

8 Structured Assurance Case Base Classes

8.1 General

This chapter presents the normative specification for the SACM Base Metamodel. It begins with an overview of the metamodel structure followed by a description of each element.

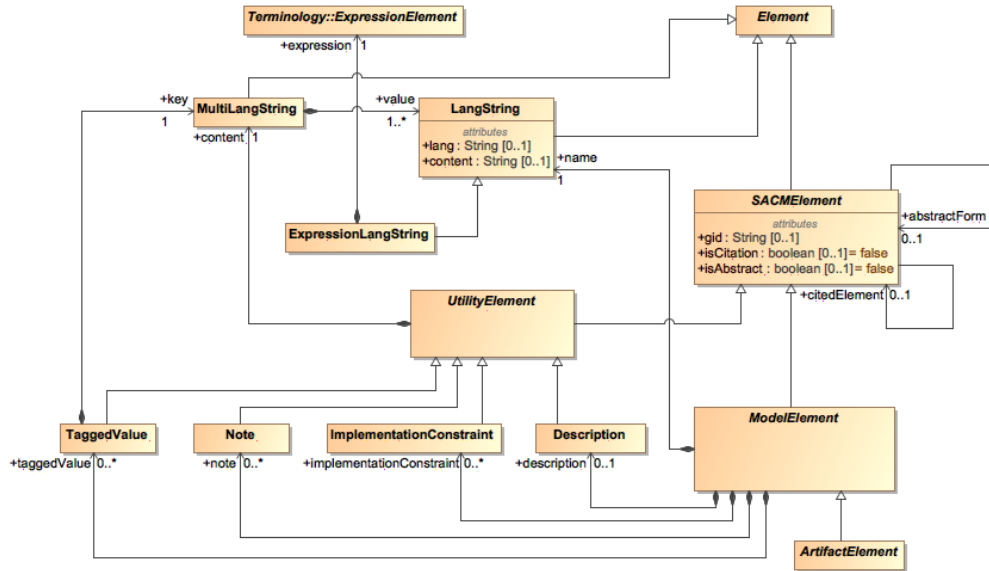


Figure 8.1 - Overall SACM Class Diagram

The Structured Assurance Case Base Classes express the foundational concepts and relationships of the base elements of the SACM metamodel and are utilized, through inheritance, by the bulk of the rest of the Structured Assurance Case Metamodel.

8.2 SACMElement (abstract)

SACMElement is the base class for SACM.

Superclass

MOF:Element

Attributes

gid:String[0..1] – a unique identifier that is unique within the scope of the model instance

isCitation[1]=false – a flag to indicate whether the SACMElement cites another SACMElement.

isAbstract[1]=false – a flag to indicate whether the SACMElement is considered to be abstract. For example, this can be used to indicate whether an element is part of a pattern or template.

Associations:

citedElement:SACMElement[0..1] – a reference to another SACMElement that the SACMElement cites

abstractForm:SACMElement[0..1] – an optional reference to another abstract SACMElement to which this concrete SACMElement conforms.

Semantics

All the elements of a structured assurance case effort created with SACM correspond to a SACMElement.

Constraints:

If citedElement is populated, isCitation must be true. OCL: self.citedElement <> null implies self.isCitation = true

When +abstractForm is used to refer to another SACMElement, +isAbstract of the SACMElement is false, and the +isAbstract of the referred SACMElement should be true. The referred SACMElement should be of the same type of the SACMElement. If ImplementationConstraints are expressed on the referred SACMElement, the SACMElement should satisfy these ImplementationConstraints.

8.3 LangString

LangString is the format SACM uses for description. It serves the same purpose as String but with the additional specification of the language used for the content.

Superclass

MOF:Element

Attributes

lang:String[0..1] – a field to indicate the language used in the string.

content:String[0..1] – the content of the string

Semantics

LangString serves the same purpose as String, SACM uses LangString for description, which containing the information of the language it uses in the content.

8.4 ExpressionLangString

ExpressionLangString is used to denote a structured expression, it contains a description (LangString) and it also (optionally) points to an ExpressionElement in the Terminology Package.

Superclass

LangString

Associations

expression:Terminology::ExpressionElement[1] (composition) – a reference to an ExpressionElement in the TerminologyPackage

Semantics

ExpressionLangString provides a means for description, it can also be used to link to an ExpressionElement in the Terminology package.

Constraints

If +expression is not empty, then +content should be empty.

8.5 MultiLangString

MultiLangString, as its name suggests, provides a means to describe things using different languages.

Superclass

Element

Associations

value:LangString[1..*] (composition) – contains the descriptions which bear the same meaning but in different languages

Semantics

MultiLangString provides a means to describing things using different languages. It contains a list of ExpressionLangString, which the user can specify their languages and the descriptions in the languages.

Constraints

For each of the ExpressionLangString in the +value property, their +lang should be unique.

8.6 ~~8.3~~ ModelElement (abstract)

ModelElement is the base element for the majority of modeling elements.

Superclass

SACMElement

Attributes

~~name: String – the name of the element~~

~~isAbstract: Boolean – a flag to indicate whether the ModelElement is considered to be abstract. This is used to indicate whether an element is part of a pattern or template.~~

Associations

~~implementationConstraint: ImplementationConstraint [0..*] – allows the description of any implementation constraint associated with converting the element from being abstract to being concrete~~

~~description: Description – the description of the element [0..1]~~

~~note: Annotation [0..*] – a collection of annotations associated with the element.~~

~~taggedValue: TaggedValue [0..*] – a collection of tagged values may be associated with each ModelElement~~

Semantics

All the individual and identifiable elements of a SACM model correspond to a ModelElement.

Constraints

~~ImplementationConstraints should only be specified if isAbstract is true.~~

ImplementationConstraints should only be specified if +isAbstract is true
OCL: self.implementationConstraint->size() > 0 implies self.isAbstract = true

8.7 ~~8.4~~ UtilityElement (abstract)

~~UtilityElement is an abstract element for a number of utility elements.~~

Superclass

SACMElement

Associations

~~expression: Expression [1] – the expression object containing the value of the UtilityElement (see Terminology section 10)~~

Semantics

UtilityElement supports the specification of additional information for a ModelElement.

8.8 ~~8.5~~ ImplementationConstraint

This class specifies details of any implementation constraints that must be satisfied whenever a referencing ModelElement is to be converted from *isAbstract = true* to *isAbstract = false*. For example in the context of a SACM pattern fragment, an element will need to satisfy the implementation rules of the pattern.

Superclass

UtilityElement

Semantics

ImplementationConstraints indicate the conditions to fulfill in order to allow an abstract ModelElement (*isAbstract = true*) to become non-abstract (*isAbstract = false*).

Constraints

ImplementationConstraints should only be specified if *isAbstract* is true.

8.9 ~~8.6~~ Description

This class specifies a description that may be associated with a ModelElement. In many cases Description is used to provide the 'content' of a SACM element. For example, it would be used to provide the text of a Claim.

Superclass

UtilityElement

Semantics

A Description provides details about ModelElements in relation to aspects such as their content or purpose. Therefore, Descriptions can be used to both characterize ModelElements and facilitate their understanding.

8.10 ~~8.7~~ Note

This class specifies a generic note that may be associated with a ModelElement. For example a note may include a number of explanatory comments.

Superclass

UtilityElement

Semantics

Notes are used to specify additional (typically optional) generic, unstructured, untyped information about a ModelElement. An example of this kind of information could be a comment about a ModelElement.

8.11 ~~8.6~~ TaggedValue

This class represents a simple key/value pair that can be attached to any element in SACM. This is a simple extension mechanism to allow users to add attributes to each element beyond those already specified in SACM.

Superclass

UtilityElement

Attributes

key: ~~Expression~~ – the key of the ~~tagged value~~

MultiLangString[1] (composition)



TaggedValue.



Semantics

TaggedValues can be used to specify attributes, and their corresponding values, for ModelElements.

~~Constraints~~

~~TaggedValues should not be used to document attributes that already form part of SACM (e.g., ArtifactProperty).~~