

9 Structured Assurance Case Base Classes

9.1 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.

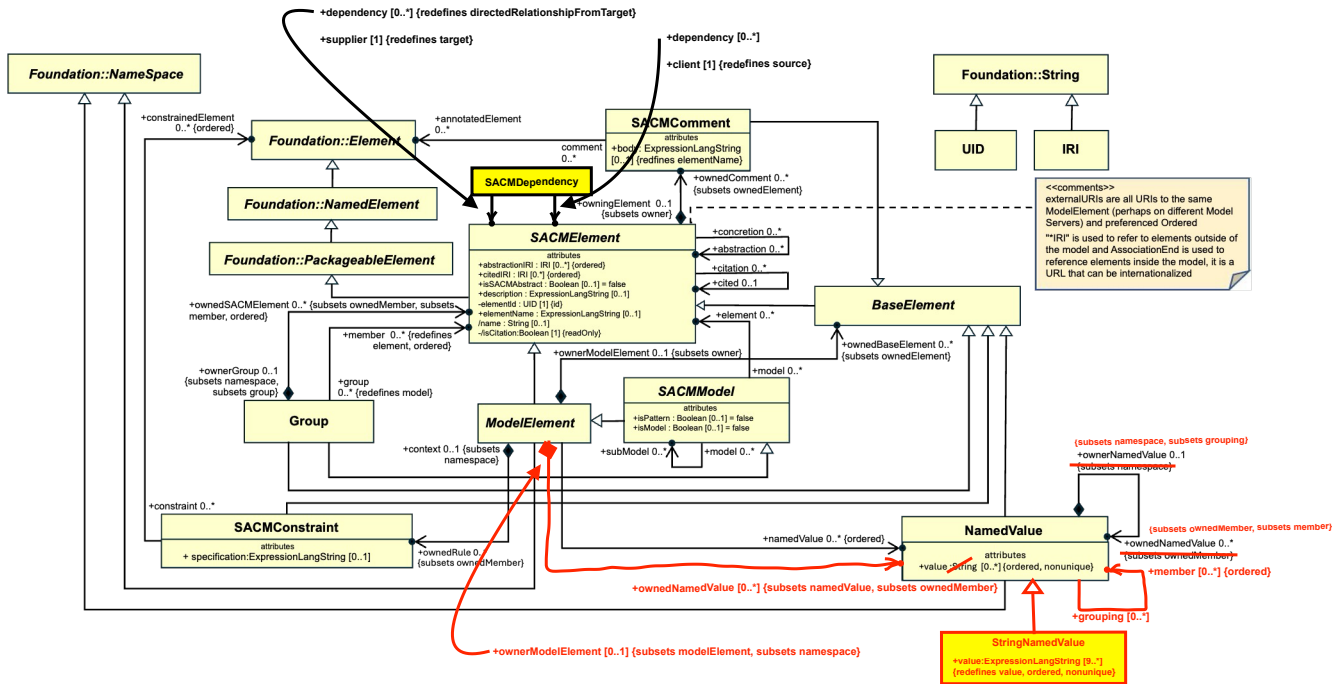


Figure 9.1 - Structured Assurance Case Base Classes Diagram

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.

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~~

Semantics

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

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.~~

LangIfContentMoreThan1
If content has more than one value, then lang must have string representing the languages of the values.
inv: content->size() > 1 implies lang->size() = content->size()

9 ~~8.6~~ 3 **ModelElement (abstract)**

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

Superclass

, Foundation::NameSpace

SACMElement

Associations

Associations
Attributes

~~ownedRule:SACMConstraint[0..*] (subsets ownedElement) – {subsets ownedMember}~~

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

~~ownedRole:SACMConstraint[0..*] – SACMConstraints owned by this ModelElement.~~

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

~~ownedBaseElement[0..*] {subsets ownedElement} - BaseElements owned by this ModelElement~~

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

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

namedValue:NamedValue

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

Semantics

ownedNamedValue:namedValue [0..*] {subsets namedValue, subsets ownedMember} - NamedValues owned by this ModelElement

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

Moved to, and revised in, 8.2

Semantics

ArtifactElement corresponds to the base class for specifying all the identifiable units of data modelled and managed in a structured assurance case effort.

~~9.8.11~~ 8 **Note** SACMComment

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

Superclass BaseElement

Attributes body:ExpressionLangString[0..1]{redefines elementName}

body:ExpressionLangString[0..1]

AssociationEnds

annotatedElement:Element[0..*] – Elements to which the SACMComment applies

Semantics

SACMComments

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.

~~9.8.12~~ 9 **TaggedValue** NamedValue attributes

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

BaseElement, Foundation::Namespace

Attributes

value:String [0..*] {ordered, nonunique}

Associations

key:MultiLangString[1] (composition) – the key of the TaggedValue.

Semantics

member:NamedValue [0..*] {ordered} - NamedValues owned by this NamedValue

ownedNamedValue [0..*] {subsets ownedMember} - NamedValues owned by this NamedValue

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

9.10 Group

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

Supertype BaseElement, SACMModel

SACMElement

ownedSACMElement [0..*] {subsets ownedMember, subsets member} - SACMElements that are owned by this Group

Association End

element:SACMElement [0..*] {ordered} – a collection of SACMElements that comprise a group

member:SACMElement [0..*] {redefines element, ordered} - a collection of SACMElements that compromise a group

~~9.11~~ 9.10 IRI

IRI is a international resource identifier is a String and is formatted according to RFC 3987.

9.12 9.11 SACMModel

An abstract subtype of ModelElement which allows one to define a model and/or a pattern.

Supertype

String

Supertype ModelElement

Foundation::String

Attributes

isPattern : Boolean [0..1] = false - if true elements in SACMModel are part of a pattern.

isModel : Boolean [0..1] = false - if true elements in SACMModel are part of a model.

{ordered}

Association

+subModel:SACMModel [0..*] - subModel is a recursive containment of elements in the model.

+elements:SACMElement [0..*] - elements are items in the model.

9.13 SACMDependency

SACMDependency provides a dependency relationship between two SACMElements. The source of the dependency is called the client and the target is the supplier. The base semantics of the relationship is that if the supplier changes, the client may need to change.

Superclass

Foundation::DirectedRelationship

AssociationEnds

supplier:SACMElement [1] {redefines target}

client:SACMElement [1] {redefines source}

9.14 StringNamedValue

StringedNamedValue is a NamedValue whose value is of type String

Superclass

NamedValue

Attributes

value:ExpresionLangString [0..*] {redefines value, ordered, nonunique}