined in this viewpoint.

Viewpoint Output:

- **SAF SFV02c System Functional Breakdown:**
  Function Actions need defined System Functions in the Functional Breakdown.
- **SAF SFV04a System Context Interaction:**
  Messages and time critical behavior between SOI and Logical Context Elements can be refined by System Context Interactions.
- **SAF SFV06a System Requirement:**
  Function Actions are refined by System Requirements.
- **SAF SFV03b System State:**
  System Processes influence System State definitions and can trigger Transitions between System States.
- **SAF External:**
  System Processes can require certain behavior by Logical Context Elements that should be aligned with their respective development teams.

SAF SFV03b System State Viewpoint

This viewpoint documents and aides the definition of States and Modes for the System of Interest from a Blackbox perspective and/or System Domain Kinds.

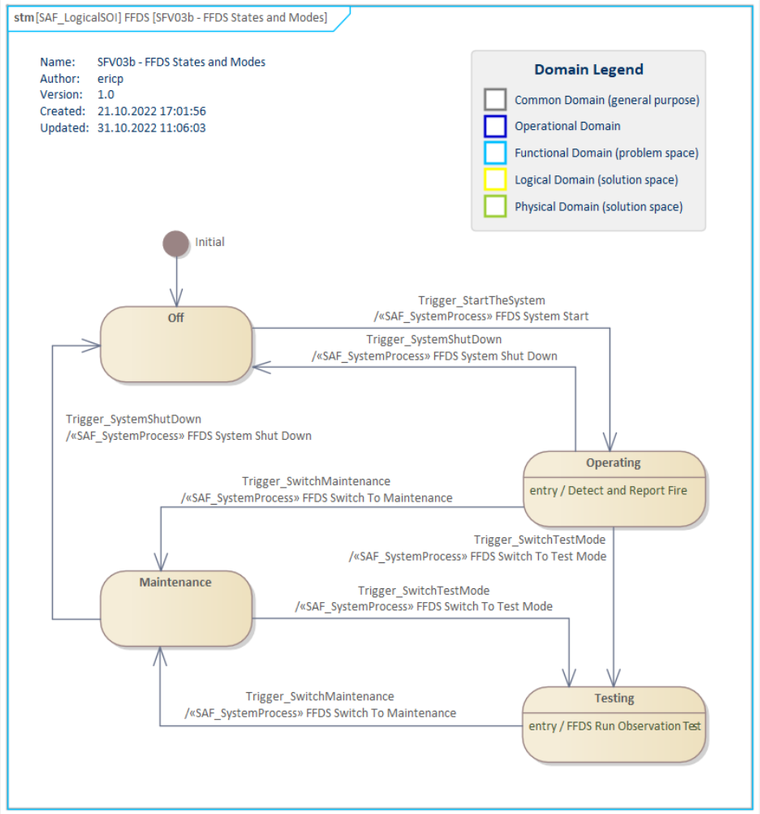
This means, that it shows system states and transitions of the System of Interest that can be observed from the outside. The behavior of Context Elements shall not be described here.

Optionally, System Functions can be linked to a System State Machine by setting them as entry, execution, or exit behavior for states.

General Recommendations and Known Pitfalls:

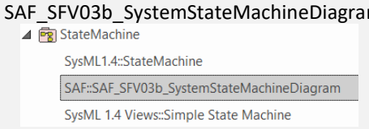
1. While it is possible to define more than one System State Machine for one System of Interest, it has to be thoroughly ensured that these are exclusive and are not running at the same time.
2. System states and transitions that cannot be observed from outside the System of Interest should not appear in the System State Machine.
3. In the problem space it is recommended to model separate reasons for a transition between two states as separate transitions instead of modelling one transition with multiple connected gates.
4. A system's safe state as a result of a safety concept should be considered in this viewpoint.

Example

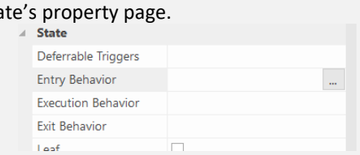


Workflow:

1. Select a structural system element or a Domain Kind in the project browser which you want to describe behaviorally with a state machine.
2. Create a new SAF System State Machine diagram below the element - right-click on the element, Add > Add Diagram > SysML 1.5 > StateMachine > SAF_SFV03b_SystemStateMachineDiagram



3. Describe the element's behavior with states and transitions.
4. If required, set a System Function as entry/execution/exit behavior in states on the state's property page.



Viewpoint Definition:

The viewpoint depicts the following stereotypes:

Stereotype	Description
State	• Represents a situation of the SOI, which can be static or dynamic.
Transition	• Logical movement between States
System Process or System Function	• Defined behavior when entering, remaining or leaving state, or when a transition is perpetrated • SAF stereotype is implemented as specialized SysML activity
Trigger	• Indication that an event is taken place • Can be caused by a System Process or System Function

For other notation elements which are exposed in the viewpoint, consider UML/SysML standard reference literature.

Viewpoint Input:

- **SAF SFV02c System Functional Breakdown**:
  System Functions break down System Processes and can be set as behavior for System States.
- **SAF SFV03a System Process**:
  System Processes can trigger Transitions.
- **SAF SFV04a System Context Interaction:**
  Messages from Logical Context Elements can trigger Transitions.
- **SAF SOV06a Stakeholder Requirements:**
  Stakeholders can contain specific requirements relevant for the elements defined in this viewpoint.

Viewpoint Output:

- **SAF SFV06a System Requirements:**
  System Requirements refine System States and Transitions.
- **SAF SFV03a System Process:**
  Transitions identified in the System State Machine can trigger System Processes.

SAF SFV04a System Context Interaction Viewpoint

This viewpoint documents and aides the definition of required interactions between the System of Interest and its System Context, represented by the Logical Roles.

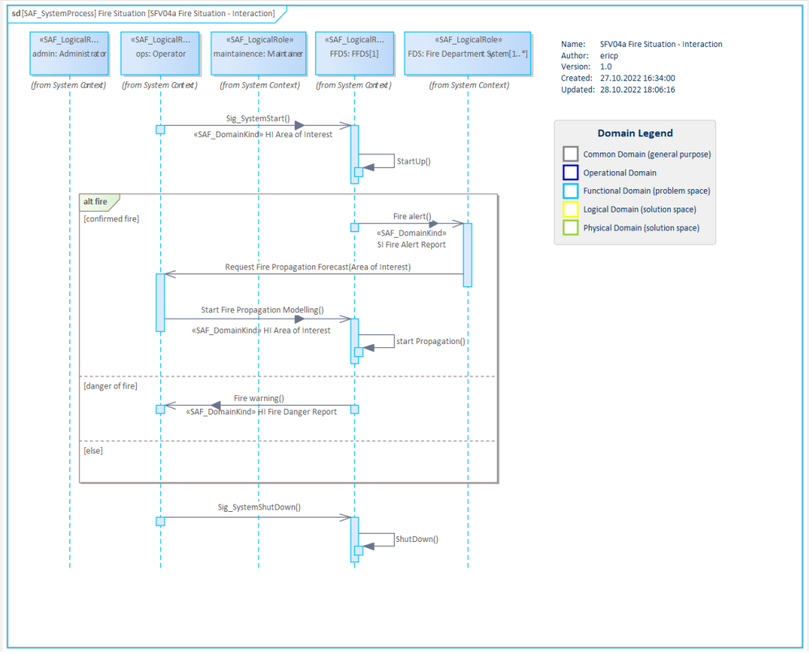
By documenting the interactions between them as a sequence of messages, it is formalizing System Stories and aides the analysis of the System of Interest contribution to Operational Processes. Other than the SAF SFV03a System Process Viewpoint it cannot show internal behavior of the involved Roles, nor can it document object flows between them.

This viewpoint is similar to SAF SOV04a Operational Interaction Viewpoint in that it focuses on the interaction between the Systems but can be more detailed.

General Recommendations and Known Pitfalls:

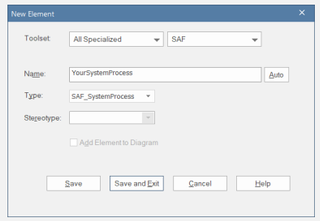
1. While this viewpoint uses other model elements than SAF SOV04a Operational Interaction Viewpoint and should be more detailed (e.g. it is possible to use specific message IDs in this viewpoint, which should be avoided in SAF SOV04a Operational Interaction Viewpoint), it is very easy to essentially model the same content in both viewpoints. This should be avoided.
2. In general, it is recommended to use System Processes to formalize System Stories on this level and only use System Context Interactions when specifically necessary. To show interactions between Systems, it is recommended to use Operational Interactions.
   However, there can be situations where the use of System Context Interactions in addition to System Processes delivers important additional information and should be incorporated into the model (e.g., when there are timing sensitive aspects in the interaction between the System of Interest and other Logical Roles).

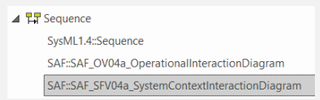
Example

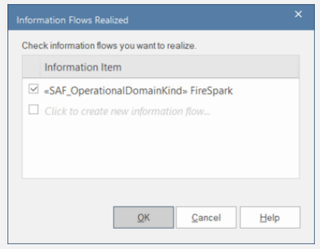


Workflow:

1. Create a System Process item in the dedicated package in your model to formalize a System Story or analyze further the SOI's contribution in an Operational Process.
2. Create a new SAF System Context Interaction diagram for the new System Process - right-click on the System Process, select Add > Add Diagram > SysML 1.5 > Sequence > SAF_SFV04a_SystemContext-InteractionDiagram
3. Drag and drop Logical Roles onto the diagram which are involved in the interaction.
4. Define the sequence in the interaction with messages.
5. Select the message > right-click on the Connector > Advanced > Information Flow Realized > Select the available Operational Domain Kind(s).







Viewpoint Definition:

The viewpoint depicts the following stereotypes:

Stereotype	Description
Logical Role	• Usage of Logical SOI or Logical Context Elements in the scope of a System Context Context • Should be created in SFV01b • SAF stereotype is implemented as extension of part property
Interaction Fragment	• Operator in interaction sequence
Message	• Exchange of material, energy or information between Logical Roles.
ItemFlow	• Defines the flow of a Domain Kind between two Logical Roles

Viewpoint Input:

- **SAF SOV04a Operational Interaction:**
  Operational Interactions are refined by System Context Interactions.
- **SAF SFV01a System Story:**
  System Stories are refined by System Context Interactions.
- **SAF SFV01b System Context Definition:**
  Defines Logical Context Elements which are represented by Logical Roles in System Context Interactions.
- **SAF SFV03a System Process:**
  System Context Interactions can refine communication between Logical Context Elements and the SOI within System Processes.
- **SAF SOV06a Stakeholder Requirements:**
  Stakeholders can contain specific requirements relevant for the elements defined in this viewpoint.

Viewpoint Output:

- **SAF SFV03a System Process:**
  System Context Interactions can influence System Processes.
- **SAF SFV06a System Requirement:**
  System Requirements refine System Context Interactions.
- **SAF External:**
  System Context Interactions can require certain behavior by Logical Context Elements that should be aligned with their respective development teams.

SAF SFV06a System Requirement Viewpoint

This viewpoint documents and aides the definition of textual system requirements for the SOI.

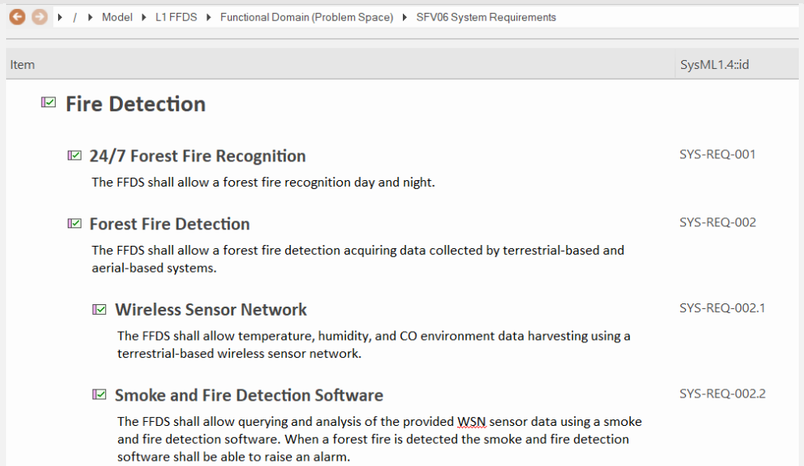
These system requirements can originate from different sources, such as stakeholder requirements or system black box descriptions (system functions, contexts, state machines, processes etc.). Only requirements for the System of Interest are documented in this viewpoint, not requirements for Context Elements.

The traceability of system requirements to its originating sources is also shown in this viewpoint.

General Recommendations and Known Pitfalls:

1. It is highly recommended to carefully analyze the regulatory QM requirements for the project in order to decide if the views in the Functional Domain have to be re-expressed and refined in formal System Requirements in this viewpoint.
2. It is recommended to establish one single System Requirement view for one System of Interest.
3. System requirements can be derived from the Operational Domain and in this case should be linked accordingly. However, they can also be derived from the analysis of the System of Interest (e.g., System Context, System State Machine or System Functional Breakdown) and thus no link to a stakeholder requirement can be created. These requirements should still be linked to their originating element/viewpoint, though.
4. There is no automatic tool support to ensure that any subset of system requirements fully covers the stakeholder requirements or system analysis they are linked to. Thorough reviews are required to maintain this consistency.
5. System requirements should be prioritized. These priorities should be aligned with the SAF SOV06a Stakeholder Requirement Viewpoint priorities.

Example



Workflow:

1. Open the global Specification Manager (and not the Specification View of a diagram).
2. Select the dedicated package for System Requirements.
3. Document or edit your identified requirements with the System Requirements stereotype.

Viewpoint Definition:

The viewpoint depicts the following stereotypes:

Stereotype	Description
System Requirement	• Requirement which applies to the SOI • SAF stereotype is implemented as specialized SysML extendedRequirement
System Functional Requirement	• System Requirement which applies to the SOI and defines a System Function • SAF stereotype is implemented as specialized SAF System Requirement
System Non-Functional Requirement	• System Requirement which applies to quality aspects of the SOI or a SOI's System Function or a given design constraints. • SAF stereotype is implemented as specialized SAF System Requirement
Stakeholder Requirement	• Requirement which is specific for one Stakeholder. • SAF stereotype is implemented as specialized SysML extendedRequirement
System Requirement Derivation	• Relationship between the derived System Requirement and a System Requirement from an abstraction layer above or a Stakeholder Requirement • SAF stereotype is implemented as specialized SysML deriveReqt
System Functional Requirement Constraint	• Relationship between the constrained System Functional Requirement and the System Non Functional Requirement • SAF stereotype is implemented as extension of UML dependency

Viewpoint Input:

- **SAF SFV02b System Capability:**
  System Capabilities are refined by System Requirements.
- **SAF SFV03a System Process:**
  System Processes are refined by System Requirements.
- **SAF SFV04a System Context Interaction:**
  System Context Interactions are refined by System Requirements.
- **SAF SOV06a Stakeholder Requirements:**
  Stakeholder Requirements are refined by System Requirements and need at least one derived System Requirement linked to it when the Stakeholder Requirement is applicable to the SOI.
- **SAF External:**
  Technical necessities can lead to System Requirements that are not strictly derived from Stakeholder Requirements.

Viewpoint Output:

- **SAF Logical and Physical Domain:**
  The System Requirements are input for the solution space of the system definition. In the Logical and Physical Domain the system is broken down into elements that contribute to the System Requirements.
- **System Component Requirements:**
  The Requirements for each System Component contribute to System Requirements and are derived from these and the Logical and Physical Domain Viewpoints.

SAF SFV08a System Functional Black Box Allocation Viewpoint

This viewpoint documents the allocation of Function Actions to the System of Interest and Context Elements, represented by Logical Roles. Thus, it clarifies which Element in the System Context is responsible for implementing a specific System Function.
This viewpoint is important to discuss the functional partitioning on a System of Systems layer with other System Leads.

General Recommendations and Known Pitfalls:

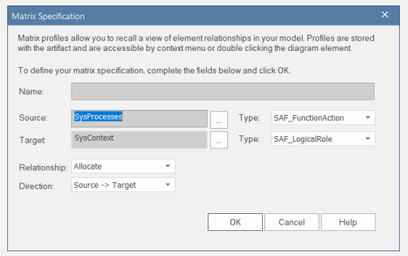
1. The tool Enterprise Architect does not automatically allocate Function Actions to Logical Roles when they are put into a partition in the SAF SFV03a System Process Viewpoint. Therefore, any allocation of a Function Action to a Logical Role has to be maintained in the System Functional Black Box Allocation Viewpoint. It is recommended to link this viewpoint in any System Process diagram, to be able to maintain the allocation matrix quickly and easily when working in the System Process Viewpoint.

Example

Source \ Target	admin	climate	FDS	FFDS	fire	FSES	maintainence	METSYS	ops	RASYS	SATSYS
:Chose Area of Interest									⇑		
:Detect and Report Fire				⇑							
:Observe Situation									⇑		
:Provide Satellite Data											⇑
:Provide Sensor Data						⇑					
:Provide Weather Data								⇑			
:Receive Fire Alert			⇑								
:Receive Fire Warning									⇑		

Workflow:

1. Create (and maintain) a matrix specification to allocate Function Actions to Logical Roles with the following configuration:
2. Maintain allocate links according to the Function Action in System Process's partitions (see SAF SFV03a System Process Viewpoint)



Viewpoint Definition:

The viewpoint depicts the following stereotypes:

Stereotype	Description
Function Action	• Usage of System Function in the scope of a System Context • SAF stereotype is implemented as extension of UML Call Behavior Actions
Logical Role	• Usage of Logical SOI or Logical Context Elements in the scope of a System Context • Should be created in SFV01b • SAF stereotype is implemented as extension of part property
Matrix Specification	• Matrix-oriented traceability representation to set and maintain allocate relationship between Function Actions and Logical Role (allocate by usage pattern)

Viewpoint Input:

- **SAF SFV03a System Process:**
  Function Actions as usages of System Functions are defined in System Processes.
- **SAF SFV01b System Context Definition:**
  Logical Roles as usages of Logical SOI or Logical Context Elements are defined in System Context Definitions.

Viewpoint Output:

- **SAF SFV06a System Requirement:**
  Allocations of Function Actions to the Logical SOI can be refined by System Requirements.
- **SAF External:**
  Allocations of Function Actions to Logical Context Elements should be aligned with their respective development teams.
- **SAF External:**
  Traceability supports the quality control of the model, showing "lose ends" or inconsistencies. Also, it supports impact analyses for change management and assessments.