

interface: AssuranceCasePackageInterface [0..\*] – a number of optional assurance case package interfaces that the current package may implement

artifactPackage: ArtifactPackage [0..\*] (composition) – a number of optional artifact sub-packages

terminologyPackage: TerminologyPackage [0..\*] (composition) – a number of optional terminology sub-packages

argumentPackage: Argument::ArgumentPackage[0..\*] (composition) – a number of optional argument packages.

## Semantics

AssuranceCasePackage is the root class for creating structured assurance cases.

## 9.3 AssuranceCasePackageInterface

AssuranceCasePackageInterface is a kind of AssuranceCasePackage that defines an interface that may be exchanged between users. An AssuranceCasePackage may declare one or more ArtifactPackageInterfaces.

### Superclass

AssuranceCasePackage

An AssuranceCasePackageInterface will consist of at least one of an ArgumentPackageInterface, ArtifactPackageInterface, and/or a TerminologyPackageInterface. Conversely, any combination of an ArgumentPackageInterface, ArtifactPackageInterface, and/or a TerminologyPackageInterface will be contained within an AssuranceCasePackageInterface. The combination depends on what parts of the assurance case are to be made “public.”

### Associations

implements: AssuranceCasePackage[1] – the AssuranceCasePackage that the AssuranceCasePackageInterface declares.

### Semantics

AssuranceCasePackageInterface enables the declaration of the elements of an AssuranceCasePackage that might be referred to (cited) in another AssuranceCasePackage. These declarations are provided by containing AssuranceCasePackageInterface(s)/ArgumentPackageInterface(s)/ArtifactPackageInterface(s)/TerminologyPackageInterface(s) to the packages contained by the AssuranceCasePackage (for which the interface is provided).

### Constraints

AssuranceCasePackageInterface are only allowed to contain the following: AssuranceCasePackageInterface, ArgumentPackageInterfaces, ArtifactPackageInterfaces, and TerminologyPackages.

### OCL:

self.assuranceCasePackage->forall(acp|acp.oclIsTypeOf(AssuranceCasePackageInterface)) and

self.argumentPackage->forall(ap|ap.oclIsTypeOf(Argumentation::ArgumentPackageInterface)) and

self.artifactPackage->forall(ap|ap.oclIsTypeOf(Artifact::ArtifactPackageInterface)) and

self.terminologyPackage->forall(tp|tp.oclIsTypeOf(Terminology::TerminologyPackageInterface))

## 9.4 AssuranceCasePackageBinding

Sub-packages within the AssuranceCasePackage can be bound together by means of AssuranceCasePackageBindings. AssuranceCasePackageBindings bind the participant packages by means of ArgumentPackageBindings/TerminologyPackageBindings/ArtifactPackageBindings elements that bind the contained packages of the participant packages.

### Superclass

AssuranceCasePackage

A Binding can include any combination of reference to Package or PackageInterface of the same type. An AssuranceCasePackageBinding can also include Package(s) or PackageInterface(s) for Argument, Artifact and/or Terminology. Each Package or Package referenced by a PackageInterface can be a participant Package in a Binding.

### Associations

participantPackage: AssuranceCasePackage[2..\*] – references to AssuranceCasePackages which the AssuranceCasePackageBinding binds together.

The participantPackage provides access to its elements through a Package Interface, if an Package Interface is named in the Package Binding, or to all elements if the Package is named in the Package Binding.

Issue:  
SACM22-15

Issue:  
SACM22-15

Issue:  
SACM22-15

Issue:  
SACM22-15

## Semantics

The AssuranceCasePackageBinding includes a reference to each participant Package which can be either an AssuranceCasePackage or an AssuranceCasePackageInterface and that are external Packages to the AssuranceCasePackage that contains this AssuranceCasePackageBinding. Within the AssuranceCasePackageBinding, there might be zero or more ArgumentPackageBinding, ArtifactPackageBinding, or TerminologyPackageBinding. Each of these that exist would refer to at least two participant packages that could be a Package or PackageInterface of the same type.

AssuranceCasePackageBinding binds peer AssuranceCasePackages together to indicate the relationship between these AssuranceCasePackages. The bindings between AssuranceCasePackages consist of the bindings of the packages (i.e. ArgumentPackageBindings, ArtifactPackageBindings and TerminologyPackageBindings) contained in the AssuranceCasePackages, together with an optional ArgumentationPackage that asserts the relationship between +participantPackage.

## Constraints

The participantPackages should be either AssuranceCasePackage or AssuranceCasePackageInterfaces.

## OCL:

```
self.participantPackage->forall(pp|pp.ocllsTypeOf(AssuranceCase::AssuranceCasePackage) or  
pp.ocllsTypeOf(AssuranceCase::AssuranceCasePackageInterface))
```

## 10.3 TerminologyGroup

TerminologyGroup can be used to associate a number of TerminologyElements to a common group (e.g. representing a common type or purpose, or being of interest to a particular stakeholder).

### Superclass

TerminologyElement

### Associations

terminologyElement[0..\*] – an optional collection of TerminologyElements that are organised within the TerminologyGroup.

### Semantics

TerminologyGroup can be used to associate a number of TerminologyElements to a common group (e.g. representing a common type or purpose, or being of interest to a particular stakeholder). The name and the description of the TerminologyGroup should provide the semantic for understanding the TerminologyGroup. TerminologyGroups serve no structural purpose in the formation of the argument network, nor are they meant as a structural packaging mechanism (this should be done using TerminologyPackages).

## 10.4 TerminologyPackage

The TerminologyPackage is the container element for SACM terminology assets.

### Superclass

TerminologyElement

### Associations

TerminologyElement:TerminologyElement[0..\*] (composition) – TerminologyElements contained in the TerminologyPackage, it can be either TerminologyPackage (and its sub-types) or TerminologyAssets (or its sub-types).

### Semantics

TerminologyPackage contains the TerminologyElements that can be used within the naming and description of SACM arguments and artifacts. TerminologyPackages can be nested.

## 10.5 TerminologyPackageInterface

TerminologyPackageInterface is a kind of TerminologyPackage that defines an interface that may be exchanged between users. An TerminologyPackage may declare one or more TerminologyPackageInterfaces.

### Superclass

TerminologyElement

### Associations

implements:TerminologyPackage[1] – the TerminologyPackage that the TerminologyPackageInterface declares.

### Semantics

TerminologyPackageInterface enables the declaration of the elements of an TerminologyPackage that might be referred to (cited) in another TerminologyPackage, thus the elements can be used for assurance in the scope of the latter AssuranceCasePackage.

A TerminologyPackageInterface resides inside the TerminologyPackage to which it refers. It refers to TerminologyElements using isCitation=True that reside within the same TerminologyPackage as itself.

## 10.6 TerminologyPackageBinding

Elements within the TerminologyPackage can be bound together by means of TerminologyPackageBindings. TerminologyPackageBindings bind the participant packages by means of terminology elements that connect the cited elements of the participant packages.

## Superclass

TerminologyPackage

## Semantics

TerminologyPackageBinding binds TerminologyPackages together to indicate the relationship between two TerminologyPackages.

A TerminologyPackageBinding resides within an AssuranceCasePackageBinding. It contains references, using isCitation=True to each TerminologyElement and connects TerminologyElements from different TerminologyPakages.

## Constraints

The participantPackages should be either TerminologyPackage or TerminologyPackageInterface

OCL: self.participantPackage->forall(pp|pp.ocllsKindOf(Terminology::TerminologyPackage))

## 10.7 TerminologyAsset (abstract)

The TerminologyAsset Class is the abstract class for the different types of terminology elements represented in SACM.

## Superclass

TerminologyElement

## Semantics

TerminologyAssets represent all of the elements required to model and categorize expressions in SACM (expressions and terminology categories).

## 10.8 Category

The Category class describes categories of ExpressionElements (Terms and Expressions) and can be used to group these elements within TerminologyPackages.

## Superclass

TerminologyAsset

## Semantics

Terms and ExpressionElements can be said to belong to Categories. Categories can group Terms, Expressions, or a mixture of both. For example, a Category could be used to describe the terminology associated with a specific assurance standard, project, or system.

## 10.9 ExpressionElement (abstract)

The ExpressionElement class is the abstract class for the elements in SACM that are necessary for modeling expressions.

## Superclass

TerminologyAsset

## Attributes

value:String[1] – the value of the expression.

## Associations

category: Category [0..\*] – optionally associates the ExpressionElement with one or more terminology categories.

## 11.5 ArgumentPackageBinding

ArgumentElement within the ArgumentPackage can be bound together by means of ArgumentPackageBinding. ArgumentPackageBinding bind the participant packages by means of argument elements that connect the cited elements of the participant packages.

### Superclass

ArgumentPackage

### Associations

participantPackage:ArgumentPackage[2..\*] - the ArgumentPackages being mapped together by the ArgumentPackageBinding.

### Semantics

ArgumentPackageBindings can be used to map resolved dependencies between the Claims of two or more ArgumentPackages.

For example, one ArgumentPackage may contain a claim that needsSupport (i.e. currently has no supporting argument). An ArgumentPackageBinding can be used to record the mapping by means of containing a structured argument linkingArgumentElements that cite the claims in question.

ArgumentPackageBinding is a sub type of ArgumentPackage, it is used to record the argument that connects the arguments of two or more ArgumentPackages.

### Constraints

The participantPackages should be only ArgumentPackages

OCL: self.participantPackage->forall(pp|pp.ocIsTypeOf(Argument::ArgumentPackage))

The ArgumentElements contained by an ArgumentPackageBinding must be ArgumentElement citations to ArgumentElements contained within the ArgumentPackages associated by the participantPackage association.

## 11.6 ArgumentPackageInterface

ArgumentPackageInterface is a kind of ArgumentPackage that defines an interface that may be exchanged between users. An ArgumentPackage may declare one or more ArgumentPackageInterface.

### Superclass

ArgumentPackage

### Associations

implements:ArgumentPackage[1] – a reference to the ArgumentPackage which the ArgumentPackageInterface declares.

### Semantics

ArgumentPackageInterfaces can be used to declare (by means of containing ArgumentElement based citations) the ArgumentAssets contained in an ArgumentPackage that form part of the explicit, declared, interface of the ArgumentPackage.

For example, whilst an ArgumentPackage may contain many Claims, it may be desirable to create an ArgumentPackageInterface that cites only a subset of those claims that are intended to be mapped / used (e.g. by means of an ArgumentPackageBinding) by other ArgumentPackages. There may be more than one ArgumentPackageInterface for a given ArgumentPackage that reveal different aspects of the ArgumentPackage for different audiences.

An ArgumentPackageInterface resides inside the ArgumentPackage to which it refers. It refers to ArgumentationElements using isCitation=True that reside within the same ArgumentPackage as itself. Similar relationships exist for an ArtifactPackageInterface and for a TerminologyPackageInterface.

/Events/Participants/ Resources/Techniques, and between Aritfacts and Activities /Events/Participants/ Resources/Techniques Participants can be recorded by means ArtifactAssetRelationships.

## 12.2 ArtifactPackage

ArtifactPackage is the containing element for artifacts involved in a structured assurance case.

### Superclass

Base::ArtifactElement

### Associations

artifactElement:Base::ArtifactElement[0..\*] (composition) – a collection of ArtifactElements forming a artifact package in a structured assurance case.

### Semantics

ArtifactPackages contain ArtifactElements that represent the artifact forming part of a structured assurance case. ArtifactPackages can also be nested.

## 12.3 ArtifactGroup

ArtifactGroup can be used to associate a number of ArtifactElements to a common group (e.g., representing a common type or purpose, or being of interest to a particular stakeholder).

### Superclass

Base::ArtifactElement

### Associations

artifactElement:ArtifactElement[0..\*] – an optional collection of ArtifactElements organised within the ArtifactGroup.

### Semantics

ArtifactGroup can be used to associate a number of ArtifactElements to a common group (e.g., representing a common type or purpose, or being of interest to a particular stakeholder). The name and the description of the ArtifactGroup should provide the semantic for understanding the ArtifactGroup. ArtifactGroups serve no structural purpose in the formation of the argument network, nor are they meant as a structural packaging mechanism (this should be done using ArtifactPackage).

## 12.4 ArtifactPackageBinding

The ArtifactPackageBinding is a sub type of ArtifactPackage used to record ArtifactAssetRelationships between the ArtifactAssets of two or more ArtifactPackages.

### Superclass

ArtifactPackage

### Associations

participantPackage:ArtifactPackage[2..\*] - the ArtifactPackages containing the ArtifactAssets being related together by the ArtifactPackageBinding.

### Semantics

ArtifactPackageBindings can be used to map dependencies between the cited ArtifactAssets of two or more ArtifactPackages. For example, a binding could be used to record a ‘derivedFrom’ ArtifactAssetRelationship between the ArtifactAsset of one package to the ArtifactAsset of another.

*An ArtifactPackageBinding resides within an AssuranceCasePackageBinding. It contains references, using isCitation=True to each ArtifactAsset needed and defines relationships among ArtifactAssets from different ArtifactPackages.*

### Contraints

ArtifactPackageBindings must only contain ArtifactAssetRelationships with source and target Artifacts, with isCitation = true citing ArtifactAssets contained within the ArtifactPackages associated by participantPackage.

## 12.5 ArtifactPackageInterface

ArtifactPackageInterface is a kind of ArtifactPackage that defines an interface that may be exchanged between users. An ArtifactPackage may define one or more ArtifactPackageInterfaces.

### Superclass

ArtifactPackage

### Associations

implements:ArtifactPackage[1] - a reference to the ArtifactPackage which the ArtifactPackageInterface declares.

### Semantics

ArtifactPackageInterface enables the declaration of the elements of an ArtifactPackage that might be referred to (cited) in another ArtifactPackage. *An ArtifactPackageInterface resides inside the ArtifactPackage to which it refers. It refers to ArtifactAssets using isCitation=True that reside within the same ArtifactPackage as itself.*

### Constraints

ArtifactPackageInterfaces are only allowed to contain Artifacts with +isCitation=true citing ArtifactAssets within the ArtifactPackage with which this ArtifactPackageInterface is associated.

## 12.6 ArtifactAsset (abstract)

ArtifactAsset represents the artifact-specific pieces of information of an assurance case, in contrast to the argument-specific pieces of information.

### Superclass

Base::ArtifactElement

### Association

property:Property[0..\*] (composition) – an optional collection of Propert(ies) which enable the specification of the characteristics of an ArtifactAsset.

### Semantics

Information about artifacts is essential for any assurance case. The artifacts correspond, for instance, to the evidence provided in support of the arguments and claims of an assurance case. It is also important to have access to related pieces of information such as the provenance of an artifact, its lifecycle, and its properties. All this information might have to be consulted for developing confidence in the validity of an assurance case.

## 12.7 Artifact

Artifact represents the distinguishable units of data used in a structured assurance case.

### Superclass

ArtifactAsset

### Attributes

version: String[0..1] - the version of the artifact

date: date[0..1] - the date on which the artifact was created.

### Semantics

Artifacts correspond to the main evidentiary support for the arguments and claims of an assurance case: an Artifact can play the role of evidence of a Claim (AssertedEvidence), or of counterevidence (AssertedCountedEvidence with