AssetCitation

10.10 TerminologyAssetCitation ... 27
Part II – Argumentation Metamodel ... 29
11 SACM Argumentation Metamodel ... 31
11.1 General ... 31
11.2 Argumentation Class Diagram ... 31
11.2.1 ArgumentationElement class (abstract) ... 32
11.2.2 ArgumentPackage Class ... 32
11.2.3 ArgumentPackageCitation Class ... 33
11.2.4 ArgumentPackageBinding Class ... 33
11.2.5 ArgumentPackageInterface Class ... 34
11.2.6 ArgumentAsset Class (abstract) ... 34
11.2.7 Assertion Class (abstract) ... 35
11.2.8 Artefact~~ElementCitation~~ Class ... 35
11.2.9 ArgumentAssetCitation Class ... 35
11.2.10 Claim Class ... 36
11.2.11 ArgumentReasoning Class ... 37
11.2.12 AssertedRelationship Class (abstract) ... 37
11.2.13 AssertedInference Class ... 38
11.2.14 AssertedEvidence Class ... 38
11.2.15 AssertedChallenge Class ... 39
11.2.16 AssertedCounterEvidence Class ... 39
11.2.17 AssertedContext Class ... 40
Part III – Artefact Metamodel ... 41
12 Artefact Classes ... 43
12.1 General ... 43
12.2 ArtefactPackageCitation ... 44
12.3 ArtefactPackageBinding ... 44
12.4 ArtefactPackageInterface ... 45
12.5 ArtefactAsset class (abstract) ... 46
12.5.1 ArtefactAssetCitation class ... 46
12.5.2 Artefact class ... 46
12.5.3 ArtefactProperty class ... 47
12.5.4 ArtefactEvent class ... 47
12.5.5 Resource class ... 48
12.5.6 Activity class ... 48
12.5.7 Technique class ... 49
12.5.8 Participant class ... 49
12.5.9 ArtefactAssetRelationship class ... 49
12.5.10 ArtefactRelationship class ... 50
12.5.11 ActivityRelationship class ... 50
12.5.12 ArtefactActivityRelationship class ... 50
12.5.13 ArtefactTechniqueRelationship class ... 51
12.5.14 ParticipantRoleRelationship class ... 51
12.5.15 ArtefactResourceRelationship class ... 52

Annex A – Mappings from existing industrial notations for assurance cases ... 53

Annex B – Examples of Assurance Cases in SACM 2.0 XMI ... 55

2 Conformance

2.1 Introduction

The Structured Assurance Case Metamodel (SACM) specification defines the following three compliance points:

- Argumentation
- Artefact Model
- Assurance Case

AssetCitation

2.2 Argumentation compliance point

Software that conforms to the SACM specification at the Argumentation compliance point shall be able to import and export XMI documents that conform with the SACM XML Schema produced by applying XMI rules to the normative MOF metamodel defined in the Argumentation subpackage of the SACM specification, including the common elements defined in the Common and Predefined diagrams of the SACM. The top object of the Argumentation package as a unit of interchange shall be the Argumentation::ArgumentPackage element of the SACM.

Conformance to the Argumentation compliance point does not entail support for the Evidence subpackage of SACM, or the terminology sub package of the SACM. The ‘Artefact~~ElementCitation~~’ class shall not be used.

This compliance point facilitates interchange of the structured argumentation documents produced by existing tools supporting existing structured argument notations such as the Goal Structuring Notation (GSN) and the Claims-Arguments-Evidence (CAE) notation which provide their own mapping onto SACM argumentation aspects. Further details of these mappings are given in Annex A.

2.3 Artefact model compliance point

Software that conforms to the specification at the Artefact Model compliance point shall be able to import and export XMI documents that conform with the SACM XML Schema produced by applying XMI rules to the normative MOF metamodel defined in this Artefact subpackage of the SACM specification, including the common elements defined in the Common and Predefined diagrams of the SACM. The top object of the Evidence package as a unit of interchange shall be the ArtefactModel::ArtefactPackage element of the SACM.

Conformance to the Artefact model compliance point does not entail support for the Argumentation subpackage of SACM, or the terminology diagram of the SACM. This compliance point facilitates interchange of the packages of evidence. In particular, this compliance point facilitates development of evidence repositories in support of software assurance and regulatory compliance.

2.4 Assurance Case compliance point

Software that conforms to the specification at the Assurance Case compliance point shall be able to import and export XMI documents that conform with the SACM XML Schema produced by applying XMI rules to the normative MOF metamodel defined in this entire specification. The top object of the Assurance Case package as a unit of interchange shall be the SACM::AssuranceCasePackage element.

The Conformance clause identifies which clauses of the specification are mandatory (or conditionally mandatory) and which are optional in order for an implementation to claim conformance to the specification.

10.9 Term

The Term class is used to model both abstract and concrete terms in SACM. Abstract Terms can be considered placeholders for concrete terms and are denoted by the inherited isAbstract attribute being set true. A concrete term is denoted by isAbstract being false.

Attributes

value: String – An attribute recording the value of the Term

externalReference: String – An attribute recording an external reference (e.g., URI) to the object referred to by the Term

Superclass

ExpressionElement

Semantics

Term class is used to model both abstract and concrete terms in SACM. Abstract Terms can be considered placeholders for concrete terms and are denoted by the inherited isAbstract attribute being set true. A concrete term is denoted by isAbstract being false.

The externalReference attribute enables the referencing of the object signified by the term (signifier). It also provides a mechanism whereby terms can reference concepts and terms defined in other ontology and terminology models.

10.10 TerminologyAssetCitation

The TerminologyAssetCitation is a citation (reference) to an ExpressionElement contained in another TerminologyPackage.

Superclass

ExpressionElement

Associations

cited~~Element~~:TerminologyAsset [1] The TerminologyAsset being cited by the TerminologyAssetCitation.

Asset

Semantics

TerminologyAssetCitations make it possible to cite TerminologyAssets from other TerminologyPackages when forming TerminologyPackages or Expressions.

For example, within a TerminologyPackage it can be useful to refer to TerminologyAssets within another TerminologyPackage (to reference terminology) that are not contained with the same TerminologyPackage and is defined elsewhere. Within an Expression it can also be useful to refer to TerminologyAssets within another TerminologyPackage that are not contained with the same TerminologyPackage and is defined elsewhere.

Constraints

The cited~~Element~~ referred to by a TerminologyAssetCitation must be outside of the containment hierarchy containing the citation.

Asset

11 SACM Argumentation Metamodel

11.1 General

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

11.2 Argumentation Class Diagram

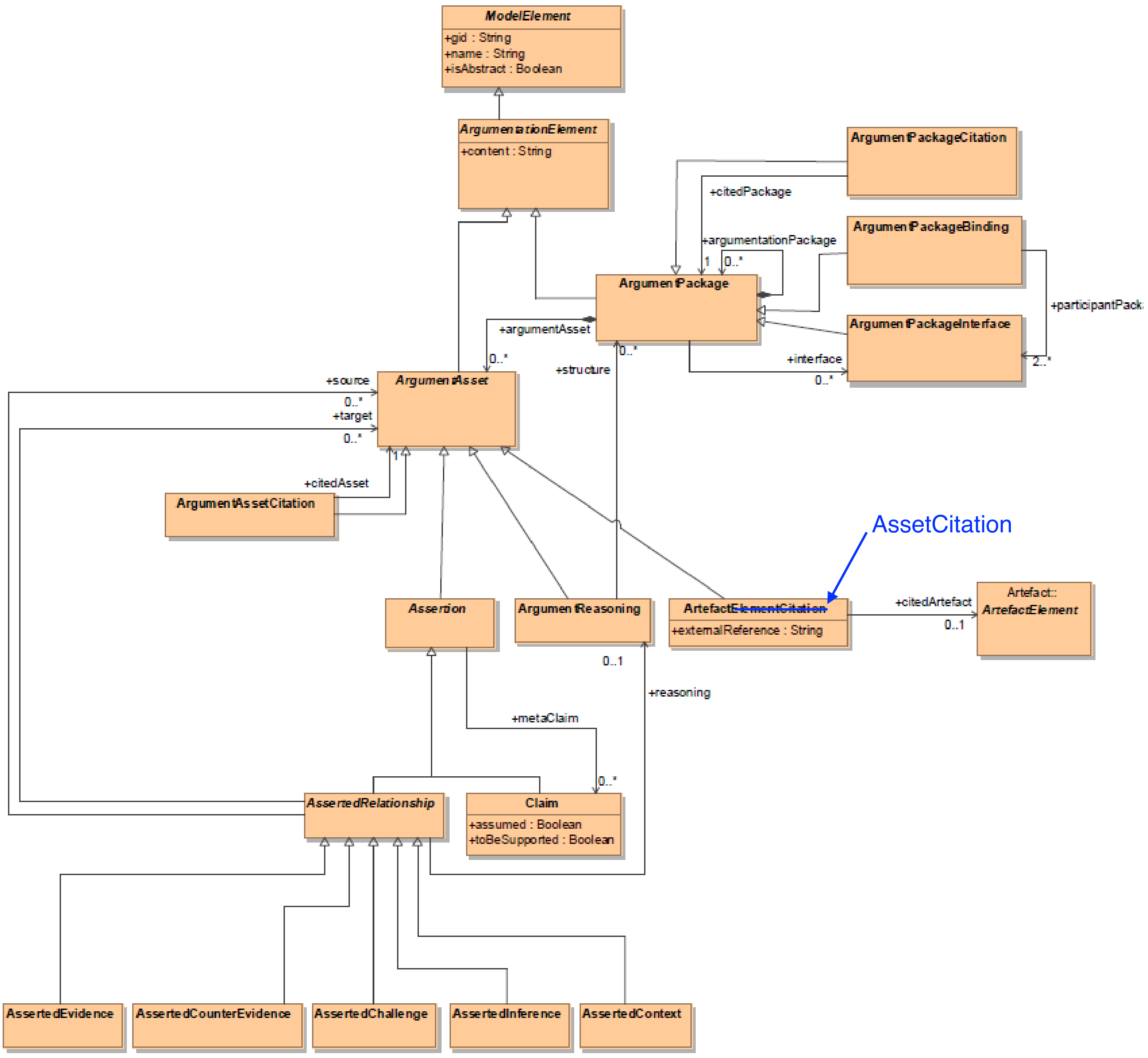

Figure 6 – Argumentation Class Diagram

AssetCitation

This portion of the SACM model describes and defines the concepts required to model structured arguments. Arguments are represented in SACM through explicitly representing the Claims and citation of artefacts (e.g., as evidence) (Artefact~~ElementCitation~~), and the 'links' between these elements – e.g., how one or more Claims are asserted to infer another Claim, or how one or more artefacts are asserted as providing evidence for a Claim (AssertedEvidence). In addition to these core elements, in SACM it is possible to provide additional description of the ArgumentReasoning associated with inferential and evidential relationships, represent counter-arguments (through AssertedChallenge), counter-evidence (through AssertedCounterEvidence), and represent how artefacts provide the context in which arguments should be interpreted (through AssertedContext.)

The packaging of structured arguments into 'modular' argument packages is enabled through ArgumentPackages, an optional declaration of an interface for the package (ArgumentPackageInterface) that cites a specific selection of the ArgumentElements contained within the package, and the ability to link (by means of an argument) two or more argument packages (through an ArgumentPackageBinding). It is also possible within a package to cite elements contained within other argument packages (through using Argument~~ElementCitation~~).

In the following sections we describe these model elements in detail.

11.2.1 ArgumentationElement class (abstract)

An ArgumentationElement is the top level element of the hierarchy for argumentation elements.

Semantics

The ArgumentationElement is a common class for all elements within a structured argument.

11.2.2 ArgumentPackage Class

The ArgumentPackage Class is the container class for a structured argument represented using the SACM Argumentation Metamodel.

Superclass

ArgumentationElement

Associations

argumentAsset:ArgumentAsset[0..*]

The ArgumentAssets contained in a given instance of an ArgumentPackage.
argumentPackage:ArgumentationPackage[0..*]

The nested argumentPackage contained in a given instance of an ArgumentPackage
interface:ArgumentationPackage[0..*]

Reference to the declared interface for the ArgumentPackage.

Semantics

ArgumentPackages contain structured arguments. These arguments are composed of ArgumentAssets. ArgumentPackages elements can be nested, and can contain citations (references) to other ArgumentPackages.

For example, one ArgumentPackage may contain a claim that is toBeSupported (i.e. currently has no supporting argument). An ArgumentPackageBinding can be used to record the mapping (by means of containing a structured argument linking ArgumentAssetCitations to the claims in question) between this claim and a supporting claim in another ArgumentPackage.

An ArgumentPackageInterface is a sub type of ArgumentPackage that can be used to create an explicit interface to an existing ArgumentPackage.

Constraints

The ‘root’ ArgumentAssets contained by an ArgumentPackageBinding (i.e. the ArgumentAssets only associated as target of an AssertedRelationship) and ‘leaf’ ArgumentAssets (i.e. the ArgumentAssets only associated as source of an AssertedRelationship) must be ArgumentAssetCitations to Claims or Artefact~~ElementCitations~~AssetCitations contained within the ArgumentPackages associated by the participantPackage association.

11.2.5 ArgumentPackageInterface Class

Superclass

ArgumentPackage

Semantics

ArgumentPackageInterfaces can be used to declare (by means of containing ArgumentAssetCitations) 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.

Constraints

ArgumentPackageInterfaces are only allowed to contain ArgumentAssetCitations to ArgumentAssets within the ArgumentPackage with which this ArgumentPackageInterface is associated (by the interface association).

11.2.6 ArgumentAsset Class (abstract)

The ArgumentAsset Class is the abstract class for the elements of any structured argument represented in SACM.

Superclass

ArgumentationElement

Semantics

ArgumentAssets represent the constituent building blocks of any structured argument contained in an ArgumentPackage.

For example, ArgumentAssets can represent the Claims made within a structured argument contained in an ArgumentPackage.

11.2.7 Assertion Class (abstract)

Assertions are used to record the propositions of Argumentation (including both the Claims about the subject of the argument and the structure of the Argumentation being asserted). Propositions can be true or false, but cannot be true and false simultaneously.

Associations

metaClaim:Claim[0..*]

references Claims concerning (i.e., about) the Assertion (e.g., regarding the confidence in the Assertion)

Semantics

Structured arguments are declared by stating claims, citing evidence and contextual information, and asserting how these elements relate to each other.

11.2.8 Artefact~~ElementCitation~~ Class

The Artefact~~ElementCitation~~ Class enables the citation of an artefact that relates to the structured argument.

Superclass

ArgumentAsset

AssetCitation

Attributes

externalReference: String An attribute recording a URL to external evidence.

Associations

citedArtefact:ArtefactElement[0..1]

The ArtefactElements cited by the current Artefact~~ElementCitation~~ object.

AssetCitations

Semantics

It is necessary to be able to cite artefacts that provide supporting evidence, context, or additional description for the core reasoning of the recorded argument. Artefact~~ElementCitations~~ allow there to be an objectified citation of this information within the structured argument, thereby allowing the relationship between this artefact and the argument to also be explicitly declared.

The externalReference attribute can be used when wishing to cite an Artefact not being modeled by an SACM ArtefactElement.

11.2.9 ArgumentAssetCitation Class

The ArgumentAssetCitation cites an ArgumentAsset within another ArgumentPackage, for use within the current ArgumentPackage.

A Claim that is intentionally declared as requiring further evidence or argumentation can be denoted by setting toBeSupported to be true.

Constraints

Self.assumed and self.toBeSupported cannot both be true simultaneously.

11.2.11 ArgumentReasoning Class

ArgumentReasoning can be used to provide additional description or explanation of the asserted inference or challenge that connects one or more Claims (premises) to another Claim (conclusion). ArgumentReasoning elements are therefore related to AssertedInferences and AssertedChallenges. It is also possible that ArgumentReasoning elements can refer to other structured Arguments as a means of documenting the detail of the argument that establishes the asserted inferences.

Superclass

ReasoningElement

Associations

structure:ArgumentPackage[0..1]

Optional reference to another the ArgumentPackage that provides the detailed structure of the argument being described by the ArgumentReasoning.

AssetCitation

Semantics

The AssertedRelationship that relates one or more Claims (premises) to another Claim (conclusion), or evidence cited by an Artefact~~ElementCitation~~ to a Claim, may not always be obvious. In such cases ArgumentReasoning can be used to provide further description of the reasoning involved.

11.2.12 AssertedRelationship Class (abstract)

The AssertedRelationship Class is the abstract association class that enables the ArgumentAssets of any structured argument to be linked together. The linking together of ArgumentAssets allows a user to declare the relationship that they assert to hold between these elements.

Superclass

Assertion

Associations

source:ArgumentAsset[0..*]

Reference to the ArgumentAsset(s) that are the source (start-point) of the relationship.

target:ArgumentAsset[0..*]

Reference to the ArgumentAsset(s) that are the target (end-point) of the relationship.

reasoning:ArgumentReasoning[0..*]

Reference to the ArgumentReasoning being described by the ArgumentReasoning.

Semantics

In SACM, the structure of an argument is declared through the linking together of primitive ArgumentAssets. For example, a sufficient inference can be asserted to exist between two claims (“Claim A implies Claim B”) or sufficient evidence can be asserted to exist to support a claim (“Claim A is evidenced by Evidence B”). An inference asserted between two claims (A – the source – and B – the target) denotes that the truth of Claim A is said to infer the truth of Claim B.

11.2.13 AssertedInference Class

The AssertedInference association class records the inference that a user declares to exist between one or more Assertion (premises) and another Assertion (conclusion). It is important to note that such a declaration is itself an assertion on behalf of the user.

Superclass

AssertedRelationship

Semantics

The core structure of an argument is declared through the inferences that are asserted to exist between Assertions (e.g., Claims). For example, an AssertedInference can be said to exist between two claims (“Claim A implies Claim B”). An AssertedInference between two claims (A – the source – and B – the target) denotes that the truth of Claim A is said to infer the truth of Claim B.

Constraints

The source of AssertedInference relationships must be Claims, or Argument~~ElementCitations~~ that cite a Claim.

The target of AssertedInference relationships must be Assertions, or Argument~~ElementCitations~~ that cite an Assertion.

11.2.14 AssertedEvidence Class

AssetCitations

The AssertedEvidence association class records the declaration that one or more artefacts of Evidence (cited by Artefact~~ElementCitations~~) provide information that helps establish the truth of a Claim. It is important to note that such a declaration is itself an assertion on behalf of the user. The artefact (cited by an Artefact~~ElementCitation~~) may provide evidence for more than one Claim.

Superclass

AssertedRelationship

AssetCitation

Semantics

Where evidence (cited by Artefact~~ElementCitation~~) exists that helps to establish the truth of a Claim in the argument, this relationship between the Claim and the evidence can be asserted by an AssertedEvidence association. An AssertedEvidence association between an artefact cited by an Artefact~~ElementCitation~~ and a Claim (A – the source evidence cited – and B – the target claim) denotes that the evidence cited by A is said to help establish the truth of Claim B.

AssetCitation

Constraints

The source of AssertedEvidence relationships must be Artefact~~ElementCitation~~.

The target of AssertedEvidence relationships must be Assertions, or Argument~~ElementCitations~~ that cite an Assertion.

11.2.15 AssertedChallenge Class

The AssertedChallenge association class records the challenge (i.e. counter-argument) that a user declares to exist between one or more Claims and another Claim. It is important to note that such a declaration is itself an assertion on behalf of the user.

Superclass

AssetCitations

AssertedRelationship

Semantics

An AssertedChallenge by Claim A (source) to Claim B (target) denotes that the truth of Claim A challenges the truth of Claim B (i.e., Claim A leads towards the conclusion that Claim B is false).

Constraints

The source of AssertedChallenge relationships must be Claims, or Argument~~ElementCitations~~ that cite a Claim.

The target of AssertedChallenge relationships must be Assertions, or Argument~~ElementCitations~~ that cite an Assertion.

11.2.16 AssertedCounterEvidence Class

AssertedCounterEvidence can be used to associate evidence (cited by Artefact~~ElementCitations~~) to a Claim, where this evidence is being asserted to infer that the Claim is false. It is important to note that such a declaration is itself an assertion on behalf of the user.

Superclass

AssertedRelationship

Semantics

An AssertedCounterEvidence association between some evidence cited by an InformationNode and a Claim (A – the source evidence cited – and B – the target claim) denotes that the evidence cited by A is counter-evidence to the truth of Claim B (i.e., Evidence A suggests the conclusion that Claim B is false).

AssetCitation

Constraints

The source of AssertedCounterEvidence relationships must be Artefact~~ElementCitation~~.

The target of AssertedCounterEvidence relationships must be Assertions, or Argument~~ElementCitations~~ that cite an Assertion.

AssetCitations

11.2.17 AssertedContext Class

The AssertedContext association class can be used to declare that the artefact cited by an Artefact~~ElementCitation~~(s) provides the context for the interpretation and scoping of a Claim or ArgumentReasoning element. In addition, the AssertedContext association class can be used to declare a Claim asserted as necessary context (i.e. a precondition) for another Assertion or ArgumentReasoning.

Superclass

AssertedRelationship

Semantics

Contextual information often needs to be cited in order to make clear the interpretation and scope of a Claim or ArgumentReasoning description. For example, a Claim can be said to be valid only in a defined context ("Claim A is asserted to be true only in a context as defined by the information cited by Artefact B" or conversely "InformationItem B is the asserted context for Claim A"). A declaration (AssertedContext) of context (Artefact~~ElementCitation~~ B) for a ReasoningElement A records that B is asserted to be contextual information required for the interpretation and scoping of A (i.e., B defines the context where the reasoning presented by A is asserted as true).

Contextual Claims often need to be cited as preconditions for a Claim or ArgumentReasoning. For example, a Claim may be asserted only in the context of another claim ("Claim A is asserted to be true only in a context where Claim B is true". Similarly, a description of ArgumentReasoning A may only be considered true in a context where Claim B is true".

Constraints

The source of AssertedContext relationships must be Artefact~~ElementCitations~~ or Claims.

The target of AssertedContext relationships must be Assertions, Argument~~ElementCitations~~ that cite an Assertion "ArgumentReasoning" elements or Argument~~ElementCitations~~ that cite ArgumentReasoning elements.