

8.2 SACMElement (abstract)

SACMElement is the base class for SACM.

Superclass

MOF:Element

/isCitation:Boolean[1]=false;

Attributes

/name:String[0..1] (composition) – the derived name of the SACMElement, which is the first element of elementName.

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

~~isCitation[0..1]=false~~ – a flag to indicate whether the SACMElement cites another SACMElement.

isSACMAbstract:Boolean

~~isAbstract[0..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.

elementName:ExpressionLangString[0..1] – the multi-language name of this SACMElement.

Associations:

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

abstraction

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

~~description: ExpressionLangString [0..1] – the description of the SACMElement.~~

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.~~

*{abstraction->notEmpty() implies abstraction.isSACMAbstract=true},
cited->notEmpty() implies oclTypeOf(cited),
elementName->notEmpty() and elementName->content->notEmpty() implies name=elementName->content->first(), cited->notEmpty() and
cited.isSACMAbstract=true implies isSACMAbstract=true}*

~~9 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.~~

Move to Terminology Class

9 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.

inherits description (LangString) and it also (optionally) points to an ExpressionElement.

Superclass

MultiLangString

Attributes

expression:ExpressionElement[0..1]

~~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.~~

9 8.5 MultiLangString

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

Superclass

~~Element~~ Foundation::Element

Associations

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

content:String[0..*] {ordered,nonunique} – the content of the string.
lang:String[0..*] {ordered} – a field to indicate the language used in the string.

Semantics

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

Constraints set of contents each of which is in a different language represented by the lang attribute.

~~For each of the LangString in the value feature, their lang must be unique.~~

9 8.3 ModelElement (abstract)

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

Superclass

SACMElement

Attributes

Associations

~~name:LangString[1] (composition) – the name of the ModelElement~~

~~implementationConstraint: ImplementationConstraint [0..*] (composition) – a collection of implementation constraints.~~

~~description: Description[0..1] (composition) – the description of the ModelElement.~~

~~note:Note[0..*] (composition) – a collection of notes for the ModelElement.~~

~~taggedValue: TaggedValue [0..*] (composition) – a collection of TaggedValues, TaggedValues can be used to describe additional features of a ModelElement~~

Semantics

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

Moved to, and revised in, 8.2 Associations

~~Constraints~~

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

~~OCL:~~

~~self.implementationConstraint->size() > 0 implies self.isAbstract = true~~

9 ~~8.7~~ 4 UtilityElement (abstract)

UtilityElement is the base element for a number of auxiliary elements which can be added to ModelElements.

Superclass

SACMElement

Associations

content:MultiLangString[0..1] (composition) – a MultiLangString to describe the content of the UtilityElement in (possibly) multiple languages

Semantics

UtilityElement supports the specification of additional information for a ModelElement.

9 ~~8.8~~ 5 ImplementationConstraint

ImplementationConstraint 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*).

~~9 ~~8.9~~ 6 Description~~

~~Description is used to specify 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.~~

Move to Artifact Class

9 ~~8.10~~ 7 ArtifactElement (abstract)

ArtifactElement acts as the base class for elements in other SACM packages. Essentially, all elements which extend ArtifactElement is considered to be an artifact, and therefore can be referenced using Argument:ArtifactReference.

Superclass

ModelElement