Python4Capella Simplified Metamodel

Document generated automatically by M2Doc from Capella model

# Tables of Content

**Aucune entrée de table des matières n'a été trouvée.**

# Foreword

**Introduction**

This model defines the Capella simplified metamodel for implementation of Python API.

The overview of the concepts is provided here: [[CDB] Capella Light Metamodel](hlink://_DEN_8IrrEeum3sRHbZTGzQ)

This diagrams provides links towards other diagrams.

Moreover, a simplified metamodel is also defined for:

* add-on PVMT: [[CDB] PVMT](hlink://_wM5H0NJrEeu6iLrOF9u83w)
* add-on Requirement: [[CDB] Requirement](hlink://_xe0CsNJrEeu6iLrOF9u83w)

**Capella version**

The model focus on the definition of Capella metamodel for version 1.4.0 / 1.4.1 / 1.4.2 / 5.0.0 (no difference between those versions).

Metamodel for Capella 1.3.x will be defined later.

**Definition of the API**

To support the definition of the Python API, some Abstract Types have been defined which are then specialized.

They are defined mainly to ease the definition of the API (to avoid repetition of similar definitions on similar types).

They are not necessarily to be retained by end user.

The main Abstract Types defined are:

* [CapellaElement](hlink://7bd88617-cd02-494f-b666-d250ff6417d6)
* [AbstractCapability](hlink://d26dd98c-93da-4303-92b8-187452212c79)
* [Function](hlink://9dff1e7f-c120-4fe8-bcb5-f3669b2a4a06)
* [BehavioralComponent](hlink://a5c9af0a-2b12-4b80-adff-c9307c2ab9e2)
* [Node](hlink://187ed12e-8722-4fef-b5a1-b605bd98840b)

**Excluded elements**

This first version only focuses on the definition of a simplified metamodel for Capella concepts (model elements).

The concept of diagram is defined (and its relation with model elements) in this diagram: [[CDB] Diagram](hlink://_TheWQM57EeurHu2jZJ4leA).

**However, the content of diagram (graphical elements) is not yet defined**.

Moreover, following elements are excluded from this simplified metamodel:

* All **technical elements** like traces which are only represented by a direct link between elements
* **Capella core requirements** (which are deprecated and replaced by the Requirement add-on)
* **Generic Traces**
* Functional Chain and Physical Path **involvement elements** (simplified metamodel defines a direct relation with Functions or NodePC)  
   As a result, it will not be possible to edit Functional Chain and Physical Path definition with the simplified metamodel  
   Moreover, Control Nodes and Sequence Links are excluded from Functional Chain definition as they rely on the involvement elements.
* **Roles in the Operational Analysis**
* **Technical elements in the definition of scenarios** allowing the definition of the fine grain sequence of events

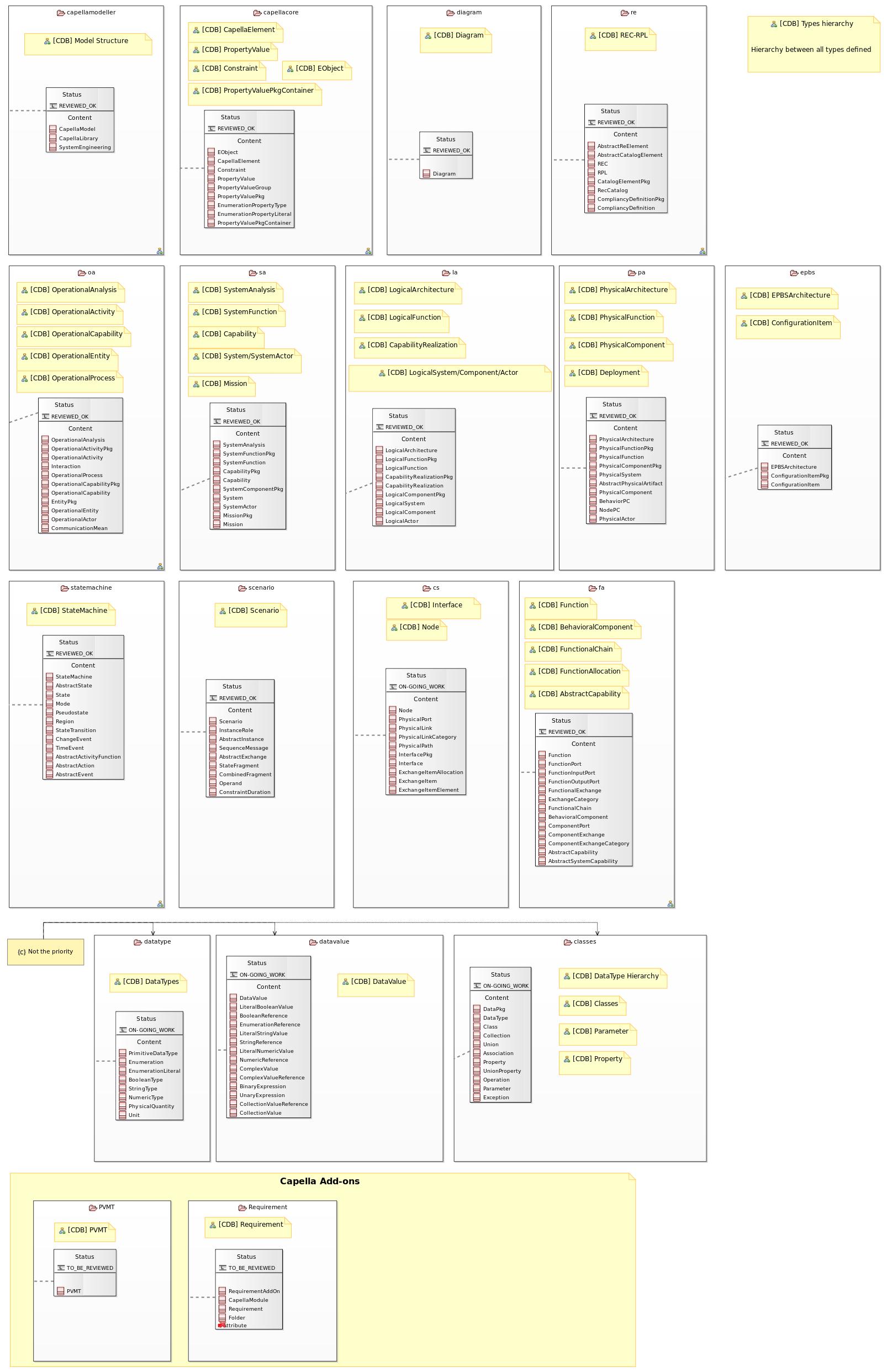
**Main differences with technical metamodel**

In addition to the excluded elements presented before, some elements are defined differently in the model compared to the real Capella technical metamodel.

The main differences are:

* **definition of Abstract Types**: the Abstract Types defined in this model are different from the Abstract Types defined in the technical metamodel. They are introduced in order to simplify the definition of the API. Many levels of Abstract Types are not introduced here. Only the necessary Abstract Types have been introduced to avoid repetition, but Abstract Types have been avoided as much as possible to avoid complexity
* **definition of Packages**: the packages have been redefined in this model and do not strictly follow the same organization as the packages defined in the technical metamodel
* **definition of Actors as a specific type (same for the System)**: in the technical metamodel, actors are defined as a generic Component with a tag isActor. In this simplified metamodel, a specific type as been introduced in order to clearly differentiate what is external to the system and what is not. Moreover, actors in Capella have some specificities: they have both the characteristics of behavioral components and node components at the same time, which is not the case of the component of the system
* **definition of Interaction as a specific type**: in the technical metamodel, Interactions are defined as generic FunctionalExchange. Interactions have been introduced here to clearly separate Functions / Functional Exchanges / Functional Chains from Activities / Interactions / Operation Processes. One of the main reason for this is that Activities do not have ports
* **definition of ComponentExchange source / target**: in the technical metamodel, a ComponentExchange has a source and a target. However, from the user point of view, the orientation of the ComponentExchange is defined from the property orientation on the ComponentPorts. As a result, this simplified metamodel defines only that a ComponentExchange connects two ports without distinguishing a source and a target
* **definition of containment for some elements**: the containment of elements like the Exchanges (FunctionalExchange, ComponentExchange, PhysicalLink) is not defined in this simplified metamodel because it is not relevant for the end user. What is important is that an exchange connects two elements together. Where the exchange is stored should be defined by the tool and not by the end user
* **definition of PropertyValue as one single type**: This is a simplification of the technical metamodel. In the technical metamodel there is: StringPropertyValue, IntegerPropertyValue, BooleanPropertyValue, FloatPropertyValue, EnumerationPropertyValue
* **definition of REC and RPL as two distinct types**: in the technical metamodel, REC and RPL are both defined as CatalogElement with a tag for the type. However, REC and RPL have relations with are specific and do not apply to the other type. As a result, they have been defined as two distinct types here

[CDB] Capella Light Metamodel:



*Note: In the following, attributes and relation followed by [ro] are in Read-Only (cannot be modified by user).*

*Example:*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Ref. name*** | ***Kind*** | ***Card.*** | ***Target Type*** | ***Defined on*** |
| *systemEngineering setter [ro]* | *COMPOSITION* | *[1 1]* | ***SystemEngineering*** | *defined on type* |

Moreover, the status of implementation of the attributes, relations and methods in the Python API is indicated by the following color code:

* Implemented and working
* Implemented but not working
* Not implemented

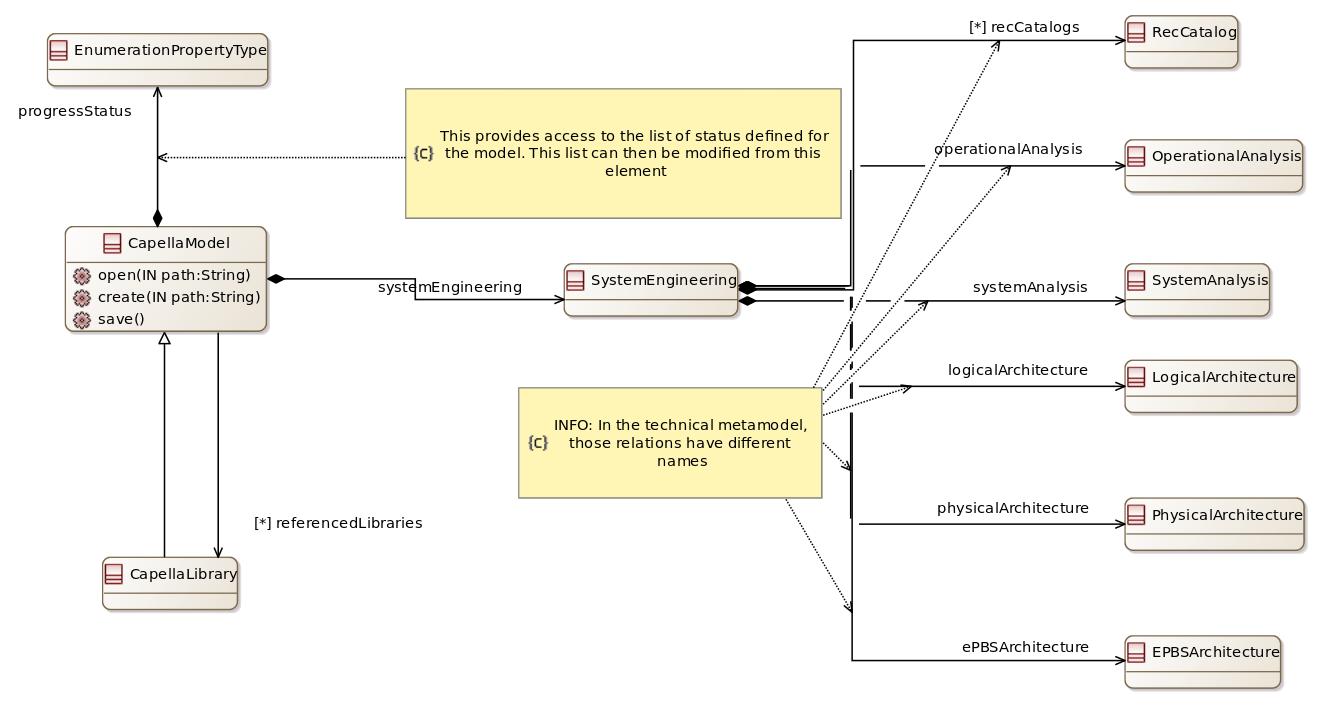
No color code is defined for Abstract types.

# [Package] capellamodeller

Diagrams:

Here are the diagrams of representing the elements introduced in this package:

[CDB] Model Structure:



## [Type] CapellaModel

Description:

A Capella model. Used to defined the content of a model, and how to read a model

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| systemEngineering getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| progressStatus getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| referencedLibraries getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| allDiagrams getter [ro] |  | *defined on type* |

Relations:

*No relation defined*

Methods:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Description** | **IN Param** | **Result** |

|  |  |  |  |
| --- | --- | --- | --- |
| getDiagrams |  | diagramType: | diags: |

|  |  |  |  |
| --- | --- | --- | --- |
| open |  | path: |  |

|  |  |  |  |
| --- | --- | --- | --- |
| create |  | path: |  |

|  |  |  |  |
| --- | --- | --- | --- |
| save |  |  |  |

## [Type] CapellaLibrary

Description:

A Capella Library. A library is a model which can be referenced by other models in order to reuse its content

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

*No attribute defined*

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] SystemEngineering

Description:

The main element in the definition of a Capella model. Contains the perspectives of the Arcadia methodology

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| recCatalogs getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| operationalAnalysis getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| systemAnalysis getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| logicalArchitecture getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| physicalArchitecture getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ePBSArchitecture getter [ro] |  | *defined on type* |

Relations:

*No relation defined*

Methods:

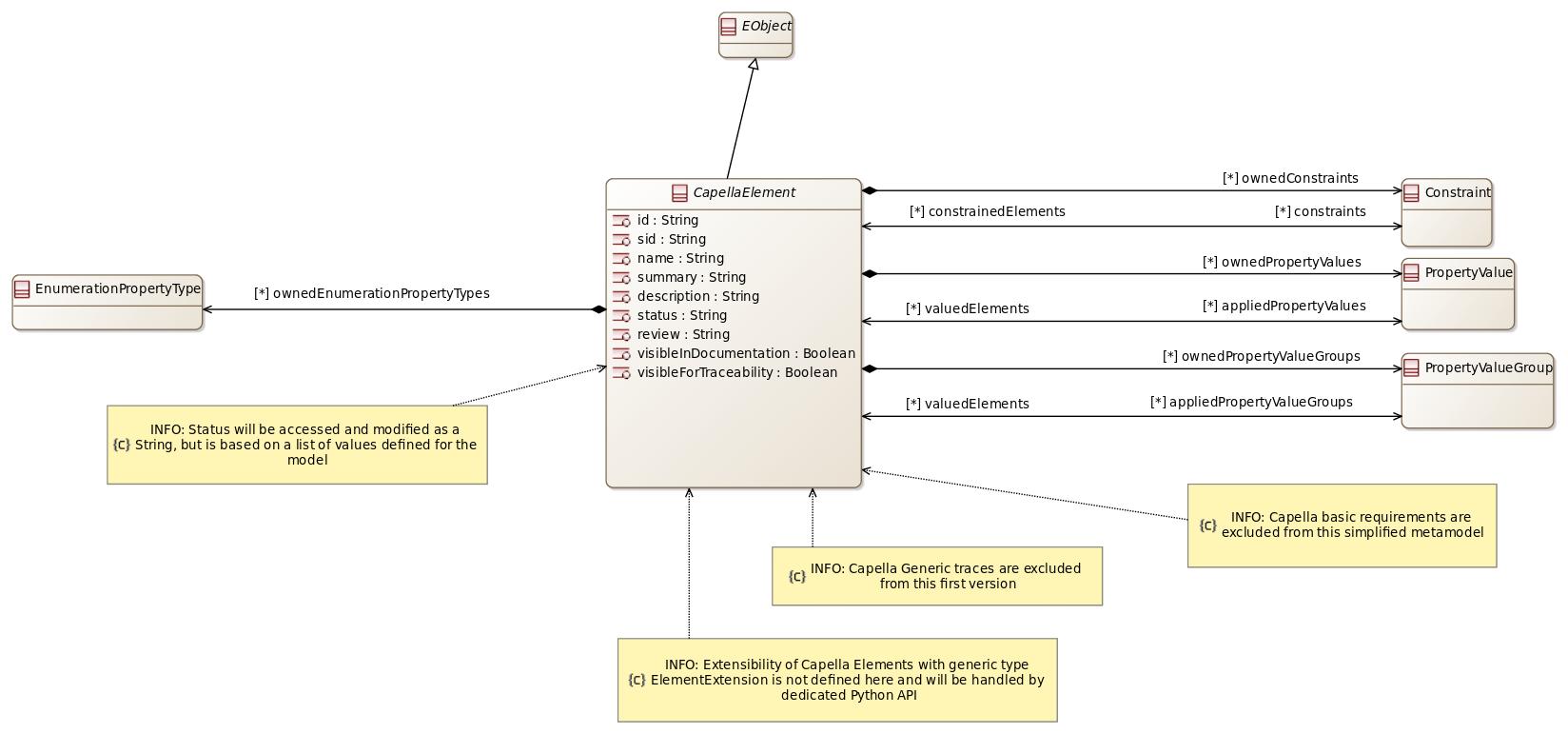
*No method defined*

# [Package] capellacore

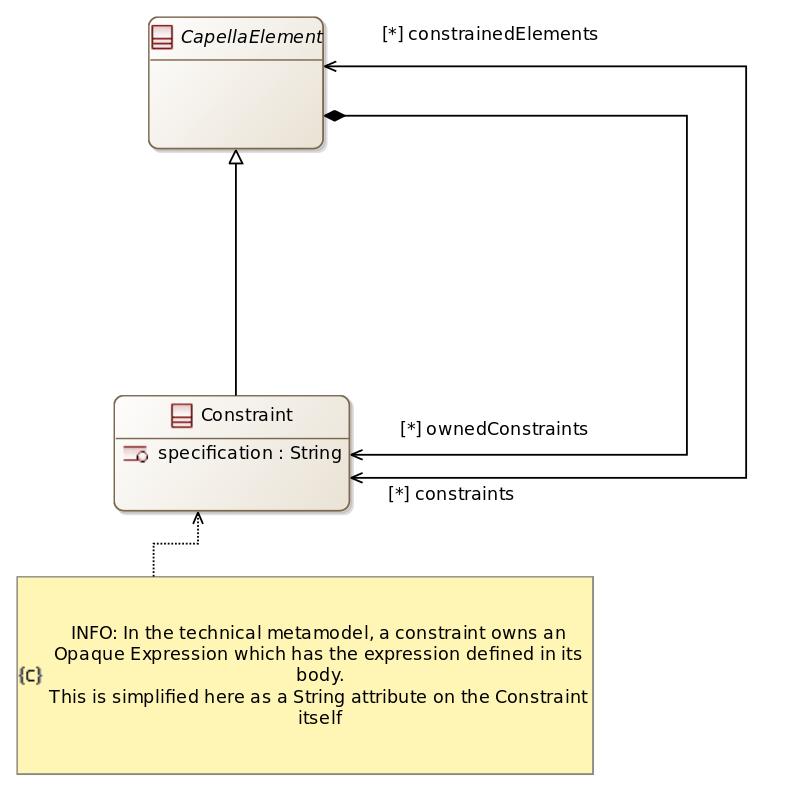
Diagrams:

Here are the diagrams of representing the elements introduced in this package:

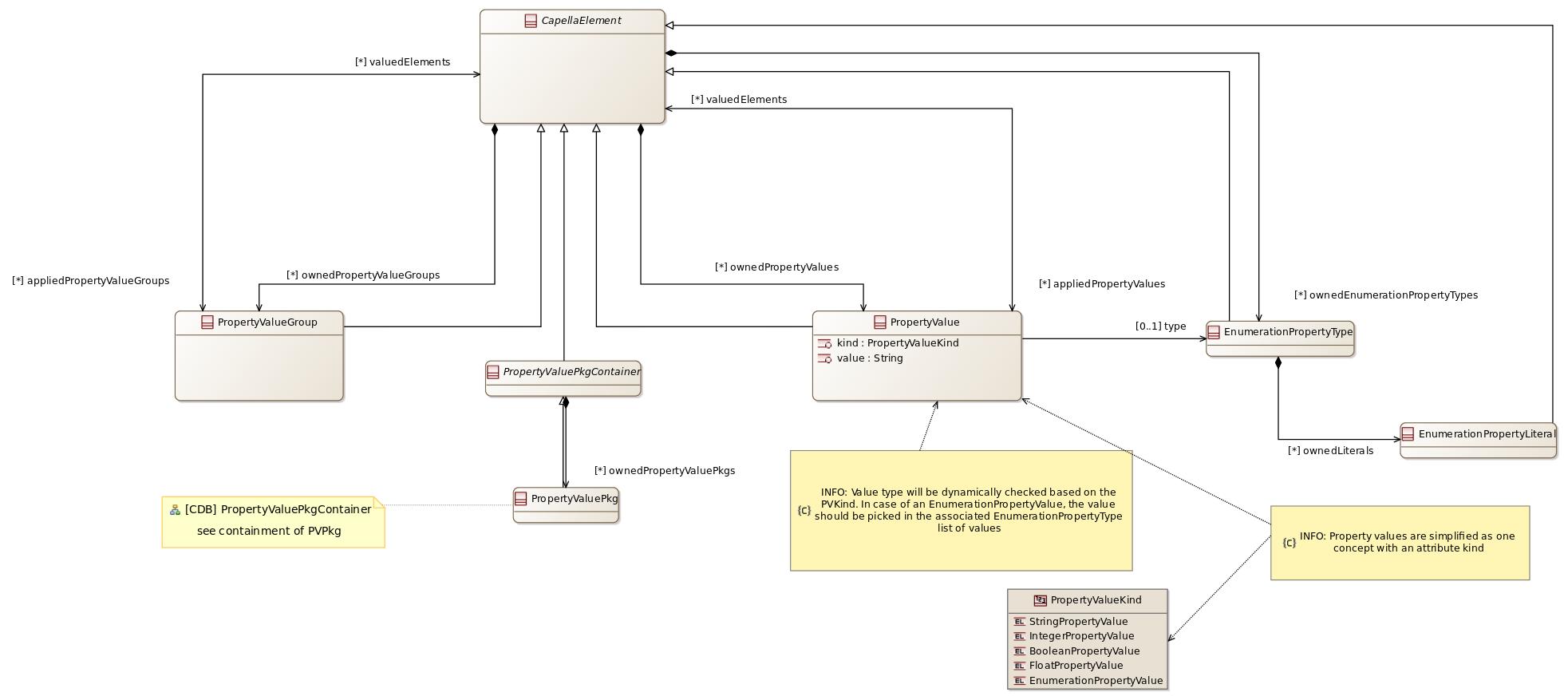
[CDB] CapellaElement:



[CDB] Constraint:



[CDB] PropertyValue:



## [Type] EObject [Abstract]

Description:

A generic object. Defines generic relations which are available for all elements

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| ownedDiagrams |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| elementOfInterestForDiagrams |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| contextualElementForDiagrams |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| representingDiagrams |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| RECs |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| RPLs |  | *defined on type* |

Relations:

*No relation defined*

Methods:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Description** | **IN Param** | **Result** |

|  |  |  |  |
| --- | --- | --- | --- |
| getLabel |  |  | label: |

|  |  |  |  |
| --- | --- | --- | --- |
| getElementType |  |  | elementType: |

|  |  |  |  |
| --- | --- | --- | --- |
| getContainer |  |  | container: |

|  |  |  |  |
| --- | --- | --- | --- |
| getContents |  |  | contents: |

|  |  |  |  |
| --- | --- | --- | --- |
| getAllContents |  |  | contents: |

|  |  |  |  |
| --- | --- | --- | --- |
| getAllContentsByType |  | type: | contents: |

|  |  |  |  |
| --- | --- | --- | --- |
| getAvailableSBQueries |  |  | queryNames: |

|  |  |  |  |
| --- | --- | --- | --- |
| getQueryResult |  | queryName: | result: |

## [Type] CapellaElement [Abstract]

Description:

A generic Capella model Element. Used to define generic attributes and relations inherited by most Capella elements

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| id |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| sid |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| name |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| summary |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| description |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| status |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| review |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| visibleInDocumentation |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| visibleForTraceability |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedConstraints |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| constraints |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedPropertyValues |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| appliedPropertyValues |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedPropertyValueGroups |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| appliedPropertyValueGroups |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedEnumerationPropertyTypes |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] Constraint

Description:

A generic constraint which can be defined on Capella elements

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| specification getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| constrainedElements getter/setter |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] PropertyValue

Description:

A generic property with a name and value which can be added to Capella elements

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| kind getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| value getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| valuedElements getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| type getter [ro] |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] PropertyValueGroup

Description:

A group which can contain several PropertyValue and be applied to CapellaElements

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| valuedElements getter/setter |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] PropertyValuePkg

Description:

A package to contain PropertyValues and/or PropertyValueGroups

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

*No attribute defined*

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] EnumerationPropertyType

Description:

The definition of an Enumeration to type a PropertyValue

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| ownedLiterals getter [ro] |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] EnumerationPropertyLiteral

Description:

A value defined in an EnumerationPropertyType

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

*No attribute defined*

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] PropertyValuePkgContainer [Abstract]

Description:

An abstract type to define all elements which can contain PropertyValuePkg

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| ownedPropertyValuePkgs |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Enumeration] PropertyValueKind

List of values:

* StringPropertyValue
* IntegerPropertyValue
* BooleanPropertyValue
* FloatPropertyValue
* EnumerationPropertyValue

Used by:

## [Enumeration] VisibilityKind

List of values:

* UNSET
* PUBLIC
* PROTECTED
* PRIVATE
* PACKAGE

Used by:

# [Package] diagram

Diagrams:

*No diagram defined for this package*

## [Type] Diagram

Description:

A generic Capella diagram

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| uid getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| name getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| type getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| package getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| description getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| status getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| review getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| visibleInDocumentation getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| visibleForTraceability getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| synchronized getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| target getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| representedElements getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| contextualElements getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| elementsOfInterest getter/setter |  | *defined on type* |

Relations:

*No relation defined*

Methods:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Description** | **IN Param** | **Result** |

|  |  |  |  |
| --- | --- | --- | --- |
| exportAsImage |  | filePath:  format: |  |

## [Enumeration] ExportFormat

List of values:

* JPG
* PNG
* SVG
* BMP
* GIF

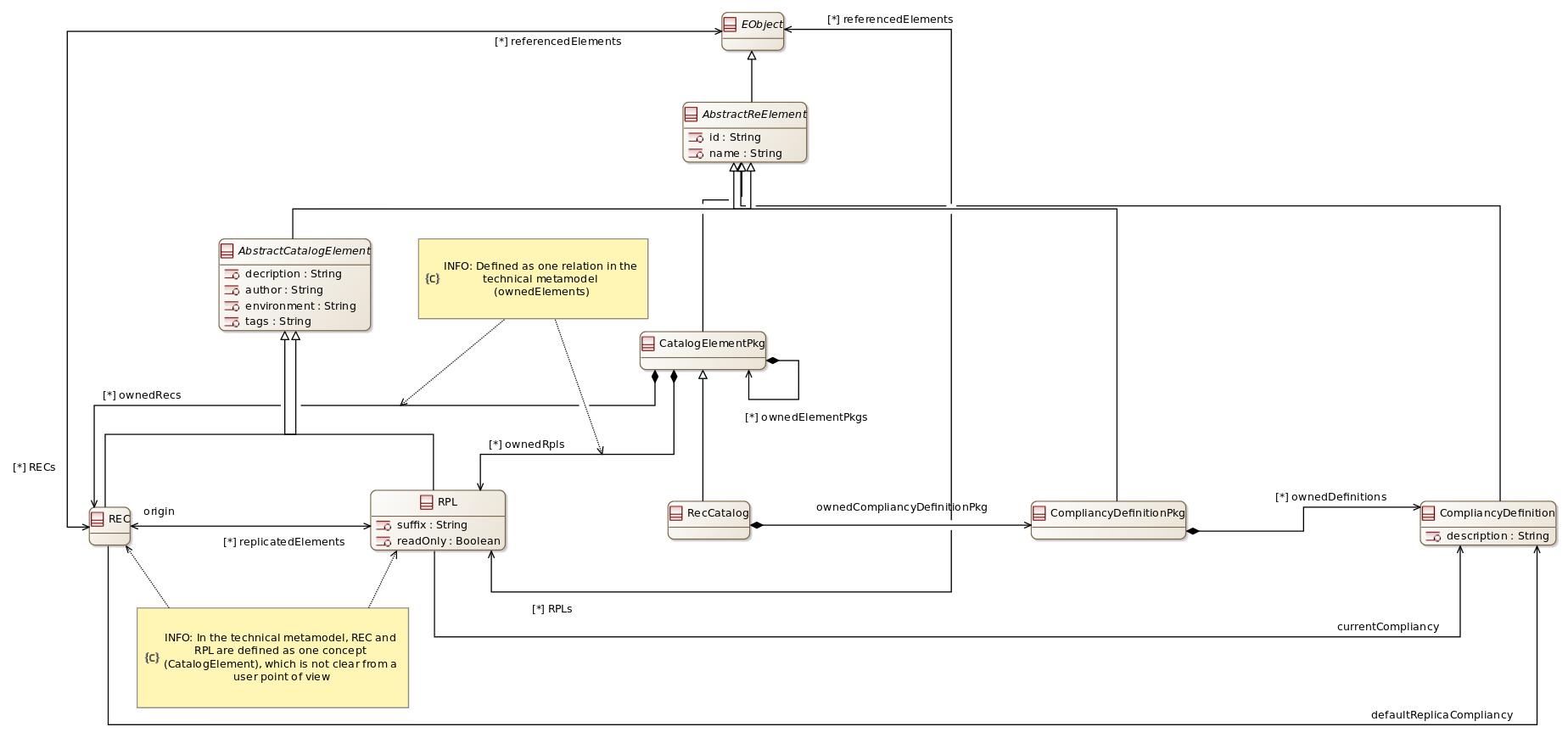
Used by:

# [Package] re

Diagrams:

Here are the diagrams of representing the elements introduced in this package:

[CDB] REC-RPL:



## [Type] AbstractReElement [Abstract]

Description:

An abstract type to define the generic attributes of REC / RPL and packages elements

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| id |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| name |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] AbstractCatalogElement [Abstract]

Description:

An abstract type to define the generic attributes and relations of REC / RPL elements

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| decription |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| author |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| environment |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| tags |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] REC

Description:

A Record element is the definition of a set of elements to be replicated together

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| referencedElements getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| defaultReplicaCompliancy getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| replicatedElements getter/setter |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] RPL

Description:

A replay element is the instantiation of a record element

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| referencedElements getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| suffix getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| readOnly getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| origin getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| currentCompliancy getter/setter |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] CatalogElementPkg

Description:

A package to structure the definition of REC / RPL elements

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| ownedElementPkgs getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedRecs getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedRpls getter [ro] |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] RecCatalog

Description:

The root package which contains the REC / RPL definitions

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| ownedCompliancyDefinitionPkg getter [ro] |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] CompliancyDefinitionPkg

Description:

A package to contain CompliancyDefinitions

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| ownedDefinitions getter [ro] |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] CompliancyDefinition

Description:

The type of compliancy which have to be respected by the RPL regarding its REC definition. Default list of compliancies is:

* BLACK\_BOX
* CONSTRAINT\_REUSE
* INHERITANCY\_REUSE

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| description getter/setter |  | *defined on type* |

Relations:

*No relation defined*

Methods:

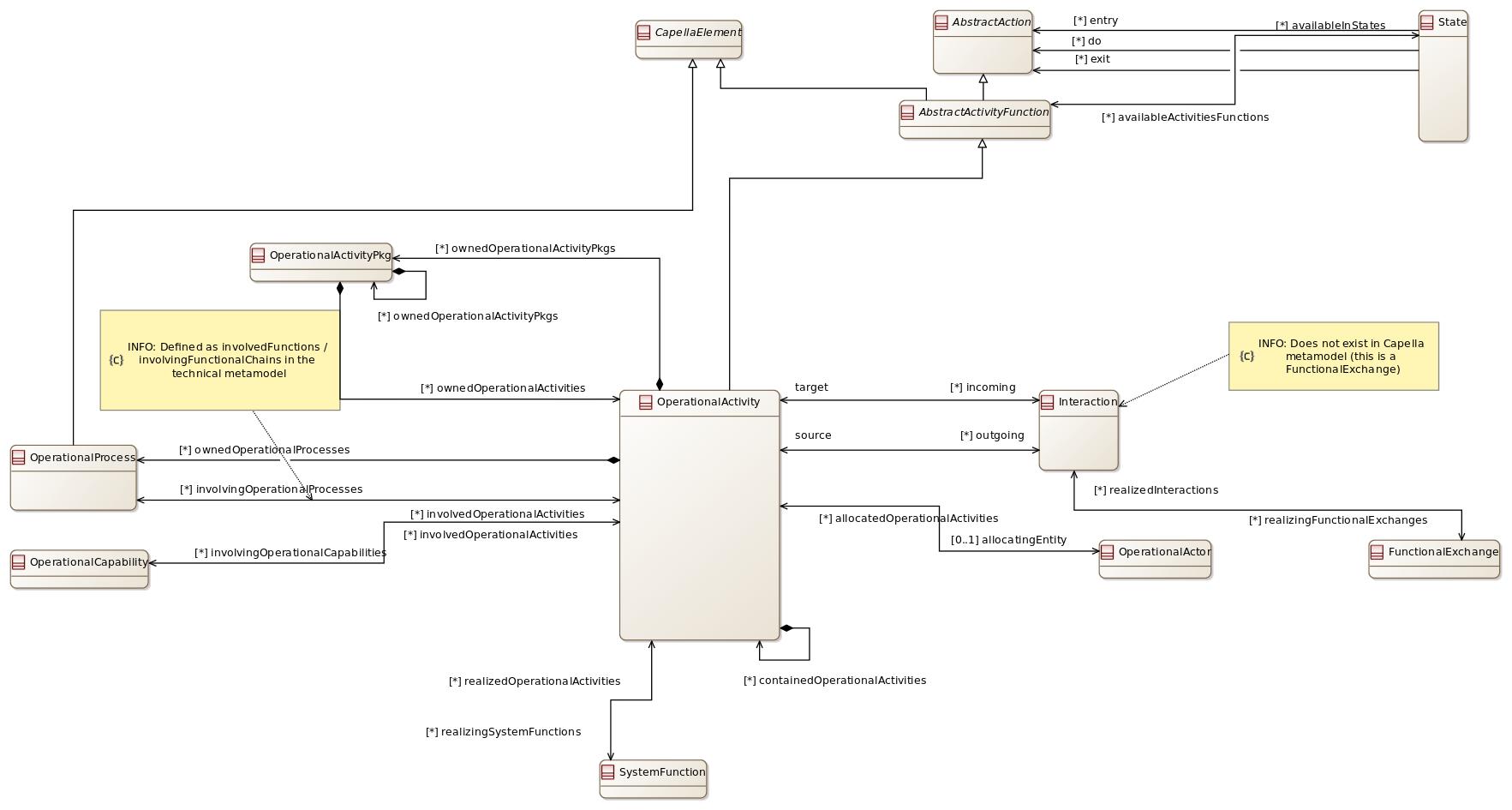
*No method defined*

# [Package] oa

Diagrams:

Here are the diagrams of representing the elements introduced in this package:

[CDB] OperationalActivity:



## [Type] OperationalAnalysis

Description:

The element containing all definitions from the Operational Analysis.

The Operational Analysis aims at defining what the users of the system need to accomplish

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| operationalActivityPkg getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| operationalCapabilityPkg getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| interfacePkg getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| dataPkg getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| entityPkg getter [ro] |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] OperationalActivityPkg

Description:

A package to contain OperationalActivity

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| ownedOperationalActivityPkgs getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedOperationalActivities getter [ro] |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] OperationalActivity

Description:

Process step or action/operation/service performed by an Operational entity / actor and likely to influence the system definition or usage. Implementing operational activities generally produces elements of interactions expected by other activities

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| containedOperationalActivities getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedOperationalActivityPkgs getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| incoming getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| outgoing getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| allocatingEntity getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedOperationalProcesses getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| involvingOperationalProcesses getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| involvingOperationalCapabilities getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| realizingSystemFunctions getter/setter |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] Interaction

Description:

Oriented dependency between Operational Activities

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| source getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| target getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| allocatingCommunicationMean getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| involvingOperationalProcesses getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| exchangedItems getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| realizingFunctionalExchanges getter/setter |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] OperationalProcess

Description:

An operational process is used to describe a particular context for performing operational activities to contribute to one or more operational capabilities

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| involvedOperationalActivities getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| involvedInteractions getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| involvedOperationalProcesses getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| preCondition getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| postCondition getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| availableInStates getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| involvingOperationalCapabilities getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| realizingFunctionalChains getter/setter |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] OperationalCapabilityPkg

Description:

A package to contain OperationalCapabilities

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| ownedOperationalCapabilityPkgs getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedOperationalCapabilities getter [ro] |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] OperationalCapability

Description:

An operational capability is an ability, expected of one or more operational entities / actors. An operational capability is characterized by a set of operational processes and scenarios

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| ownedOperationalProcesses getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| involvedOperationalProcesses getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| involvedOperationalActivities getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| involvedEntities getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| realizingCapabilities getter/setter |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] EntityPkg

Description:

A package to define Operational entities / actors

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| ownedEntityPkgs getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedEntities getter [ro] |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] OperationalEntity

Description:

An Operational Entity is a real world entity (other system, device, group or organisation…) carrying Operational Activities to which the system is likely to contribute

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| ownedEntities getter [ro] |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] OperationalActor

Description:

An Operational Actor is a kind of Operational Entity, usually human. It cannot be broken down

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| incomingCommunicationMeans getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| outgoingCommunicationMeans getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| allocatedOperationalActivities getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| involvingOperationalCapabilities getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedStateMachines getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| realizingSystemActors getter/setter |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] CommunicationMean

Description:

Describes the media between the Operational Entities / Actors to support the Operational Interactions

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| sourceEntity getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| targetEntity getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| allocatedInteractions getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| convoyedInformations getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| realizingComponentExchanges getter/setter |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

# [Package] sa

Diagrams:

*No diagram defined for this package*

## [Type] SystemAnalysis

Description:

The element containing all definitions from the System Analysis.

The System Analysis aims at defining what the system has to accomplish for the users

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| systemFunctionPkg getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| capabilityPkg getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| interfacePkg getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| dataPkg getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| systemComponentPkg getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| missionPkg getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| system getter [ro] |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] SystemFunctionPkg

Description:

A package to contain SystemFunctions

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| ownedSystemFunctionPkgs getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedSystemFunctions getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedCategories getter [ro] |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] SystemFunction

Description:

The definition of a Function in the System Analysis

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| containedSystemFunctions getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedSystemFunctionPkgs getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| realizedOperationalActivities getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| realizingLogicalFunctions getter/setter |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] CapabilityPkg

Description:

A package to contain Capabilities

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| ownedCapabilityPkgs getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedCapabilities getter [ro] |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] Capability

Description:

The ability of the system to supply a service contributing to fulfilling one or more Missions. A Capability represents a system usage context. It is characterized by a set of Functional Chains and Scenarios it references.

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| purposeMissions getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| realizedOperationalCapabilities getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| realizingCapabilityRealizations getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| involvedSystemActors getter/setter |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] SystemComponentPkg

Description:

A package to contain the System and the Actors

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| ownedSystemComponentPkgs getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedSystem getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedActors getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedComponentExchangeCategories getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedPhysicalLinkCategories getter [ro] |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] System

Description:

Set of elements functioning as a whole, responding to customer and user demand and needs

The System defined in the SystemAnalysis is seen as a black box

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

*No attribute defined*

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] SystemActor

Description:

Entity (human or not) that is external to the System (in term of responsibility), interacting with it, via its interfaces

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| isHuman getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedActors getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedSystemComponentPkgs getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| involvingMissions getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| realizedOperationalEntities getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| involvingCapabilities getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| realizingLogicalActors getter/setter |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] MissionPkg

Description:

A package to contain Missions

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| ownedMissionPkgs getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedMissions getter [ro] |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] Mission

Description:

High-level goal to which the System should contribute. To be fulfilled, a Mission should use a number of system Functions regrouped within one or more Capabilities

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| exploitedCapabilities getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| involvedActors getter/setter |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

# [Package] la

Diagrams:

*No diagram defined for this package*

## [Type] LogicalArchitecture

Description:

The element containing all definitions from the Logical Architecture.

The Logical Architecture (or conceptual solution) aims at defining how the system will work in order to fulfil expectations

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| logicalFunctionPkg getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| capabilityRealizationPkg getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| interfacePkg getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| dataPkg getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| logicalComponentPkg getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| logicalSystem getter [ro] |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] LogicalFunctionPkg

Description:

A package to contain LogicalFunctions

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| ownedLogicalFunctionPkgs getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedLogicalFunctions getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedCategories getter [ro] |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] LogicalFunction

Description:

The definition of a Function in the Logical Architecture

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| containedLogicalFunctions getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedLogicalFunctionPkgs getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| realizedSystemFunctions getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| realizingPhysicalFunctions getter/setter |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] CapabilityRealizationPkg

Description:

A package to contain CapabilityRealizations

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| ownedCapabilityRealizationPkgs getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedCapabilityRealizations getter [ro] |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] CapabilityRealization

Description:

The implementation of the system Capabilities in Logical Architecture and Physical Architecture

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| realizedCapabilities getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| realizedCapabilityRealizations getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| realizingCapabilityRealizations getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| involvedLogicalActors getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| involvedLogicalComponents getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| involvedPhysicalComponents getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| involvedPhysicalActors getter/setter |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] LogicalComponentPkg

Description:

A package to contain the LogicalSystem, LogicalComponents and LogicalActors

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| ownedLogicalComponentPkgs getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedLogicalSystem getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedLogicalActors getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedLogicalComponents getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedComponentExchangeCategories getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedPhysicalLinkCategories getter [ro] |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] LogicalSystem

Description:

The definition of the system in Logical Architecture

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| ownedLogicalComponents getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedLogicalComponentPkgs getter [ro] |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] LogicalComponent

Description:

Logical Components are the artefacts enabling a notional decomposition of the system as a "white box", independently from any technological solutions, but dealing with major system decomposition constraints

Logical components are identified according to logical abstractions (i.e. functional grouping, logical interfaces)

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| ownedLogicalComponents getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedLogicalComponentPkgs getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| isHuman getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| realizingBehaviorPCs getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| involvingCapabilityRealizations getter/setter |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] LogicalActor

Description:

An external Actor defined in the Logical Architecture

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| ownedLogicalActors getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedLogicalComponentPkgs getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| realizedSystemActors getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| isHuman getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| realizingPhysicalActors getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| involvingCapabilityRealizations getter/setter |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

# [Package] pa

Diagrams:

*No diagram defined for this package*

## [Type] PhysicalArchitecture

Description:

The element containing all definitions from the Physical Architecture.

The Physical Architecture (or finalized solution) aims at defining how the system will be developed and built

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| physicalFunctionPkg getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| capabilityRealizationPkg getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| interfacePkg getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| dataPkg getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| physicalComponentPkg getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| physicalSystem getter [ro] |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] PhysicalFunctionPkg

Description:

A package to contain PhysicalFunctions

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| ownedPhysicalFunctionPkgs getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedPhysicalFunctions getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedCategories getter [ro] |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] PhysicalFunction

Description:

The definition of a Function in the Physical Architecture

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| containedPhysicalFunctions getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedPhysicalFunctionPkgs getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| realizedLogicalFunctions getter/setter |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] PhysicalComponentPkg

Description:

A package to contain the PhysicalSystem, PhysicalComponents and PhysicalActors

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| ownedPhysicalComponentPkgs getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedPhysicalSystem getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedPhysicalActors getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedPhysicalComponents getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedComponentExchangeCategories getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedPhysicalLinkCategories getter [ro] |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] PhysicalSystem

Description:

The definition of the system in Physical Architecture

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| ownedPhysicalComponents getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedPhysicalComponentPkgs getter [ro] |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] AbstractPhysicalArtifact [Abstract]

Description:

An abstract type to defined the relation between elements of the Physical Architecture and ConfigurationItems

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| allocatorConfigurationItems |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] PhysicalComponent [Abstract]

Description:

A generic Physical Component which can be either a BehaviorPC or a NodePC

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| kind |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedPhysicalComponents |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedPhysicalComponentPkgs |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| isHuman |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| involvingCapabilityRealizations |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] BehaviorPC

Description:

System component in charge of implementing / realizing some of the functions devoted to the system

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| deployingNodePC getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| realizedLogicalComponents getter/setter |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] NodePC

Description:

Component hosting a number of behavioral components, providing them with the resource they require to function and to interact with their environment

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| deployedBehaviorPCs getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedStateMachines getter [ro] |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] PhysicalActor

Description:

An external Actor defined in the Physical Architecture

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| ownedPhysicalActors getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedPhysicalComponentPkgs getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| realizedLogicalActors getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| isHuman getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| involvingCapabilityRealizations getter/setter |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Enumeration] PhysicalComponentKind

List of values:

* UNSET
* HARDWARE
* HARDWARE\_COMPUTER
* SOFTWARE
* SOFTWARE\_DEPLOYMENT\_UNIT
* SOFTWARE\_EXECUTION\_UNIT
* SOFTWARE\_APPLICATION
* FIRMWARE
* PERSON
* FACILITIES
* DATA
* MATERIALS
* SERVICES
* PROCESSES

Used by:

# [Package] epbs

Diagrams:

*No diagram defined for this package*

## [Type] EPBSArchitecture

Description:

The element containing all definitions from the End-Product Breakdown Structure.

The End-Product Breakdown Structure aims at defining the construction strategy of the product, taking into account industrial and subcontracting constraints

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| capabilityRealizationPkg getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| configurationItemPkg getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| dataPkg getter [ro] |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] ConfigurationItemPkg

Description:

A package to contain ConfigurationItems

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| ownedConfigurationItemPkgs getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedConfigurationItems getter [ro] |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] ConfigurationItem

Description:

System part to be acquired or produced, in as many copies as the physical architecture requires

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| itemIdentifier getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| kind getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedConfigurationItems getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedConfigurationItemPkgs getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| allocatedPhysicalArtifacts getter/setter |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Enumeration] ConfigurationItemKind

List of values:

* Unset
* COTSCI
* CSCI
* HWCI
* InterfaceCI
* NDICI
* PrimeItemCI
* SystemCI

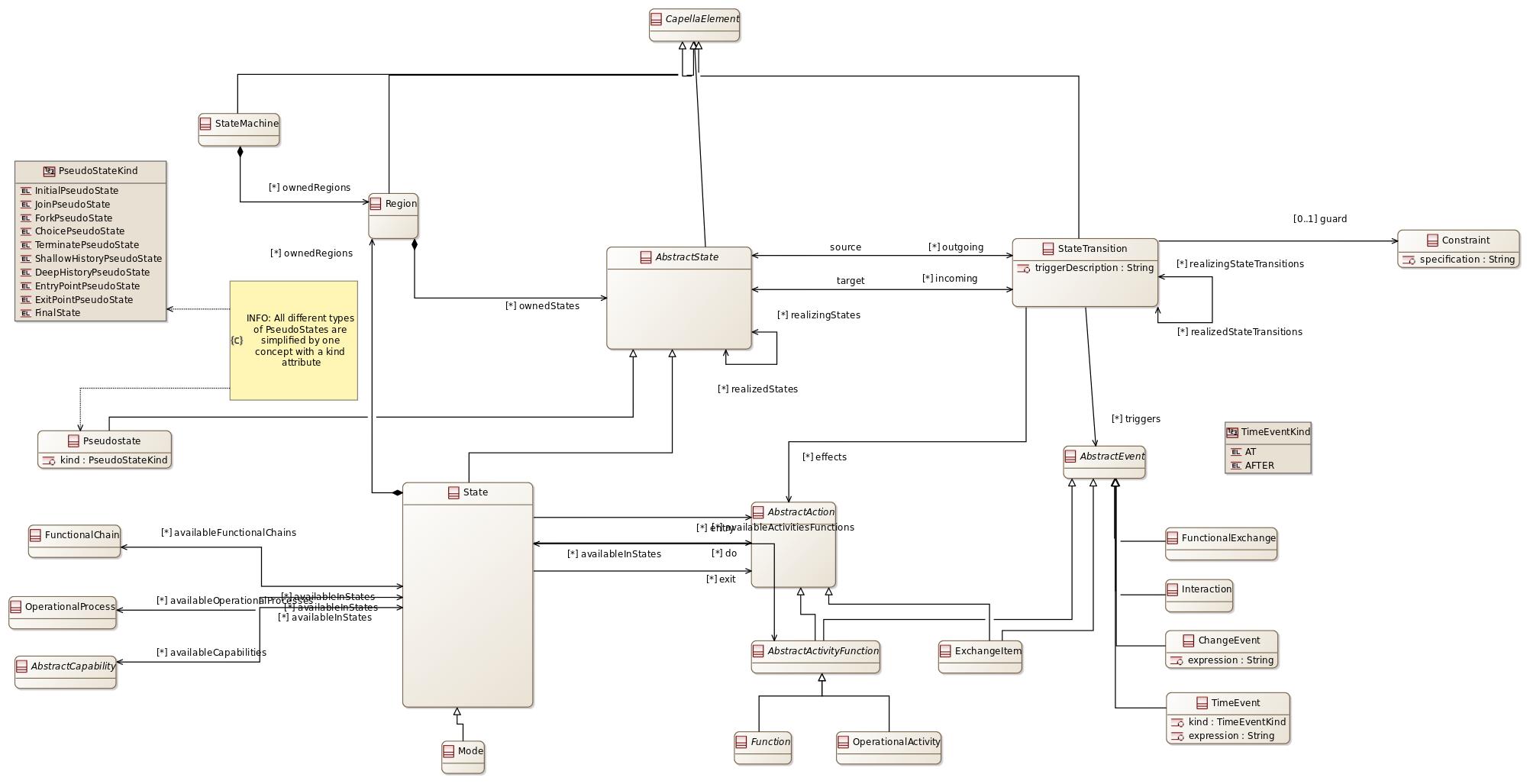
Used by:

# [Package] statemachine

Diagrams:

Here are the diagrams of representing the elements introduced in this package:

[CDB] StateMachine:



## [Type] StateMachine

Description:

State Machine is a way to define some of the expected behavior of the System, a Component or an external Actor

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| ownedRegions getter [ro] |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] AbstractState [Abstract]

Description:

An abstract type to define the generic relation of modes and states

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| incoming |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| outgoing |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| realizedStates |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| realizingStates |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] State

Description:

A State is a context undergone by the system, an actor or a component in specific circumstances (for example imposed by the environment)

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| ownedRegions getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| availableActivitiesFunctions getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| entry getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| do getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| exit getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| availableFunctionalChains getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| availableOperationalProcesses getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| availableCapabilities getter/setter |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] Mode

Description:

A Mode is a behavior expected from the system, an Actor or a component in chosen conditions

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

*No attribute defined*

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] Pseudostate

Description:

A pseudo states are transitive states (meaning they don't remain active). They are used to define entry / exit point of a state machine, and to connect multiple transitions into more complex transition paths

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| kind getter [ro] |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] Region

Description:

A region is an orthogonal part of either a composite state or a state machine.

Inside of a region, only one more or state can be active at a time

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| ownedStates getter [ro] |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] StateTransition

Description:

A possible transition between 2 modes or 2 states

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| triggerDescription getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| source getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| target getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| triggers getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| guard getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| effects getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| realizedStateTransitions getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| realizingStateTransitions getter/setter |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] ChangeEvent

Description:

An event defined by WHEN something occurs

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| expression getter/setter |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] TimeEvent

Description:

An event defined by AT a given time, or AFTER a certain time

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| kind getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| expression getter/setter |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] AbstractActivityFunction [Abstract]

Description:

An abstract type to define the link between mode / state and activity / function (available in states)

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| availableInStates |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] AbstractAction [Abstract]

Description:

A generic action which can be triggered by a Mode / State or a StateTransition

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

*No attribute defined*

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] AbstractEvent [Abstract]

Description:

An generic event which can trigger a transition

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

*No attribute defined*

Relations:

*No relation defined*

Methods:

*No method defined*

## [Enumeration] PseudoStateKind

List of values:

* InitialPseudoState
* JoinPseudoState
* ForkPseudoState
* ChoicePseudoState
* TerminatePseudoState
* ShallowHistoryPseudoState
* DeepHistoryPseudoState
* EntryPointPseudoState
* ExitPointPseudoState
* FinalState

Used by:

## [Enumeration] TimeEventKind

List of values:

* AT
* AFTER

Used by:

# [Package] scenario

Diagrams:

*No diagram defined for this package*

## [Type] Scenario

Description:

A scenario of use of the system defined by a specific sequence

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| kind getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| preCondition getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| postCondition getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedInstanceRoles getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedMessages getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedStateFragments getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedCombinedFragments getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedConstraintDurations getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| referencedScenarios getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| realizedScenarios getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| realizingScenarios getter/setter |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] InstanceRole

Description:

The involvement of an element (function, system, component or actor) in a scenario

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| representedInstance getter [ro] |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] AbstractInstance [Abstract]

Description:

A generic element which can be involved in a scenario as an InstanceRole

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

*No attribute defined*

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] SequenceMessage

Description:

An exchange between InstanceRole performed in the frame of a scenario

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| kind getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| sendingInstanceRole getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| receivingInstanceRole getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| invokedExchange getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| exchangedItems getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| invokedOperation getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| exchangeContext getter [ro] |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] AbstractExchange [Abstract]

Description:

A generic exchange which can be used in a scenario as a sequence message

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| invokingSequenceMessages |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] StateFragment

Description:

The call of a function or mode / state by an InstanceRole in the context of a scenario

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| coveredInstanceRole getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| relatedState getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| relatedActivityFunction getter [ro] |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] CombinedFragment

Description:

The identification of a specific operator (ALT, OPT, LOOP...) in the sequence of a scenario

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| operator getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| operands getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| coveredInstanceRoles getter [ro] |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] Operand

Description:

A specific "region" in a Combined Fragment

For example, an ALTERNATIVE contains one operand for each of the alternative conditions

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| guard getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| referencedMessages getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| referencedFragments getter [ro] |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] ConstraintDuration

Description:

A constraint about the execution time of a scenario defined between 2 points

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| duration getter/setter |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Enumeration] ScenarioKind

List of values:

* INTERFACE
* DATA\_FLOW
* INTERACTION
* FUNCTIONAL

Used by:

## [Enumeration] MessageKind

List of values:

* UNSET
* ASYNCHRONOUS\_CALL
* SYNCHRONOUS\_CALL
* REPLY
* DELETE
* CREATE
* TIMER

Used by:

## [Enumeration] InteractionOperatorKind

List of values:

* UNSET
* ALT
* OPT
* PAR
* LOOP
* CRITICAL
* NEG
* ASSERT
* STRICT
* SEQ
* IGNORE
* CONSIDER

Used by:

# [Package] cs

Diagrams:

*No diagram defined for this package*

## [Type] Node [Abstract]

Description:

An abstract type defining the generic relation of a Node (implementation ressource) element

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| containedPhysicalPorts |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| physicalLinks |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| involvingPhysicalPaths |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedPhysicalLinkCategories |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedPhysicalPaths |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] PhysicalPort

Description:

A port on a Node component defining a physical interaction point

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| physicalLinks getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| allocatedComponentPorts getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| realizedPhysicalPorts getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| realizingPhysicalPorts getter/setter |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] PhysicalLink

Description:

Means of communication, transport or routing between two Node components, used as a support for behavioral exchanges

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| connectedPhysicalPorts getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| categories getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| involvingPhysicalPaths getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| connectedComponents getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| allocatedComponentExchanges getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| realizedPhysicalLinks getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| realizingPhysicalLinks getter/setter |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] PhysicalLinkCategory

Description:

A regroupement of PhysicalLinks for graphical simplification of diagrams

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| links getter/setter |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] PhysicalPath

Description:

Set of Physical Links defining a continuous path likely to route one or more behavioral exchanges

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| involvedPhysicalLinks getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| involvedNodePCs getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| allocatedComponentExchanges getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| realizedPhysicalPaths getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| realizingPhysicalPaths getter/setter |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] InterfacePkg

Description:

A package to contain Interfaces

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| ownedInterfacePkgs getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedInterfaces getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedExchangeItems getter [ro] |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] Interface

Description:

The definition of ExchangeItems which can be send / received by a ComponentPort

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| visibility getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedExchangeItemAllocations getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| exchangeItems getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| providingComponentPorts getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| requiringComponentPorts getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| userComponents getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| implementorComponents getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| super getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| sub getter/setter |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] ExchangeItemAllocation

Description:

The involvement of an ExchangeItem by an Interface.

Mainly used for the involvement of ExchangeItems in Interface Scenarios based on the definition of Interfaces between components

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| transmissionProtocol getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| acquisitionProtocol getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| allocatedItem getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| invokingSequenceMessages getter/setter |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] ExchangeItem

Description:

Ordered set of references to elements carried together during an interaction or exchange between functions, components and actors. The elements are carried simultaneously, in the same conditions, with the same non-functional properties. The “elements” are called data

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| abstract getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| final getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| exchangeMechanism getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedElements getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| allocatorInterfaces getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| super getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| sub getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| realizedExchangeItems getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| realizingExchangeItems getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| realizingOperations getter/setter |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] ExchangeItemElement

Description:

A part of the information contained by an ExchangeItem

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| type getter/setter |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Enumeration] ExchangeMechanism

List of values:

* UNSET
* FLOW
* OPERATION
* EVENT
* SHARED\_DATA

Used by:

## [Enumeration] CommunicationLinkProtocol

List of values:

* UNSET
* UNICAST
* MULTICAST
* BROADCAST
* SYNCHRONOUS
* ASYNCHRONOUS
* READ
* ACCEPT

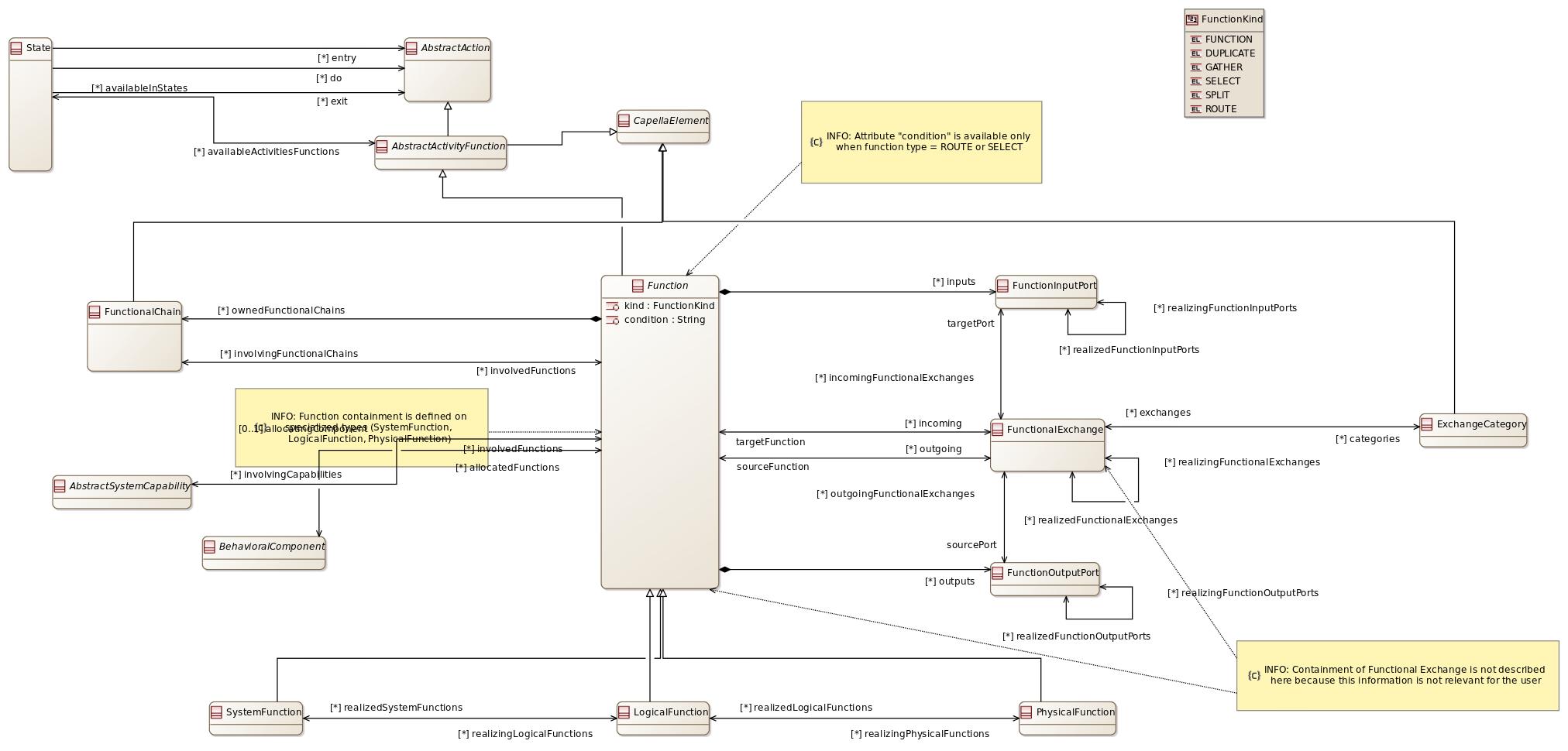
Used by:

# [Package] fa

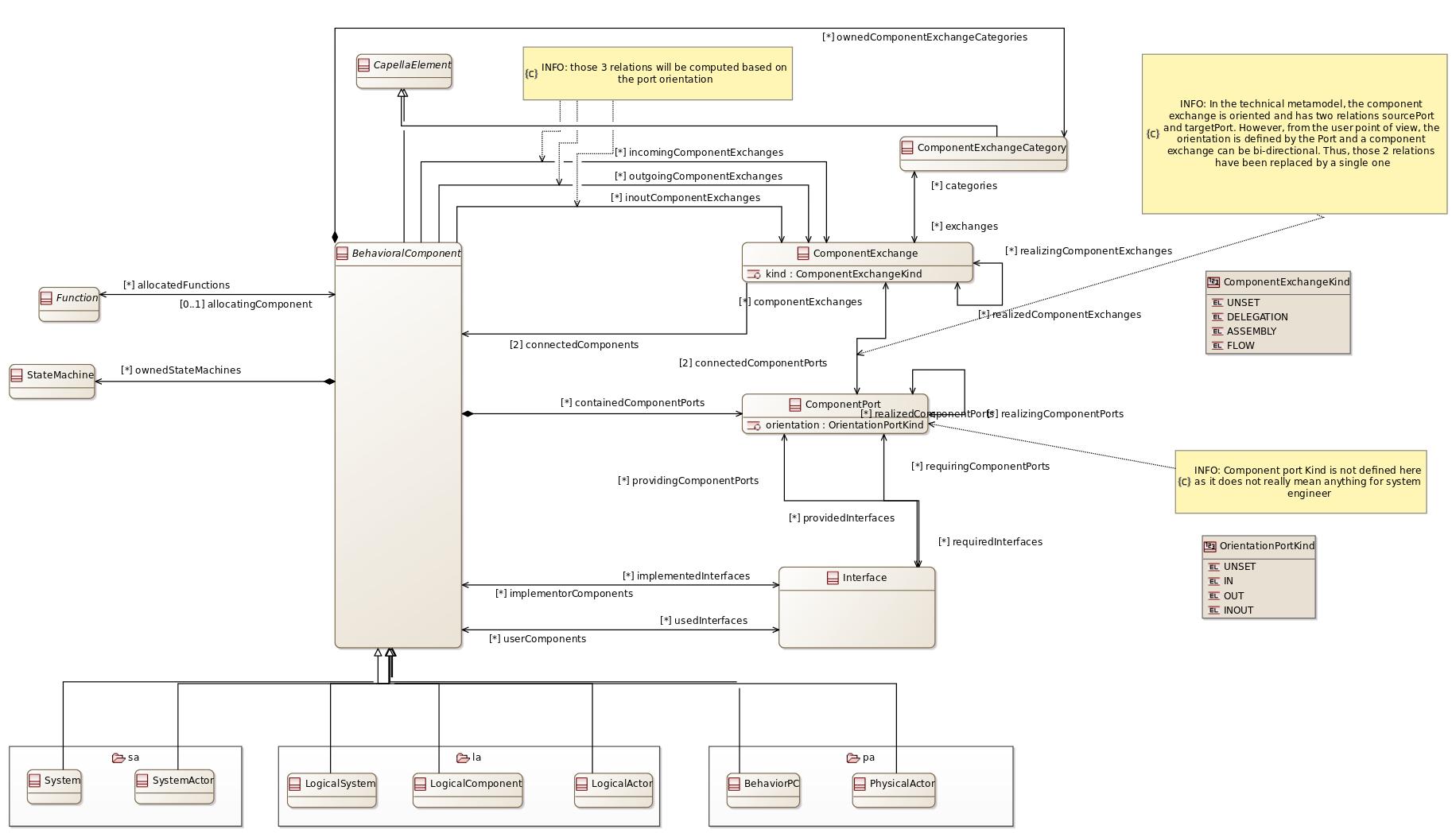
Diagrams:

Here are the diagrams of representing the elements introduced in this package:

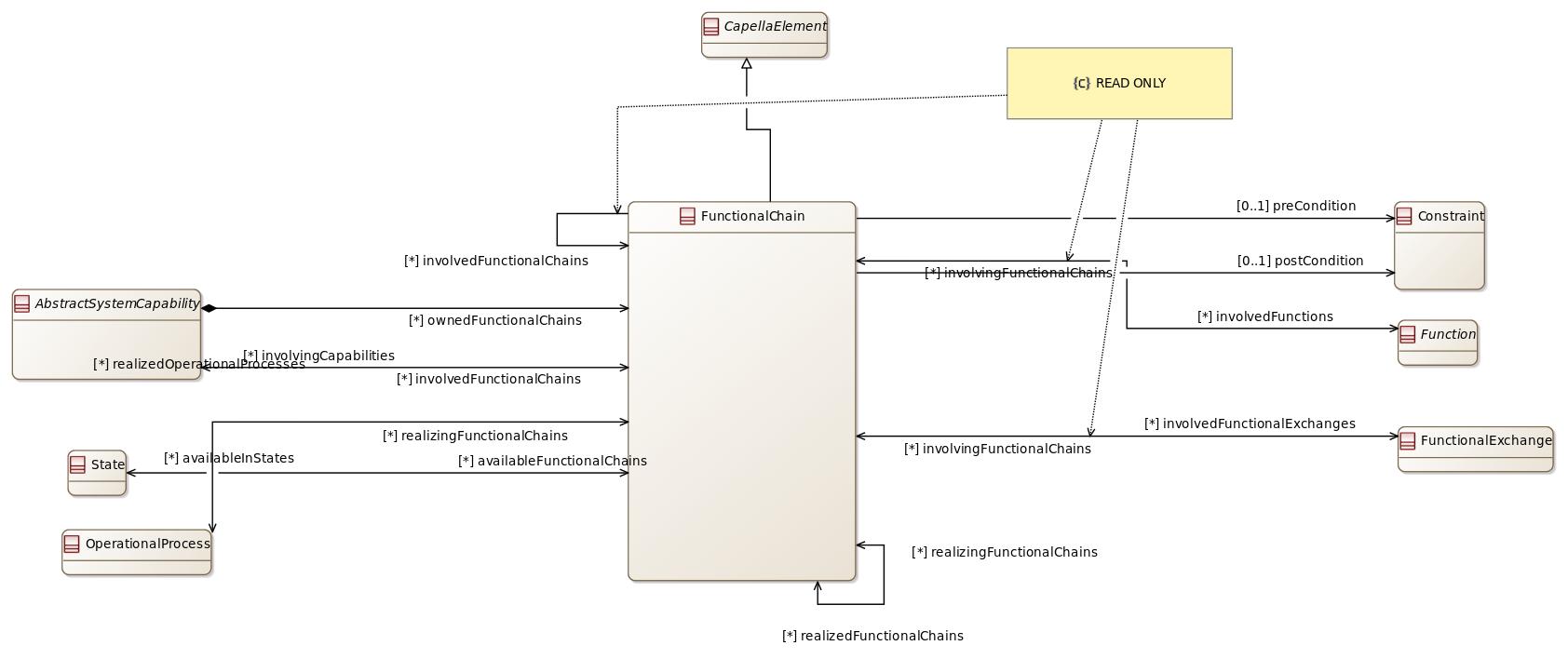
[CDB] Function:



[CDB] BehavioralComponent:



[CDB] FunctionalChain:



## [Type] Function [Abstract]

Description:

Action performed by the System, an Actor or a component, in order to realize a Capability

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| kind |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| condition |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| inputs |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| outputs |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| incoming |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| outgoing |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| allocatingComponent |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedFunctionalChains |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| involvingFunctionalChains |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| involvingCapabilities |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] FunctionPort [Abstract]

Description:

An generic FunctionPort to define the allocation with ComponentPorts

A Function Port specify what a Function is capable of producing or is requiring

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| allocatorComponentPort |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] FunctionInputPort

Description:

An FunctionInputPort defines what a Function is requiring

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| incomingFunctionalExchanges getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| incomingExchangeItems getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| realizedFunctionInputPorts getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| realizingFunctionInputPorts getter/setter |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] FunctionOutputPort

Description:

A FunctionOutputPort defines what a Function is capable of producing

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| outgoingFunctionalExchanges getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| outgoingExchangeItems getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| realizedFunctionOutputPorts getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| realizingFunctionOutputPorts getter/setter |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] FunctionalExchange

Description:

A functional exchange represents a dependency between a source function and a target one. Exchanges connect Function Ports, which specify what a Function is capable of producing or is requiring.

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| sourcePort getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| targetPort getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| sourceFunction getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| targetFunction getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| exchangedItems getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| involvingFunctionalChains getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| categories getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| allocatingComponentExchange getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| realizedFunctionalExchanges getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| realizingFunctionalExchanges getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| realizedInteractions getter/setter |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] ExchangeCategory

Description:

A regroupement of FunctionalExchanges for graphical simplification of diagrams

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| exchanges getter/setter |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] FunctionalChain

Description:

Describe the system behaviour in a particular usage context with references towards Functions and Functional Exchanges

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| preCondition getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| postCondition getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| involvedFunctions getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| involvedFunctionalExchanges getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| involvedFunctionalChains getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| involvingCapabilities getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| availableInStates getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| realizedOperationalProcesses getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| realizedFunctionalChains getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| realizingFunctionalChains getter/setter |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] BehavioralComponent [Abstract]

Description:

An abstract type to define the generic relation of behavioral elements

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| containedComponentPorts |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| incomingComponentExchanges |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| outgoingComponentExchanges |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| inoutComponentExchanges |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| allocatedFunctions |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| usedInterfaces |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| implementedInterfaces |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedStateMachines |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedComponentExchangeCategories |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] ComponentPort

Description:

A port on a BehavioralComponent defining a logical interaction point

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| orientation getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| componentExchanges getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| allocatedFunctionPorts getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| providedInterfaces getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| requiredInterfaces getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| allocatingPhysicalPorts getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| realizedComponentPorts getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| realizingComponentPorts getter/setter |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] ComponentExchange

Description:

Represent the interactions between Logical / Behavioral Components. Exchanges connects Component Ports.

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| kind getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| connectedComponentPorts getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| connectedComponents getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| categories getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| allocatedFunctionalExchanges getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| convoyedInformations getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| allocatingPhysicalLink getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| allocatingPhysicalPath getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| realizedCommunicationMeans getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| realizedComponentExchanges getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| realizingComponentExchanges getter/setter |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] ComponentExchangeCategory

Description:

A regroupement of ComponentExchanges for graphical simplification of diagrams

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| exchanges getter/setter |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] AbstractCapability [Abstract]

Description:

An abstract type to define the generic relations of all capabilities (operational and system)

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| preCondition |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| postCondition |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedScenarios |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| super |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| sub |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| includedCapabilities |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| includingCapabilities |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| extendedCapabilities |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| extendingCapabilities |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| availableInStates |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] AbstractSystemCapability [Abstract]

Description:

An abstract type to define the generic relation of system capabilities (as defined in SA, LA and PA)

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| ownedFunctionalChains |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| involvedFunctionalChains |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| involvedFunctions |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Enumeration] FunctionKind

List of values:

* FUNCTION
* DUPLICATE
* GATHER
* SELECT
* SPLIT
* ROUTE

Used by:

## [Enumeration] OrientationPortKind

List of values:

* UNSET
* IN
* OUT
* INOUT

Used by:

## [Enumeration] ComponentExchangeKind

List of values:

* UNSET
* DELEGATION
* ASSEMBLY
* FLOW

Used by:

# [Package] datavalue

Diagrams:

*No diagram defined for this package*

## [Type] DataValue [Abstract]

Description:

*No description*

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

*No attribute defined*

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] LiteralBooleanValue

Description:

*No description*

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

*No attribute defined*

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] BooleanReference

Description:

*No description*

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

*No attribute defined*

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] EnumerationReference

Description:

*No description*

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

*No attribute defined*

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] LiteralStringValue

Description:

*No description*

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

*No attribute defined*

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] StringReference

Description:

*No description*

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

*No attribute defined*

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] LiteralNumericValue

Description:

*No description*

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

*No attribute defined*

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] NumericReference

Description:

*No description*

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

*No attribute defined*

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] ComplexValue

Description:

*No description*

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

*No attribute defined*

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] ComplexValueReference

Description:

*No description*

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

*No attribute defined*

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] BinaryExpression

Description:

*No description*

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

*No attribute defined*

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] UnaryExpression

Description:

*No description*

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

*No attribute defined*

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] CollectionValueReference

Description:

*No description*

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

*No attribute defined*

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] CollectionValue

Description:

*No description*

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

*No attribute defined*

Relations:

*No relation defined*

Methods:

*No method defined*

# [Package] classes

Diagrams:

*No diagram defined for this package*

## [Type] DataPkg

Description:

*No description*

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| ownedClasses getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedDataPkgs getter/setter |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] DataType [Abstract]

Description:

*No description*

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

*No attribute defined*

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] Class

Description:

*No description*

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| abstract getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| final getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| primitive getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| visibility getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| containedProperties getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| containedOperations getter [ro] |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] Collection

Description:

*No description*

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| ordered getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| unique getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| minInclusive getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| maxInclusive getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| abstract getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| final getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| primitive getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| collectionKind getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| aggregationKind getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| visibility getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| containedOperations getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| minCard getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| maxCard getter [ro] |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] Union

Description:

*No description*

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| final getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| containedUnionProperties getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| discriminant getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| defaultProperty getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| kind getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| containedOperations getter [ro] |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] Association

Description:

*No description*

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

*No attribute defined*

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] Property

Description:

*No description*

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| type getter/setter |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] UnionProperty

Description:

*No description*

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

*No attribute defined*

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] Operation

Description:

*No description*

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| visibility getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| realizedExchangeItems getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedParameters getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| thrownExceptions getter/setter |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] Parameter

Description:

*No description*

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

*No attribute defined*

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] CapellaException

Description:

*No description*

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| abstract getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| visibility getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| super getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| sub getter/setter |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Enumeration] UnionKind

List of values:

* UNION
* VARIANT

Used by:

## [Enumeration] CollectionKind

List of values:

* ARRAY
* SEQUENCE

Used by:

## [Enumeration] AggregationKind

List of values:

* UNSET
* ASSOCIATION
* AGGREGATION
* COMPOSITION

Used by:

# [Package] datatype

Diagrams:

*No diagram defined for this package*

## [Type] PrimitiveDataType [Abstract]

Description:

*No description*

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| abstract |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| final |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| discrete |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| visibility |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| super |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| sub |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| realizedInformations |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| realizingInformations |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] Enumeration

Description:

*No description*

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| minInclusive getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| maxInclusive getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| pattern getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedLiterals getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| defaultValue getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| minValue getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| maxValue getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| nullValue getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| domainType getter/setter |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] EnumerationLiteral

Description:

*No description*

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

*No attribute defined*

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] BooleanType

Description:

*No description*

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| ownedLiterals getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| defaultValue getter/setter |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] StringType

Description:

*No description*

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| minInclusive getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| maxInclusive getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| pattern getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| minLength getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| maxLength getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| defaultValue getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| nullValue getter [ro] |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] NumericType

Description:

*No description*

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| minInclusive getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| maxInclusive getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| pattern getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| kind getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| minValue getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| maxValue getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| defaultValue getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| nullValue getter [ro] |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] PhysicalQuantity

Description:

*No description*

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| unit getter/setter |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] Unit

Description:

*No description*

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

*No attribute defined*

Relations:

*No relation defined*

Methods:

*No method defined*

## [Enumeration] NumericTypeKind

List of values:

* INTEGER
* FLOAT

Used by:

# [Package] PVMT

Diagrams:

*No diagram defined for this package*

## [Type] PVMT

Description:

*No description*

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

*No attribute defined*

Relations:

*No relation defined*

Methods:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Description** | **IN Param** | **Result** |

|  |  |  |  |
| --- | --- | --- | --- |
| getPVNames |  | elem: | pvs: |

|  |  |  |  |
| --- | --- | --- | --- |
| isPVDefined |  | elem:  PVName: | result: |

|  |  |  |  |
| --- | --- | --- | --- |
| getPVValue |  | elem:  PVName: | value: |

# [Package] Requirement

Diagrams:

*No diagram defined for this package*

## [Type] RequirementAddOn

Description:

*No description*

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

*No attribute defined*

Relations:

*No relation defined*

Methods:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Description** | **IN Param** | **Result** |

|  |  |  |  |
| --- | --- | --- | --- |
| getRequirementModules |  | model: | modules: |

|  |  |  |  |
| --- | --- | --- | --- |
| getIncomingRequirements |  | elem: | reqs: |

|  |  |  |  |
| --- | --- | --- | --- |
| getOutgoingRequirements |  | elem: | reqs: |

|  |  |  |  |
| --- | --- | --- | --- |
| getRelationType |  | elem1:  elem2: | type: |

## [Type] CapellaModule

Description:

*No description*

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| ownedRequirements getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| id getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| longName getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| name getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| prefix getter/setter |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] Requirement

Description:

*No description*

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| id getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| longName getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| name getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| chapterName getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| description getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| prefix getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| text getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedAttributes getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| ownedRelations getter [ro] |  | *defined on type* |

Relations:

*No relation defined*

Methods:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Description** | **IN Param** | **Result** |

|  |  |  |  |
| --- | --- | --- | --- |
| getAllAttributes |  |  | attributeNames: |

|  |  |  |  |
| --- | --- | --- | --- |
| getAttribute |  | attributeName: | value: |

|  |  |  |  |
| --- | --- | --- | --- |
| setAttribute |  | attributeName:  value: |  |

|  |  |  |  |
| --- | --- | --- | --- |
| getIncomingLinkedElems |  |  | elems: |

|  |  |  |  |
| --- | --- | --- | --- |
| getOutgoingLinkedElems |  |  | elems: |

## [Type] Folder

Description:

*No description*

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| ownedRequirements getter [ro] |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] Attribute

Description:

*No description*

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| definition getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| value getter/setter |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] ReqIFElement

Description:

*No description*

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

*No attribute defined*

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] AbstractRelation

Description:

*No description*

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| relationType getter [ro] |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] CapellaIncomingRelation

Description:

*No description*

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| source getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| target getter [ro] |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] CapellaOutgoingRelation

Description:

*No description*

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| source getter [ro] |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| target getter [ro] |  | *defined on type* |

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] AbstractType

Description:

*No description*

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

*No attribute defined*

Relations:

*No relation defined*

Methods:

*No method defined*

## [Type] RelationType

Description:

*No description*

Direct super types:

*No direct super type*

Direct sub-types:

*No direct sub-type*

Possible parent elements:

*No possible parent defined*

Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Attribute type** | **Defined on** |

|  |  |  |
| --- | --- | --- |
| longName getter/setter |  | *defined on type* |

|  |  |  |
| --- | --- | --- |
| name getter/setter |  | *defined on type* |

Relations:

*No relation defined*

Methods:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Description** | **IN Param** | **Result** |

|  |  |  |  |
| --- | --- | --- | --- |
| getRelationTypes |  |  | architecture: |

|  |  |  |  |
| --- | --- | --- | --- |
| getRelationTypeByLongName |  |  | architecture:  name: |

|  |  |  |  |
| --- | --- | --- | --- |
| getRelationTypeByName |  |  | architecture:  name: |