Date: June 2012

Structured Assurance Case Metamodel (SACM)

FTF - Convenience Document Beta 2

OMG Document Number: ptc/2012-06-06
Standard document URL: http://www.omg.org/spec/SACM
Associated Schema Files:

ptc/2012-05-10 -- http://www.omg.org/spec/SACM/20120501/SACM.emof
ptc/2012-05-11 -- http://www.omg.org/spec/SACM/20120501/SACM.ecore
ptc/2012-06-08 -- http://www.omg.org/spec/SACM/20120501/SACM.xsd
ptc/2012-06-09 -- http://www.omg.org/spec/SACM/20120501/Argumentation.xsd
ptc/2012-06-10 -- http://www.omg.org/spec/SACM/20120501/Evidence.xsd

---

This OMG document replaces the individual adopted specifications (ptc/2010-08-36, ARM, Beta 1 and ptc/2010-08-37, SAEM, Beta 1). It is an OMG Adopted Beta Specification and is currently in the finalization phase. Comments on the content of this document are welcome, and should be directed to *issues@omg.org* by February 1, 2011.

You may view the pending issues for this specification from the OMG revision issues web page *http://www.omg.org/issues/*.

The FTF Recommendation and Report for this specification will be published on July 24, 2012. If you are reading this after that date, please download the available specification from the OMG Specifications Catalog.

USE OF SPECIFICATION - TERMS, CONDITIONS & NOTICES

The material in this document details an Object Management Group specification in accordance with the terms, conditions and notices set forth below. This document does not represent a commitment to implement any portion of this specification in any company's products. The information contained in this document is subject to change without notice.

LICENSES

The companies listed above have granted to the Object Management Group, Inc. (OMG) a nonexclusive, royalty-free, paid up, worldwide license to copy and distribute this document and to modify this document and distribute copies of the modified version. Each of the copyright holders listed above has agreed that no person shall be deemed to have infringed the copyright in the included material of any such copyright holder by reason of having used the specification set forth herein or having conformed any computer software to the specification.

Subject to all of the terms and conditions below, the owners of the copyright in this specification hereby grant you a fully-paid up, non-exclusive, nontransferable, perpetual, worldwide license (without the right to sublicense), to use this specification to create and distribute software and special purpose specifications that are based upon this specification, and to use, copy, and distribute this specification as provided under the Copyright Act; provided that: (1) both the copyright notice identified above and this permission notice appear on any copies of this specification; (2) the use of the specifications is for informational purposes and will not be copied or posted on any network computer or broadcast in any media and will not be otherwise resold or transferred for commercial purposes; and (3) no modifications are made to this specification. This limited permission automatically terminates without notice if you breach any of these terms or conditions. Upon termination, you will destroy immediately any copies of the specifications in your possession or control.

PATENTS

The attention of adopters is directed to the possibility that compliance with or adoption of OMG specifications may require use of an invention covered by patent rights. OMG shall not be responsible for identifying patents for which a license may be required by any OMG specification, or for conducting legal inquiries into the legal validity or scope of those patents that are brought to its attention. OMG specifications are prospective and advisory only. Prospective users are responsible for protecting themselves against liability for infringement of patents.

GENERAL USE RESTRICTIONS

DISCLAIMER OF WARRANTY

RESTRICTED RIGHTS LEGEND

TRADEMARKS

COMPLIANCE

The copyright holders listed above acknowledge that the Object Management Group (acting itself or through its designees) is and shall at all times be the sole entity that may authorize developers, suppliers and sellers of computer software to use certification marks, trademarks or other special designations to indicate compliance with these materials.

Software developed under the terms of this license may claim compliance or conformance with this specification if and only if the software compliance is of a nature fully matching the applicable compliance points as stated in the specification. Software developed only partially matching the applicable compliance points may claim only that the software was based on this specification, but may not claim compliance or conformance with this specification. In the event that testing suites are implemented or approved by Object Management Group, Inc., software developed using this specification may claim compliance or conformance with the specification only if the software satisfactorily completes the testing suites.

OMG's Issue Reporting Procedure

All OMG specifications are subject to continuous review and improvement. As part of this process we encourage readers to report any ambiguities, inconsistencies, or inaccuracies they may find by completing the Issue Reporting Form listed on the main web page *http://www.omg.org*, under Documents, Report a Bug/Issue (http://www.omg.org/technology/agreement.htm).

Table of Contents

Preface

About the Object Management Group

OMG

Founded in 1989, the Object Management Group, Inc. (OMG) is an open membership, not-for-profit computer industry standards consortium that produces and maintains computer industry specifications for interoperable, portable and reusable enterprise applications in distributed, heterogeneous environments. Membership includes Information Technology vendors, end users, government agencies and academia.

OMG member companies write, adopt, and maintain its specifications following a mature, open process. OMG's specifications implement the Model Driven Architecture® (MDA®), maximizing ROI through a full-lifecycle approach to enterprise integration that covers multiple operating systems, programming languages, middleware and networking infrastructures, and software development environments. OMG's specifications include: UML® (Unified Modeling Language™); CORBA® (Common Object Request Broker Architecture); CWM™ (Common Warehouse Metamodel); and industry-specific standards for dozens of vertical markets.

More information on the OMG is available at *http://www.omg.org/*.

OMG Specifications

As noted, OMG specifications address middleware, modeling and vertical domain frameworks. A catalog of all OMG Specifications is available from the OMG website at:

http://www.omg.org/technology/documents/spec_catalog.htm

Specifications within the Catalog are organized by the following categories:

Business Modeling Specifications

- Business Rules and Process Management Specifications

Language Mappings

- IDL/Language Mapping Specifications
- Other Language Mapping Specifications

Middleware Specifications

- CORBA/IIOP
- CORBA Component Model
- Data Distribution
- Specialized CORBA

Modeling and Metadata Specifications

- UML
- MOF
- XMI
- CWM
- Profile specifications.

Modernization Specifications

- KDM

Platform Independent Model (PIM), Platform Specific Model (PSM), and Interface Specifications

- CORBAservices
- CORBAfacilities
- OMG Domain specifications
- OMG Embedded Intelligence specifications
- OMG Security specifications

All of OMG's formal specifications may be downloaded without charge from our website. (Products implementing OMG specifications are available from individual suppliers.) All specifications are available in PostScript and PDF format and may be obtained from the Specifications Catalog cited above. Certain OMG specifications are also available as ISO standards. Please consult *http://www.iso.org*

OMG Contact Information

OMG Headquarters
140 Kendrick Street
Building A, Suite 300
Needham, MA 02494
USA
Tel: +1-781-444-0404
Fax: +1-781-444-0320
http://www.omg.org/
Email: *pubs@omg.org*

Typographical Conventions

The type styles shown below are used in this document to distinguish programming statements from ordinary English. However, these conventions are not used in tables or section headings where no distinction is necessary.

Times/Times New Roman - 10 pt.: Standard body text

Helvetica/Arial - 10 pt. Bold: OMG Interface Definition Language (OMG IDL) and syntax elements.

`Courier - 10 pt. Bold:` Programming language elements.

Helvetica/Arial - 10 pt: Exceptions

Note – Terms that appear in *italics* are defined in the glossary. Italic text also represents the name of a document, specification, or other publication.

Issues

The reader is encouraged to report any technical or editing issues/problems with this specification to *http://www.omg.org/technology/agreement.htm*.

1 Scope

This specification defines a metamodel for representing structured assurance cases. Assurance Case is a set of auditable claims, arguments and evidence created to support the claim that a defined system/service will satisfy the particular requirements. Assurance case is a document that facilitates information exchange between suppliers and acquirers, and between the operator and regulator, where the knowledge related to the safety and security of the system is communicated in a clear and defendable way. Assurance case represents the scope of the system, the operational context, the claims, the safety and/or security arguments, along with the corresponding evidence.

Systems Assurance is the process of building clear, comprehensive and defensible arguments regarding the safety and security properties of systems. The vital element of Systems Assurance is that it makes clear and well-defined claims about the safety and security of systems. Certain claims are supported through reasoning. Reasoning is expressed by explicit annotated links between claims, where one or more claims (called sub-claims) when combined provide inferential support to a larger claim. Certain associations between claims and subclaims are justified. Justification explains the selection of argument strategy Claims are propositions which are expressed by statements in some natural language. The degree of precision in formulation of the claims may contribute to the comprehensiveness of an assurance case. The context is important to communicate the scope of the claim, and to clarify the language used by the claim by providing necessary definition and explanations. Context involves assumptions made about the system and its environment. Explicit statement of the assumptions contributes to the comprehensiveness of the argument. Argumentation flow between claims is structured to facilitate communication of the entire assurance case.

1.1 Structured Arguments

Part of this specification defines a metamodel for representing structured arguments. A convincing and valid argument that a system meets its assurance requirements is at the heart of an assurance case, which also may contain extensive references to evidence. The Argumentation Metamodel facilitates projects by allowing them to effectively and succinctly communicate in a structured way how their systems and services are meeting their assurance requirements. The scope of the Argumentation Metamodel is therefore to allow the interchange of structured arguments between diverse tools by different vendors. Each Argumentation Metamodel instance represents the argument that is being asserted by the stakeholder that is offering the argument for consideration.

This specification is designed to stand alone, or may be ued in combination with the SACM Evidence Metamodel. The Evidence Metamodel is designed to represent aspects of evidence and properties about evidence in further detail. In this the Argumentation Metamodel we have a simplified support to model the relation of evidence to a structured argument.

Standardization will ensure that end users are investing not just in individual tools but also rather into a coordinated strategy.

The metamodel for argumentation provides a common structure and interchange format that facilitates the exchange of system assurance arguments contained within individual tool models. The metamodel represents the core concepts for structured argumentation that underlie a number of existing argumentation notations.

1.2 Evidence

Part of this specification provides a metamodel for collecting, developing, evaluating, communicating, and managing Evidence (referred as the SACM Evidence Metamodel). Specifically, this Evidence Metamodel does all of the following:

- Identifies the main factors that determine the evidence collection process.
- Identifies the main factors that determine the evaluation of evidence.

- Identifies and defines the elements of evidence.
- Defines a common interchange format to facilitates the exchange of information between different Software Assurance tools and services.

The SACM Evidence Metamodel defines a catalog of elements for constructing and interchangins precises statements related to evidence in support of various assurance efforts. This speficication facilitates development of new type of Assurance tools related to assurance of safety and security of software-intensive systems, and automation of the processes of regulatory compliance and risk assessments.

The SACM Evidence Metamodel provides the basis for logical design of easily-constructed tools for storing, managing, cross-referencing, evaluating and reporting the elements of evidence during assurance efforts.

An assurance case is a collection of auditable claims, arguments and evidence created to support the contention that a defined system/service will satisfy the particular requirements.

Certain claims are supported through evidence, i.e., rely on external documented facts to confer evidentiary support. Evidence is collected by applying systematic methods and procedures and is often collected by automated tools.

Evidence is information, based on established fact or expert judgment, which is presented to show that the claim to which it relates is valid (i.e., true). Anything that supports the Claim can be presented as evidence. Often, this information is a record of some sort, demonstrating that a certain event took place. Evidence can be diverse as various things may be produced as evidence, such as documents, expert testimony, test results, measurement results, records related to process, product, and people, etc.

The following characteristics are usually attributed to evidence:

- Direct or indirect evidence. These characteristics refer to the nature of support provided by evidence item to the corresponding claim. To be considered “direct evidence,” it must be sufficient on its own to make a statement without the necessity of introducing other records. Direct evidence specifically makes a statement. Indirect evidence (or circumstantial evidence as it is often called) requires introduction of other pieces of information to complete a statement. Direct evidence has more weight than indirect. Whenever additional records are drawn to supply missing information there is a chance for error. Because of that, less weight is assigned to indirect evidence. Additionally, the source of evidence can be weighted.
- Primary or secondary information. These characteristics refer to the quality of information provided as evidence. The record is primary if it was made at or near the time of the event, by someone in a position to know firsthand (such as an eyewitness). Alternatively, a record is considered primary if it was made in writing by an officer charged by law, canon, or bylaws with creating an accurate record. Primary information carries more weight than secondary information. Various communities disagree on whether primary information remains primary when copied. For example the legal community states that a primary record becomes secondary when copied. Other communities focus at the information rather than the record, from which standpoint the primary information remains primary when copied.
- Original or derived source. These characteristics refer to the document (record) that is the source of evidence. The original source is one that contributes written, oral, or visual information not derived from a prior written or visual record or oral communication. A derivative source is one that contributes information that was copied, transcribed, abstracted, summarized, duplicated or repeated from information is a previously existing source (that is from the original or another derivative).

2 Conformance

Structured Assurance Case Metamodel (SACM) specification defines the following 3 compliance points:

- Argumentation
- Evidence Container
- Assurance Case

2.1 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::Argumentation element of the SACM.

Conformance to the Argumentation compliance point does not entail support for the Evidence subpackage of SACM, or the Administration diagram of the SACM. Links to the evidence items in the Argumentation::InformationElement shall be made using the ‘url’ attribute. The ‘evidence’ association shall not be used.

This compliance point facilitates interchange of the structured argumentation documents produced by existing tools supporting The Goal Structuring Notation (GSN) and Claims-Arguments-Evidence (CAE) notation. Examples of the SACM XML interchange documents and the corresponding GSN and CAE diagrams are provided in Annex B.

2.2 Evidence Container compliance point

Software that conforms to the specification at the Evidence Container 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 Evidence 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 Evidence::EvidenceContainer element of the SACM.

Conformance to the Evidence compliance point does not entail support for the Argumentation subpackage of SACM, or the Administration diagram of the SACM. Claims in the Evidence::ReferencedClaim element shall be explicitly defined using the ‘content’ attribute of the Evidence::ReferencedClaim element. The ‘claim’ association shall not be used.

This compliance point facilitates interchange of the precise statements related to evidence. In particular, this compliance point facilitates development of evidence repositories in support of software assurance and regulatory compliance.

2.3 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::AssuranceCase element.

3 Normative References

The following normative documents contain provisions which, through reference in this text, constitute provisions of this specification. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply.

- OMG UML 2.2 Infrastructure Specification formal/2009-02-04
- OMG Meta-Object Facility (MOF) ver. 2.0 formal/2006-01-01
- OMG MOF XML Metadata Interchange (XMI) Specification, ver. 2.1, formal/05-09-01
- OMG Semantics of Business Vocabularies and Business Rules (SBVR) Specification, ver. 1.0 formal/08-01-02
- ISO/IEC 15026 Systems and software engineering - Systems and software assurance - Part 1: Concepts and vocabulary, 2009
- ISO/IEC 15026 Systems and software engineering - Systems and software assurance - Part 2: Assurance case, 2009

4 Terms and Definitions

For the purposes of this specification, the terms and definitions given in the normative reference and the following apply.

Argument

A body of information presented with the intention to establish one or more claims through the presentation of related supporting claims, evidence and contextual information.

Assurance Case

A collection of auditable claims, arguments and evidence created to support the contention that a defined system/service will satisfy the particular requirements.

Claim

A proposition being asserted by the author or utterer that is a true or false statement.

Evidence

Information or objective artifacts being offered in support of one or more claims.

Evidence Item

A unique element of the body of evidence, such as an exhibit, a claim, or other element of meaning associated with an exhibit, an evidence attribute of one of the predefined relations between evidence elements representing assertions made during the evidence collection and evaluation of evidence

Evidence Repository

A software service providing access to, and information about a collection of evidence items, such as records, documents and other exhibits together with related information that facilitates management of evidence, the interpretation of evidence and understanding the evidentiary support provided to claims.

Structured argument

A particular kind of argument where the relationships between the asserted claims, and from the evidence to the claims are explicitly represented.

5 Symbols

There are no symbols defined in this specification.

6 Additional Information

6.1 Changes to Adopted OMG Specifications

None

6.2 How to Proceed

The rest of this document contains the technical content of this specification.

Chapter 7. Specification overview - Provides design rationale for the SACM Argumentation Metamodel specification.

Part 1 of the specification defines the normative common elements. Material in this part of the specification is related to all compliance points.

Chapter 8. SACM Asssurance Case defines the common elements of the Structured Assurance Case Metamodel.

Part 2 of the specification defines the SACM Argumentation metamodel. The Argumentation Metamodel defines the catalog of elements for constructing and interchaning structured statements describing argumentations. Material in this part of the specification is related to the Assurance Case and Argumentation compliance points, and it not required for the Evidence Container compliance point. This part includes a single chapter. The non-normative Annex B contains some examples of the SACM XML interchange format for Argumentation, and describes how SACM Argumentation is related to existing graphical notations for descirbing structured arguments, such as the Goal Structuring Notation (GSN) and the Claims-Arguments-Evidence (CAE) notation.

Chapter 8. The SACM Argumentation Metamodel - Provides the details of the Argumentation Metamodel specification.

Part 3 of the specification defines the SACM Evidence metamodel. The Evidence Metamodel defines the catalog of elements for constructing and interchanging precise statements involved in evidence-related efforts. The non-normative Annex A provides the SBVR vocabulary of the concepts of the SACM Evidence Metamodel. Material in this part of the specification is related to the Assurance Case and the Evidence Container compliance points, and it not required for the Argumentation compliance point. This part includes 6 chapters.

Chapter 10 defines the key elements of the Evidence metamodel.

Chapter 11 defines the statements related to the fundamental properties of the evidence items

Chapter 12 defines the formal statements for SACM.

Chapter 13 defines the statements related to the properties of evidence, including provenance, custody, timing and evidence events in the lifecycle of an evidence element.

Chapter 14 defines the statements related to the evaluation of evidence.

Chapter 15 defines the auxiliary statements involved in managing evidence-related efforts.

7 Background and Rationale

7.1 The need for assurance cases

All sectors of society are placing growing reliance on software-dependent systems, both information systems and embedded systems. Adequate functioning of many of these systems is critical to the well-being of organizations and society. Today, these numerous, large, complex systems provide increased benefits by connecting with others and generally directly or indirectly to the Internet.

However the societal and individual risks posed by attacks on, or in the maladaptive behavior of such systems are significant enough to warrant a pro-active technology adoption approach whereby the emergent risks can be analyzed, explored, communicated, and ultimately accepted by those responsible for the assurance.

Thus, software suppliers face the task of engineering their products and services to meet these challenges and threats in such a way that users and other stakeholders can rationally possess the needed confidence in them – or at least judge their level of risk. This means that suppliers must not only ensure their delivery of adequate systems, but acquirers and users require the explicit, valid, well-reasoned, and evidence-supported grounds[1] for their confidence and decision making including related engineering conclusions and their uncertainty.

Historically assurance cases covering safety and security requirements for systems have been seen as an important tool for the interchange of assurance information.

To make software assurance more practical, automation and meaningful exchange of this assurance-related information is needed. Software suppliers, tool vendors, acquirers, users, and others would benefit from a flexible and extensible means for its representation and exchange.

The concept of an assurance case is one that provides a framework for analyzing and communicating the assurance arguments and evidence that relate to a system under consideration. Suppliers and customers can see how the system lifecycle products (system requirements, design, testing, field experience, etc.) relate to and satisfy the assurance requirements, enabling sufficient confidence to be gained in the behavior and integration of the system within its operational context.

Simply put, an assurance case comprises the arguments and evidence that a system will meet its assurance requirements over its lifecycle.

7.2 Structured Arguments

Arguments have always been used – albeit informally – to communicate and persuade stakeholders that sufficient confidence can be had in a particular system. However these arguments are often spread over a range of system and management documentation, and it is difficult to see the argument as a whole in a clear way.

In the assurance domain an '**argument**' is defined as "a connected series of statements or reasons intended to establish a position…; a process of reasoning"[2]. In attempting to persuade others of a position, we cite reasons why a claim should be accepted as **true**. These reasons are described as the **premises** of the argument, and the claim they support as its **conclusion**. These terms can be used to define the 'normal form' of an argument as:

1. Suppliers also need the same or similar case to justify release and deployment.
2. *Shorter Oxford English Dictionary*, 6th Edition (2007)

Premise
Premise
Premise
So, Conclusion

This form reduces argument to its most primitive building blocks, for example:

Premise: All complex systems are susceptible to failure.
Premise: Failures can lead to accidents.
Therefore,
Conclusion: Accidents can occur in complex safety-critical systems.

The terms ‘premise’ and ‘conclusion’ are relative. The premise of one reasoning step (e.g., that “All complex systems are susceptible to failure”) may itself need further reasoning support and will become the conclusion of a subsequent supporting argument. This gives rise to hierarchical argument structures (‘chains of reasoning’) in which arguments are established by the composition of a number of (premise-conclusion) reasoning steps in order to support an overall conclusion, as illustrated in Figure 7.1.

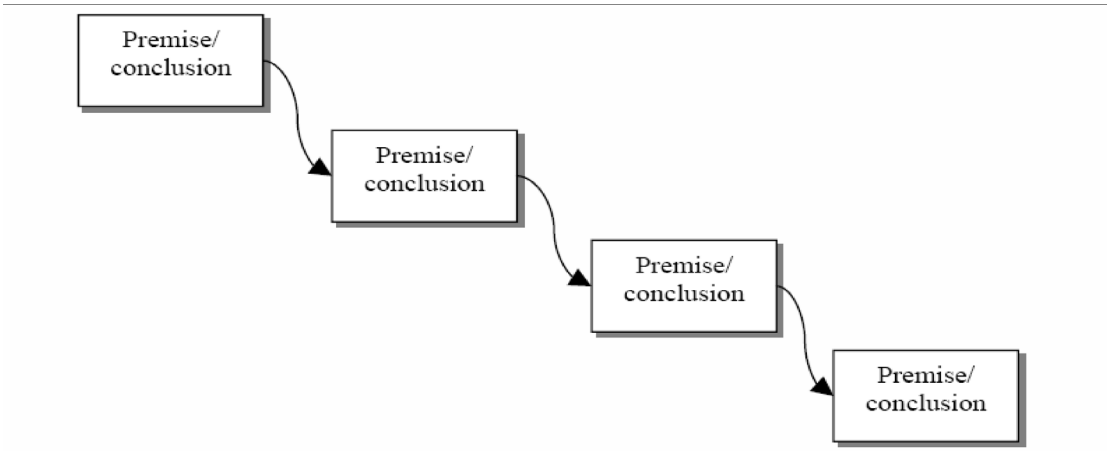

Figure 7.1 - Argument Chain Structure

Structured arguments are therefore one way to allow the communication of how a series of claims can establish a conclusion.

7.3 Arguments as asserted positions

It is important to note that the representation of an argument is not the same as a valid argument. The process of argument representation and communication is separate from that of argument evaluation. For example, an argument may include invalid reasoning, or may have a reliance on irrelevant or false information.

Therefore representations of arguments should be seen as positions that are effectively asserted by the authors or organizations that are putting forward the argument.

Clearly professional ethics require that assurance stakeholders should present arguments that they believe to be correct, valid, and relevant.

A key concept is that structured arguments allow users to express and declare what they consider the argument to be.

7.4 Structured Arguments in SACM

SACM contains those elements presented as fundamental to the expression and exchange of structured arguments.

As noted above, a typical natural language dictionary definition of an argument is that an argument comprises a series of linked premises (propositions), leading to a conclusion. From this we can derive a set of practical modeling approaches that allow users to link together propositions (claims) and to communicate how they consider that higher level claims to be supported or derived from the lower level claims. Since a claim can be used to support one or more other claims, the general form of a directed graph emerges.

SACM aims to provide a modeling framework to allow users to express and exchange their argument structures. The representation of an argument in SACM does not imply that the argument is complete, valid, or correct. Similarly, the evaluation or acceptance of an argument by a separate party is not covered by the SACM.

In the SACM model, structured arguments comprise argument elements (primarily claims) that are being asserted by the author of the argument, together with relationships that are asserted to hold between those nodes.

7.5 Precise statements related to evidence

In the simplest form, evidence consists of a collection of documents or records that provide evidentiary support to a set of claims. These claims are called subject claims, as the are made by an argument related to some selected subject area. Subject claims are different from evidence claims, which are the assertions about the evidence items that help establish the exact nature of the evidentiary support they provide to the subject claims in a clear, comprehensive and defensible way. Evidence claims can be reused as opposed to subject claims and arguments, which are specific to each subject area for which an assurance case is developed. Thus the SACM Evidence Metamodel defines the evidence vocabulary for constructing precise statements related to evidence. Evidence vocabulary is reused in every argument for various diverse subject areas.

The Evidence Metamodel defines an interchange format for evidence (XSD schema defined through the application of XMI rules defined by MOF and XMI specifications) in which each evidence element, including claims about evidence, is represented by a specific XML tags. The evidence interchange format is then utilized to exchange bodies of evidence related to specific projects that require argumentation, for example, in presenting an assurance case.

Evidence Metamodel defines the vocabulary for constructing and interchanging precise statements describing evidence-related efforts, including

- Collection of evidence
- Management of evidence
- Interpretation of evidence
- Evaluation of evidence

Collection of Evidence includes activities of identifying evidence items, and recording various information about them, including their origin, timing and custody. Evidence Metamodel defines precise statements related to the pedigree of an evidence item, including evidence collection method or tool used.

The primary items of evidence are Documents, Records, Assertions and Objects. Documents may have Properties that are characteristics independent of an assurance case being developed.

Properties in the Evidence Metamodel include the following:

- Fundamental characteristics of Documents, for example
  - Media of document
  - Language of document
  - Security classification of document
- Quality of Documents, for example
  - Primary or secondary document
  - Original or derived document
  - Consistency
  - Completeness
  - Accuracy

Management of Evidence compliments evidence collection activities with some planning and tracking activities. Important to the management of evidence is the set of Project Elements, including an Evidence Container, for grouping evidence items and assertions, as well as several elements for planning management collection Activities, including their dependencies, objectives, input and output data, and the evidence requests, which are the placeholders for evidence items that are being planned to be obtained. Combined with the evidence events, provenance, custody and timing clauses, these project elements are powerful enough to support management of evidence-related efforts and interchange of the relevant managerial data as part of evidence packages.

- Provenance of Evidence Elements, for example
  - Who created
  - Who approved
  - Who owns
- Custody of Evidence Elements, for example
  - Where the element was aquired
  - Where the element is located
  - Who is the custodian of the element
- Timing of Evidence Elements, for example
  - When the element was created or acquired
  - Effective Time of an assertion

Interpretation of Evidence includes activities of assigning meaning to documents (what a document is, what claims does it make, etc). Interpretation of evidence is an important step in legal community, when a physical object is submitted as evidence.

The following assertions are made to establish the meaning of evidence items.

Meaning Attributes of Documents, stating the Meaning of Documents

- Definition
- Meaning
- Scope
- Characteristics

Evaluation of Evidence includes the activities of making certain assertions about evidence items and their relation to subject claims.

Evidence Assertions are defined within the Evidence Metamodel and include the following categories:

- Quality Attributes of Evidentiary Support
  - Direct or indirect
  - Relevance
  - Confidence
  - Strength
  - Significance
- Nature of the Evidentiary support
  - Supports
  - Challenges
- Observations and Resolutions
  - The entire evidence package needs to be evaluated
  - Relations between Evidence Items need to satisfy one of the well-defined “Standards of proof,” such as
    - Clean and Convincing Evidence (CCE)
    - Preponderance of evidence (POE)
    - Resolved Counter Evidence (RCE), often used in the field of Genealogy as the Genealogical Proof Standard
    - Beyond the reasonable doubt (BRD)

The following diagram is related to the so-called Resolved Counter Evidence Proof Standard, which illustrates the steps involved in evaluating evidence.

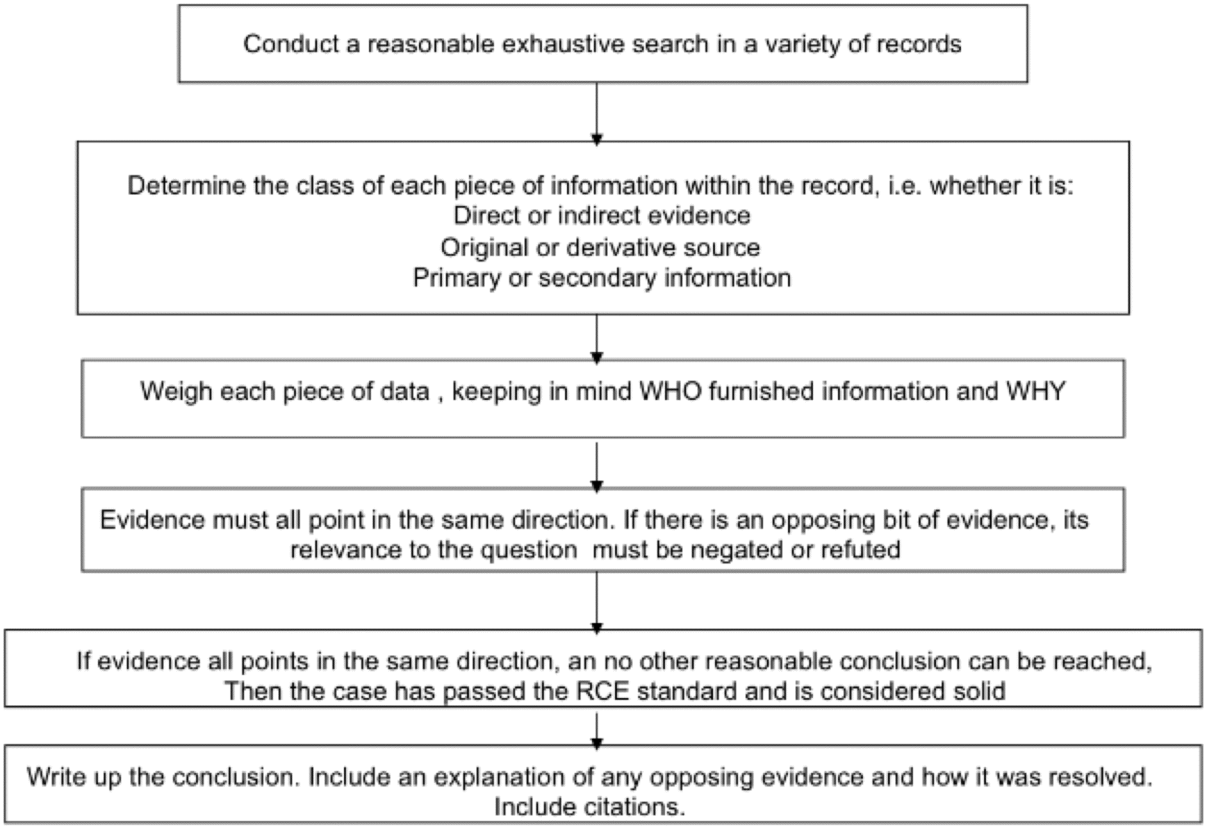

Figure 7.1 Example Evidence Evaluation Process (non-normative)

7.6 The Key Elements of Evidence

The key concept of evidence is a Document that provides evidentiary support to some Subject claim. Document is collected during the course of Evidence collection process. Usually a Document is interpreted as a description of a certain state of affairs involving several objects in the subject area (for which certain claims are being made). Subject claims are assertions related to the state of affairs in the subject area. Evidence evaluation (as opposed to Evidence collection) involves certain specific Claims about Evidence, in particular, Evidence Relation describes the nature of the evidentiary support between a Document and a Subject Claim, or the interpretation of a Document as a meaning. Evidence Relation involves certain attributes that qualify relations between Documents and Subject Claims, or Documents and meanings. Evidence Observations describe conflicts between evidence relations. Evidence Resolutions record judgments that resolve conflicts in evidence relations. Note, that Documents and Subject Claims simply exist. A Document becomes Evidence only insofar as it is claimed to provide evidentiary support to a certain Subject Claim.

7.7 The Evidence Element Lifecycle

History and custody of evidence elements including Documents, Objects, and various Assertions, as well as evidence collection Activities is represented through Provenance, Timing, and Custody properties. In a formally consistent Evidence Package, each Assertion has a timestamp and provenance, so the entire history of the evidence collection and evaluation activities can be generated. Figure 7.2 summarizes the life cycle of an Evidence Item (A Document or an Object).

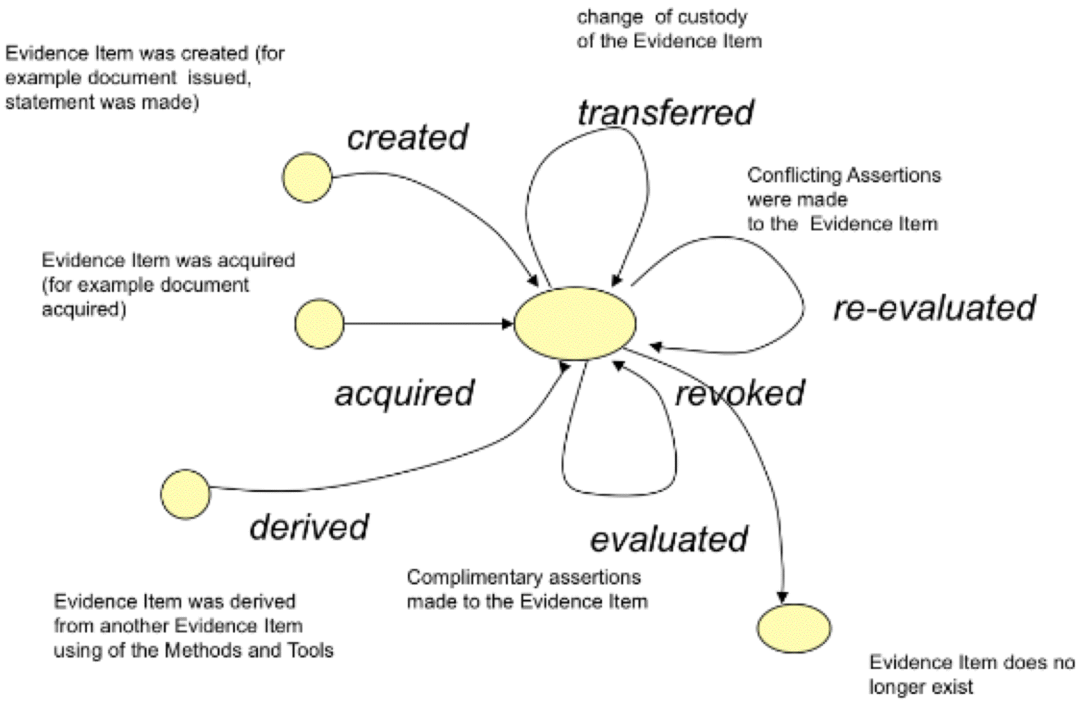

Figure 7.2 - The life-cycle of an evidence item (non-normative)

Acquisition and subsequent transfers of a Document or a Domain Object establish the so-called chain of custody, which is an important consideration of the quality of evidence in the legal community. Decision to revoke a piece of evidence can be made, making a prior acquired piece of evidence inadmissible. Any claims supported by this piece of evidence need to be identified and re-evaluated.

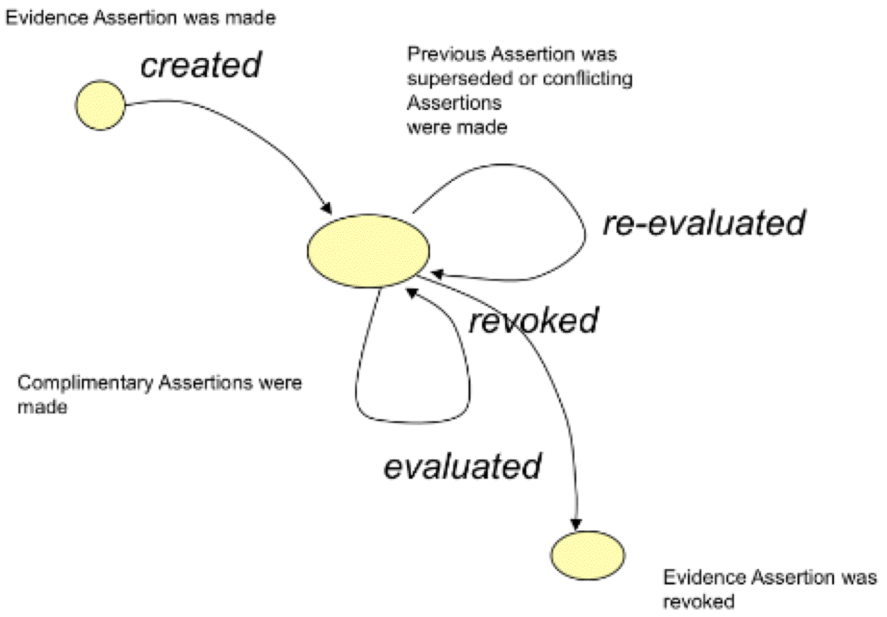

Figure 7.3 - Life-cycle of an Evidence Assertion (non-normative)

Evidence Assertions are statements related to evidence items and the evidentiary support provided by these items to various claims. Evidence Assertions have simpler life cycle, where they are created and evaluated and, possibly, re-evaluated, see Figure 7.3. Evidence Assertions cannot be acquired, derived or transferred. However Evidence Assertions can be revoked.

	Document, Exhibit	Formal Object, Formal Assertion	Evaluation
IsCreatedAt	At location By stakeholder (person) Approved by supervisor At time Effective time Owned by organization	By stakeholder Approved by supervisor At time Effective time Owned by organization	By stakeholder Approved by supervisor At time Owned by organization
IsAcquiredAt	At location By stakeholder (person) At time Owned by organization	N/A	N/A
IsGeneratedAt	At location By stakeholder (person) Approved by supervisor At time Owned by organization	N/A	N/A

IsModifiedBy	At location By stakeholder (person) At time Approved by supervisor Owned by organization	At time By stakeholder Approved by supervisor Owned by organization	At time By stakeholder Approved by supervisor Owned by organization
Evidence Evaluation (*Supports, Challenges, Weakens, Amplifies, Conflicts, Refutes, Negates, Resolves as well as Document and Evidence attributes*)	By stakeholder Approved by supervisor At time Owned by organization	By stakeholder Approved by supervisor At time Owned by organization	N/A
IsTransferredTo	At location To custodian By stakeholder At time Approved by supervisor Owned by organization	N/A	N/A
IsRevokedAt	By stakeholder Approved by supervisor At time Owned by organization	N/A	N/A

Part 1 Common Elements

The firt part of the specification defines the common elements of the Structured Assurance Case Metamodel. Subsequent parts define the Argumentation Metamodel and the Evidence Metamodel.

8 SACM Assurance Case

This chapter defines the common elements of the Structured Assurance Case Metamodel.

8.1 Administration Class Diagram

This section describes the common elements of SACM that are involved in managing assurance cases, exchanging assurance cases and related concerns. The elements described in this chapter organize instances of SACM. In particular, this section defines the root object of an assurance case - the AssuranceCase element. This element contains other objects in an assurance case, such as the Argumentation objects and EvidenceContainer objects and constitutes a unit of exchange using the SACM as the protocol.

In addition, the SACM Argumentation Metamodel and the SACM Evidence Metamodel constitute two independent protocols within SACM, so Argumentation packages can be developed and exchanged using the Argumentation elements, and also the EvidenceContainers can be developed, managed and exchanged independently of the Argumentation elements or in combination with them. Independently developed Argumentation packages and EvidenceContainer packages can be later assembled into complete assurance cases. Specifications of the Evidence Metamodel can be used to develop an evidence repository that can be used to store and manage evidence in support of multiple assurance cases.

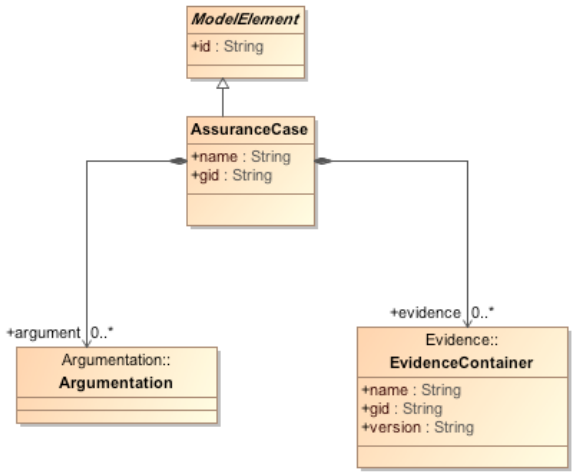

Figure 8.1 Administration Class Diagram

8.1.1 AssuranceCase

AssuranceCase element

Superclass

ModelElement

Attributes

- name:Stringthe name of an assurance case
- gid:Stringthe globally unique identifier assigned to the current assurance case

Associations

- Argumentation::Argumentation[0..*] the argument component of an assurance case
- Evidence::EvidenceContainer[0..*] the evidence component of an assurance case

Semantics

An AssuranceCase element represents assurance cases as defined in ISO/IEC 15206. Argument and Evidence components of an AssuranceCase are optional which allows representing incomplete assurance cases.

An AssuranceCase element involves both a globally unique "gid" and a locally unique "id". The global referencing scheme may involve gid+id combination, while a local scheme may use id component.

AssuranceCase shall have a globally unique gid attribute.

Constraints

gid is a string that has the following structure:

- unique url of the organization that created an assurance case
- the text ‘AssuranceCase’
- a unique number

For each contained object of an assurance case the gid+id identifier is globally unique, i.e., no two elements of the same type produced by the same organization shall have the same number.

8.2 CommonElements Class Diagram

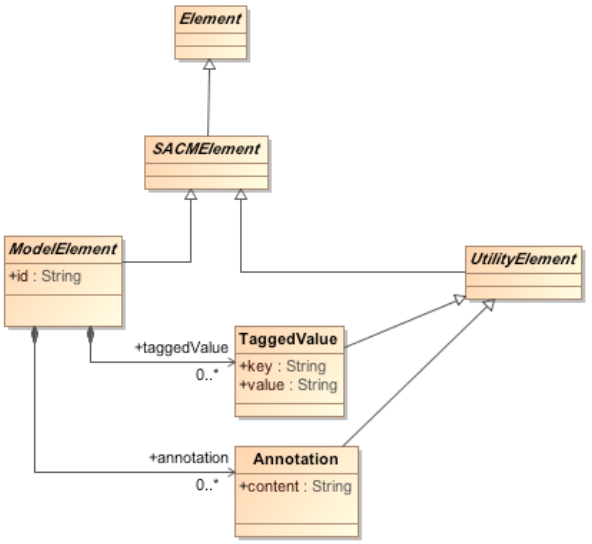

Figure 8.2 CommonElements Class Diagram

8.2.1 SACMElement (abstract)

A SACM element is a top-level element for the Structure Assurance Case Metamodel. This is an abstract class that directly extends MOF::Element. Every class in SACM is a (direct or indirect) subclass of SACMElement.

Superclass

- MOF::Element

Semantics

The SACMElement is a common class for all meta-model elements that represent some element of a structured assurance case.

8.2.2 ModelElement (Abstract)

A ModelElement is an atomic constituent of a structured assurance case represented using the Structured Assurance Case Metamodel. In the meta-model, ModelElement is the top meta-element in the SACM Common class hierarchy. ModelElement is an abstract meta-model element.

Attributes

- id: String A unique identifier for the SACM entity.

Associations

- taggedValue:TaggedValue[0..*] This association enables the association of one or more user defined TaggedValues to any ModelElement.

- annotation:Annotation[0..*] user defined annotations associated with the current element

Semantics

The ModelElement is a common class for all meta-model elements that represent some element of a structured assurance case.

id of the model element shall be unique in the corresponding package (AssuranceCase, Argumentation or EvidenceContainer). Integration of multiple packages into a larger package, for example, adding Argumentation and EvidenceContainer to an AssuranceCase shall not affect the uniqueness of ids of all the objects involved.

Invariants

- context ModelElement inv UniqueIdentifier: ModelElement.allInstances()-> select(me:ModelElement|me.identifier=self.identifier)->size()= 1

8.2.3 UtilityElement (Abstract)

A UtilityElement is an atomic constituent of a structured assurance case represented using the Structured Assurance Case Metamodel. In contrast to a ModelElement, UtiiltyElement represents auxiliary constructs that extend ModelElement and that are only used as part of some ModelElement. In particular, such UtilityElement cannot be referenced outside of the owner ModelElement. UtilityElement is an abstract class.

Semantics

The UtilityElement is a common class for all meta-model elements that represent some auxiliary element of a structured assurance case.

8.2.4 TaggedValue

A TaggedValue is a structured annotation that can be provided on any ModelElement in the Strucutred Assurance Case Metamodel.

Attributes

- key: String
  A key for the TaggedValue.
- value: String
  The value of the TaggedValue.

Semantics

It can be useful to be able to tag values onto the ModelElements. For example, TaggedValues can record versioning information, ownership information, and external URI references. This is a deliberately general mechanism to allow users to associate tags that they find useful for any Structured Assurance Case Metamodel object.

8.2.5 Annotation

An Annotation element represents informal and unstructured user-defined content to any ModelElement of the Structure Assurance Case Metamodel. In contrast, a TaggedValue element allows more structured content to be added to elements.

Superclass

UtilityElement

Attributes

- content:Stringthe text of the annotation

Semantics

It can be useful to be able to add informal text to the ModelElements. For example, Annotation elements can record comments, notes and general explanations. It may also be useful to provide annotations such as review comments and the relevant clauses of assurance standards. This is a deliberately general mechanism to allow users to associate annotations that they find useful for any Structure Assurance Case Metamodel object.

Part 2 Argumentation Metamodel

This part of the specification defines the Argumentation Metamodel.

9 SACM Argumentation Metamodel

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.

9.1 Argumentation Class Diagram

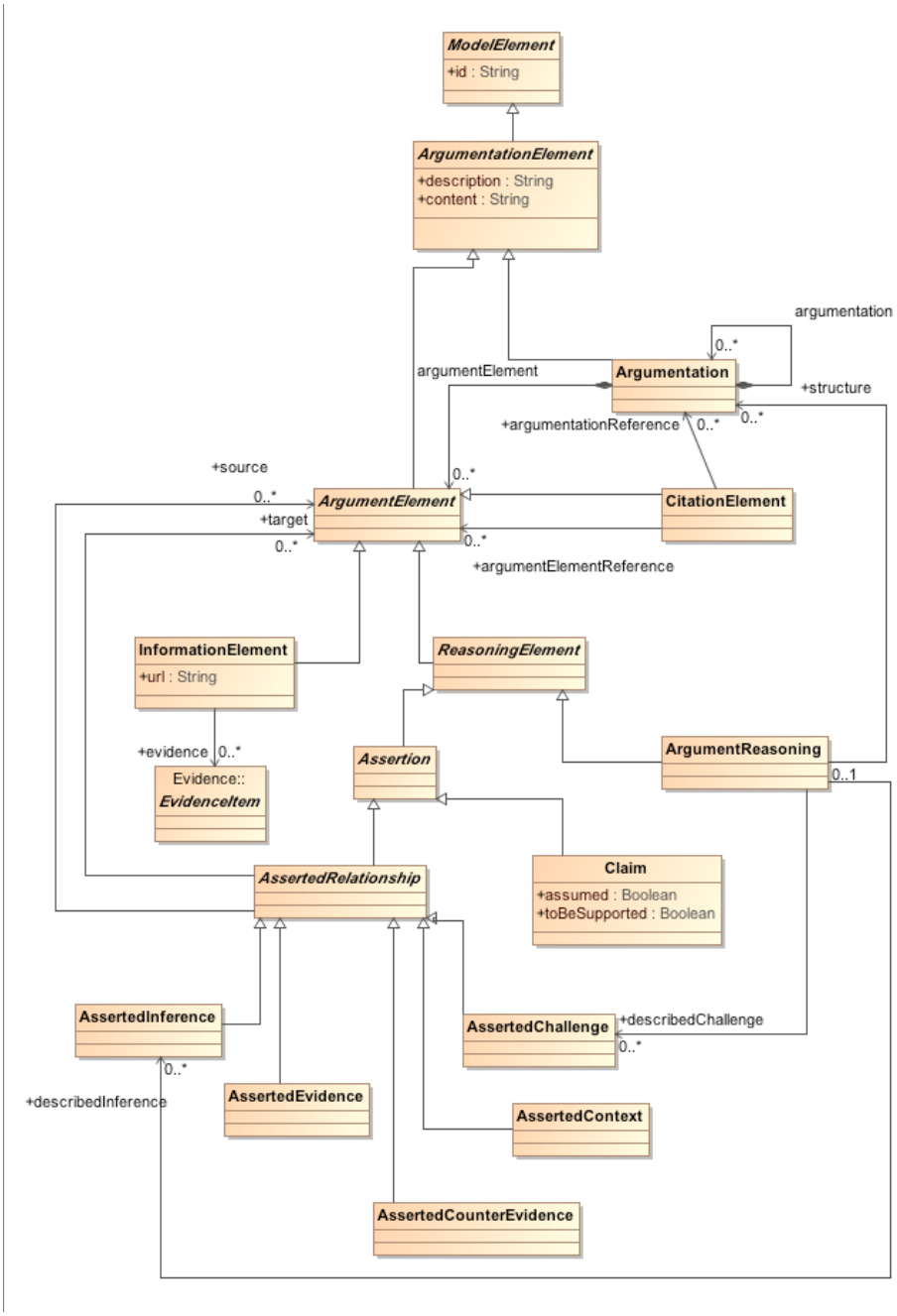

Figure 9.1 - Argumentation Class Diagram

In the following sections we describe the model elements.

9.1.1 ArgumentationElement class (abstract)

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

Attributes

- description: String A description of the Argumentation entity.
- content: String Supporting content of the Argumentation entity.

Semantics

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

9.1.2 Argumentation Class

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

Superclass

ModelElement

Associations

- argumentElement:ArgumentElement[0..*]
  The ArgumentElements contained in a given instance of an Argumentation.
- argumentation:Argumentation[0..*]
  The nested Argumentation contained in a given instance of an Argumentation

Semantics

Structured arguments represented using The Argumentation Metamodel are composed of ArgumentElements. Argumentaion elements can be nested.

For example, arguments can be established through the composition of Claims (propositions) and the AssertedInferences between those Claims.

Example

```
 <ARM:Argument xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:ARM="ARM" xmi:id="0">
</ARM:Argument>
```

9.1.3 ArgumentElement Class (Abstract)

The ArgumentElement Class is the abstract class for the elements of any structured argument represented using the Argumentation Metamodel.

Superclass

ModelElement

Semantics

ArgumentElements represent the constituent building blocks of any structured Argument.

For example, ArgumentElements can represent the Claims made within a structured Argument.

9.1.4 Assertion Class (Abstract)

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

Superclass

ReasoningElement

Semantics

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

9.1.5 ReasoningElement Class (Abstract)

The ReasoningElement Class is the abstract class for the elements that comprise the core reasoning of any structured argument represented using the Argumentation Metamodel – Assertions and ArgumentReasoning (the description of inferential reasoning that exists between Claims.

Superclass

ArgumentElement

Semantics

The core of any argument is the reasoning that exists to connect assertions of that argument. Reasoning is captured in the SACM through the linking of fundamental claims and the description of the relationships between the claims. ReasoningElements represent these two elements.

9.1.6 InformationElement Class

The InformationElement Class enables the citation of a source of that *relates* to the structured argument. The citation is made by the InformationElement class. The declaration of relationship is made by the AssertedRelationship class.

Superclass

ArgumentElement

Attributes

- url: String An attribute recording a URL to external evidence.

Associations

- evidence:Evidence::EvienceItem[0..*]
  The EvidenceItems refernced by the current InformationElement object.

Semantics

It is necessary to be able to cite sources of information that support, provide context for, or provide additional description for the core reasoning of the recorded argument. InformationElements allow there to be an objectified citation of this information within the structured argument, thereby allowing the relationship between this information and the argument to also be explicitly declared.

The url attribute is only to be used when only the argumentation aspects of the SACM are complied with. If compliance is claimed against both the argumentation and evidence packages, then the association to Evidence::EvidenceItem shall be used to reference evidence by means of a URL.

Example

<containsArgumentElement xsi:type="ARM:InformationElement" xmi:id="14" identifier="S2.1" description="" content="black box testing"/>

9.1.7 CitationElement Class

The CitationElement Class cites an Argumentation, or an ArgumentElement within another Argumentation, for use within the current Argumentation.

Superclass

ArgumentElement

Associations

- argumentElementReference:ArgumentElement[0..*]
  References an ArgumentElement within another Argument.
- argumentationReference:Argumentation[0..*]
  References an Argumentation.

Semantics

Within an Argumentation (package) it can be useful to be able to cite elements of an Argumentation (i.e., ArgumentElements) to act as explicit proxies for those elements acting within the argumentation structure. For example, in supporting a Claim it may be useful to cite a Claim or InformationElement declared within another Argumentation. It can also be useful to be able to cite entire Argumentationss. For example, in supporting a Claim it may be useful to cite an existing (structured) Argumentation.

9.1.8 Claim Class

Claims are used to record the propositions of any structured Argumentation. Propositions are instances of statements that could be true or false, but cannot be true and false simultaneously.

Superclass

Assertion

Attributes

- assumed: Boolean
  An attribute recording whether the claim being made is declared as being assumed to be true rather than being supported by further reasoning.
- toBeSupported: Boolean
  An attribute recording whether further reasoning has yet to be provided to support the Claim (e.g. further evidence to be cited).

Semantics

The core of any argument is a series of claims (premises) that are asserted to provide sufficient reasoning to support a (higher-level) claim (a conclusion).

A Claim that is *intentionally* declared without any supporting evidence or argumentation can be declared as being *assumed* to be true. It is an *assumption*. However, it should be noted that a Claim that is not 'assumed' (i.e., assumed = false) is not being declared as false.

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

Invariants

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

Example

<containsArgumentElement xsi:type="ARM:Claim" xmi:id="5" identifier="C1.1" description="" content="Unintended opening of press (after PoNR) can only occur as a result of component failure"/>

9.1.9 EvidenceAssertion Class

A sub-type of Claim used to record propositions (assertions) made regarding an InformationElement being used as supporting evidence to the Argument. This is intended to be used as an interface element to external evidence. An evidence assertion is a minimal assertion (proposition) about an item of evidence, and there is no supporting argumentation being offered within the current structured argument.

Superclass

Claim

Semantics

Well supported arguments are those where evidence can be cited that is said to support the most fundamental claims of the argument. It is good practice that these fundamental claims of the argument state clearly the property that is said to exist in, be derived from, or be exhibited by the cited evidence. Where such claims are made these are said to be basic EvidenceAssertions.

Example

<containsArgumentElement xsi:type="ARM:EvidenceAssertion" xmi:id="12" identifier="C2.1.1" content="Failure 1 of PLC state

machine includes BUTTON_IN remaining true"/>

9.1.10 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

- describedInference:AssertedInference[0..*]
  Reference to the AssertedInference being described by the ArgumentReasoning.
- describedChallenge:AssertedChallenge[0..*]
  Reference to the AssertedChallenge being described by the ArgumentReasoning.
- structure:Argument[0..1]
  Optional reference to another structured Argument to provide the detailed structure of the Argument being described by the ArgumentReasoning.

Semantics

The argument step that relates one or more Claims (premises) to another Claim (conclusion) may not always be obvious. In such cases ArgumentReasoning can be used to provide further description of the reasoning steps involved.

Example

<containsArgumentElement xsi:type="ARM:ArgumentReasoning" xmi:id="2" identifier="RC1.1" content="Argument by omission of all identified software hazards" describes="5 6"/>

9.1.11 AssertedRelationship Class (Abstract)

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

Superclass

Assertion

Associations

- source:ArgumentElement[0..*]
  Reference to the ArgumentElement(s) that are the source (start-point) of the relationship.
- target:ArgumentElement[0..*]
  Reference to the ArgumentElement(s) that are the target (end-point) of the relationship.

Semantics

In the SACM, the structure of an argument is declared through the linking together of primitive ArgumentElements. 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.

Example

9.1.12 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, a 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.

Example

```
<containsAssertedRelationship xsi:type="ARM:AssertedInference" xmi:id="16" identifier="C1.1.1" description="" target="5"
source="1"/>
```

Invariants

```
context AssertedInference
inv SourceMustBeClaim : self.source->forAll(s|s.oclIsTypeOf(Claim))
inv TargetMustBeClaimOrAssertedRelationship : self.target->forAll(t|t.oclIsTypeOf(Claim) or
 t.oclIsTypeOf(AssertedRelationship))
```

9.1.13 AssertedEvidence Class

The AssertedEvidence association class records the declaration that one or more items of Evidence (cited by InformationItems) provides 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 information (cited by an InformationItem) may provide evidence for more than one Claim.

Superclass

AssertedRelationship

Semantics

Where evidence (cited by InformationItems) 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 some information cited by an InformationElement 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.

Example

<containsAssertedRelationship xsi:type="ARM:AssertedEvidence" xmi:id="22" identifier="S1.1" target="10" source="5 6"/>

Invariants

context AssertedEvidence
inv SourceMustBeInformationElement : self.source->forAll(s|s.oclIsTypeOf(InformationElement))
inv TargetMustBeClaimOrAssertedRelationship : self.target->forAll(t|t.oclIsTypeOf(Claim) or
t.oclIsTypeOf(AssertedRelationship))

9.1.14 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

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).

Invariants

context AssertedChallenge
inv SourceMustBeClaim : self.source->forAll(s|s.oclIsTypeOf(Claim))
inv TargetMustBeClaimOrAssertedRelationship : self.target->forAll(t|t.oclIsTypeOf(Claim) or
t.oclIsTypeOf(AssertedRelationship))

9.1.15 AssertedCounterEvidence Class

AssertedCounterEvidence can be used to associate evidence (cited by InformationElements) 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).

Invariants

context AssertedCounterEvidence
inv SourceMustBeInformationElement : self.source->forAll(s|s.oclIsTypeOf(InformationElement))
inv TargetMustBeClaimOrAssertedRelationship : self.target->forAll(t|t.oclIsTypeOf(Claim) or t.oclIsTypeOf(AssertedRelationship))

9.1.16 AssertedContext Class

The AssertedContext association class declares that the information cited by an InformationElement provides a context for the interpretation and definition of a Claim or ArgumentReasoning element.

Superclass

AssertedRelationship

Semantics

Claim and ArgumentReasoning often need contextual information to be cited in order for the scope and definition of the reasoning to be easily interpreted. 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 InformationItem B" or conversely "InformationItem B is the valid context for Claim A"). A declaration (AssertedContext) of context (InformationItem) for a ReasoningElement (A – the contextual InformationItem – and B – the ReasoningElement) denotes that A is asserted to be valid contextual information for B (i.e., A defines context where the reasoning presented by B holds true).

Example

<containsAssertedRelationship xsi:type="ARM:AssertedContext" xmi:id="21" identifier="CIRC1.1" target="4" source="2"/>

Invariants

context AssertedContext
inv SourceMustBeInformationElement :self.source->forAll(s|s.oclIsTypeOf(InformationElement))
inv TargetMustBeReasoningElement : self.target->forAll(t|t.oclIsTypeOf(ReasoningElement))

Part 3 Evidence Metamodel

This part of the Strucutred Assurance Case Metamodel defines the normative SACM Evidence Metamodel.

SACM Evidence Metamodel consists of 18 class diagrams. SACM Evidence Metamodel is delivered as a single UML subpackage ‘Evidence’ of SACM.

The SACM Evidence Metamodel consists of the following logical parts:

- Evidence Items
- Formal Elements
- Evidence Assertions
- Administration

The **Evidence Items** part defines the physical evidence, provided in the form of documents, records and sometimes other material exhibits. The **Formal Elements** part defines the logical assertions, provided in the form of individual propositions. These propositions use an external vocabulary related to the subject area for which an argument is being provided. The Formal Elements part defines a subset of an OMG Semantics of Business Vocabularies and Business Rules (SBVR) fact model in the form of atomic formulations based on fact types with roles bound to individual concepts. SBVR is not used directly because of the semantic differences between fact models in linguistic models as they are defined in SBVR, conceptual models and “asserted fact models” involved in evidence collection and evaluation. Formal Elements represent a conceptual model underlying the entire assurance case. **Evidence Assertions** part defines various *statements* that can be made about the evidence items, such as documents, records and exhibits, and their relations to the subject area claims. Evidence Assertions includes statements that are related to various esential properties of evidence items. A large group of statements are the so-called *evidence evaluations*, including assertions of the evidentiary support (relations between evidence items and the subject area claims), assertions related to the interpretation of physical evidence and document, assertions about the conflicts in evidenctiay support and resolutions of these conflicts. Other statements are assertions related to provenance, custody and timing of the evidence items and evidence evaluations. The last group of statements qualify the evidentiary support that evidence items confer on the subject area claims. The **Administration** part defines an EvidenceContainer element which organizes individual evidence items and evaluations into a package that becomes a unit of exchange. The Administrative part also provides several means for managing evidence-related efforts.

10 Evidence Elements

10.1 Evidence Elements Class Diagram

This section defines the key concepts of the SACM Evidence Metamodel. The elements in this section are defined as abstract classes and subsequent sections elaborate the detail, while this section provides a convenient outline of the entire vocabulary focusing at the key noun concepts.

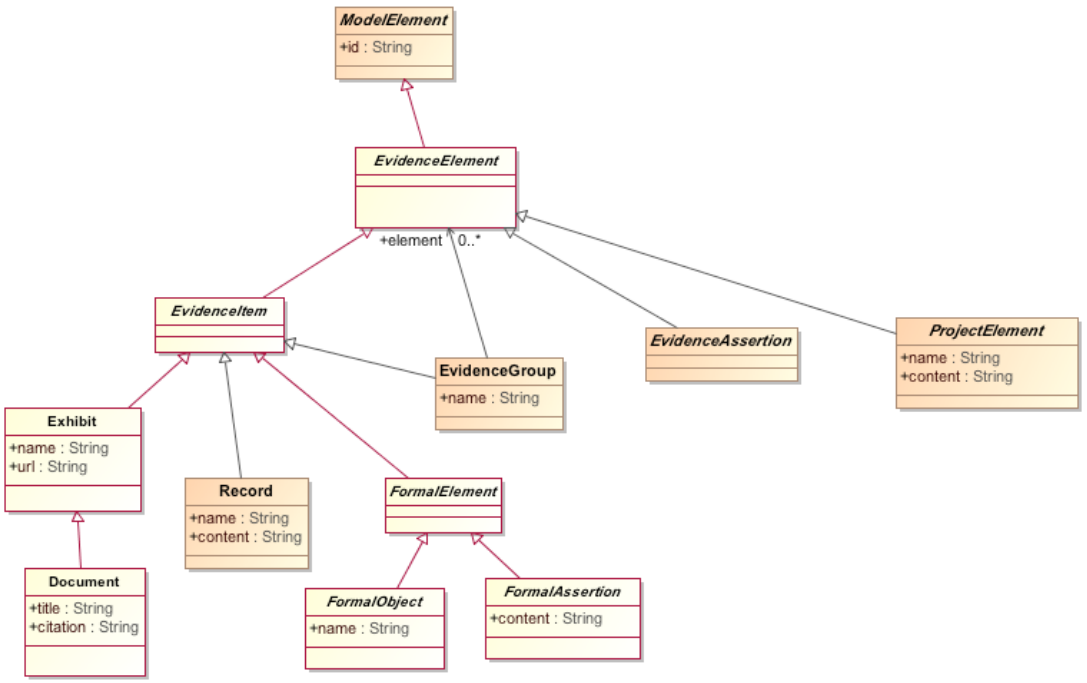

Figure 10.1 - EvidenceElements class diagram

10.1.1 EvidenceElement (abstract)

EvidenceElement class is the root element of the SACM Evidence Metamodel. All other classes in the SACM Evidence Metamodel extend EvidenceElement. The main subclass of the EvidenceElement is EvidenceItem, which defines the primary elements of the Evidence Metamodel. Other elements represent various secondary elements and dependent parts of other evidence elements. The following elements are direct subclasses of EvidenceElement: EvidenceItem, EvidenceAssertion, and ProjectElement.

Superclass

ModelElement

Associations

- provenance:Provenance[0..*]
  Provenance properties of the EvidenceElement

- timing:TimingProperty[0..*]
  Timing properties of the EvidenceElement
- custody:CustodyProperty[0..*]
  Custody properties of the EvidenceElement
- event:EvidenceEvent[0..*]
  Event properties describing a set of events with timing clauses determined by the lifecycle of the EvidenceElement

Note: This is the complete list of associations for EvidenceElement as they are introduced by several other diagrams of the Evidence Metamodel.

Semantics

EvidenceElement class is an abstract class that represents any element of the SACM Evidence Metamodel. Every class of the SACM Evidence Metamodel extends EvidenceElement directly or indirectly (through other classes).

EvidenceElement may own certain EvidenceProperties. When an EvidenceElement owns an EvidenceProperty, the property represents a relationship between the current EvidenceElement object and some other object referenced by the corresponding EvidenceProperty. Similarly, EvidenceElement may own certain EvidenceAttribute. When an EvidenceElement owns an EvidenceAttribute, the attribute represents a relationship between the current EvidenceElement object and some other object that is referenced by the corresponding EvidenceAttribute.

10.1.2 EvidenceItem (abstract)

EvidenceItem is an abstract class that represents objects that are collected as evidence or are somehow involved with evidence being collected. These objects are either physical documents, records, formal objects (representing concrete objects or concepts), or formal assertions (see below). EvidenceItem owns a set of events that represent the lifecycle and the chain of custody of the item.

The very nature of evidence is that some physical objects called “exhibits” are produced to provide justification to the claims made in an argument. This form of justification conferred by a physical object to a claim is called evidentiary support. So, the main evidence item is an Exhibit - a physical object produced believed to be conferring evidentiary support to some claims in the argument.

The most common form of an exhibit is a Document. Document is a special object, because it is a direct expression of some meaning in certain media. In Software Assurance, most documents are electronic, however some documents may exist on paper or any other media. In comparison any other physical object may represent a meaning only in a very indirect way. Physical objects other than documents require non-trivial (and highly contestable) interpretation, as to what meaning they may represent. Classes Exhibit and Document are described below. Statements related to their properties, are represented by the subclasses of the abstract class ExhibitProperties and DocumentProperties are described in chapter 11 “Exhibit Properties”.

Superclass

EvidenceElement

Semantics

EvidenceItem represents objects that are collected as evidence. The subclasses of EvidenceItem are Exhibit, representing physical objects presented as evidence, Record, EvidenceGroup and FormalElement, which represents associated elements of meaning, such as concepts and propositions/claims.

10.1.3 Exhibit

Exhibit element represents a physical object presented as evidence because it is believed to confer evidential support to some claims. Exhibit element in the Evidence Metamodel is a representative of this physical object within the Evidence Model, so that additional properties can be attached to it, and so that it can participate in various relationships with other elements of the Evidence Model. The nature of Exhibit as something that is presented as evidence and subsequently stored in an appropriate evidence repository, provides the scope of what can be presented as evidence. For example, a “knife” can be presented as evidence, but a person cannot be. A person can have viewed as a witness or an expert, and his opinion recorded as a document, which then can be presented as evidence. The SACM Evidence Metamodel emphasizes computer-based evidence repositories, which can only store electronic representations of physical objects. So the "electronic source" of a "knife" object will likely be a photograph of the knife.

A most common kind of an exhibit is a Document. Document is a special object, because it is a direct expression of some meaning in certain media. Document involves the use of a language to express its meaning. In comparison any other physical object may represent a meaning only in a very indirect way. Physical objects require non-trivial (and highly contestable) interpretation, as to what meaning they may represent. The important of documents as elements of evidence can not be underestimated, since evidentiary support is a form of establishing defensible relation between some physical objects and claims, which are elements of meaning. This transition from physical objects to meanings needs to be performed as early as possible in the process of building an assurance case. The Evidence Metamodel provides the means to document this transition and confine it to the scope of the evidence package, so that the rest of an assurance case can operate only with claims as elements of meaning, rather than with any physical objects, including documents.

The Evidence Metamodel defines some common properties of exhibits including the name (short title) of the exhibit, electronic source of the exhibit, the media (the material of the object).

Superclass

EvidenceItem

Attributes

- name:String
  The short title of the exhibit.
- url:String
  The URL to the original exhibit, if it is a web resource.

Associations

- property:ExhibitProperty[0..*]
  The set of essential properties of the exhibit.

Semantics

Exhibit element represents a physical object that is presented as evidence in support of some claims. Properties of an Exhibit are defined as attributes of the Exhibit class itself, as well as the owned elements of the ExhibitProperty class. Each subclass of the ExhibitProperty class owned by an Exhibit object defines a characteristic of the exhibit, represented by the Exhibit object.

10.1.4 Document

Document element represents a "document" which is defined as follows:

1. an original or official paper relied on as the basis, proof, or support of something;

2. something (as a photograph or a recording) that serves as evidence or proof;

3a: a writing conveying information b: a material substance (as a coin or stone) having on it a representation of thoughts by means of some conventional mark or symbol [Merriam-Webster Dictionary].

Document element is the main subclass of Exhibit. Document is a special object, because it is a direct expression of some meaning in certain media. In Software Assurance, most documents are electronic, however some documents may exist on paper or any other media. Document involves the use of a language to express its meaning. In comparison any other physical object may represent a meaning only in a very indirect way. Physical objects require non-trivial (and highly contestable) interpretation, as to what meaning they may represent. FormalAssertion and FormalObject on the other hand are representations of some meaning rather than of an expression of a meaning (direct or indirect). FormalObject may refer to some physical objects as its extent but it may not correspond to any physical object whatsoever. From this perspective, a Document is a vital kind of a physical object, which is directly related to some meaning, and requires only a limited interpretation. The importance of documents as elements of evidence cannot be underestimated, since evidentiary support is a form of establishing defensible relation between some physical objects and claims, which are elements of meaning. This transition from physical objects to meanings needs to be performed as early as possible in the process of building an assurance case. The Evidence Metamodel provides the means to document this transition and confine it to the scope of the evidence package, so that the rest of an assurance case can operate only with claims.

The SACM Evidence Metamodel defines some common properties of documents, such as Title, version, language, etc. Several properties are defined as attributes of the class Document, others are defined as owned properties through named association classes, which are concrete subclasses of DocumentProperty. In addition, the Evidence Metamodel allows several attributes of a Document that characterize its quality as evidence.

Superclass

Exhibit

Attributes

- title:String
  The full title of the document
- citation:String
  The full citation of the document (bibliographical reference)

Semantics

Document element represents a physical object that is directly expresses a certain meaning. The meaning is the content of the document. Because of the ambiguity of natural languages, some documents may expresses more that one meaning. Formal documents usually have a single meaning. Properties of a Document are defines attributes of the Document class itself, as well as the owned elements of the DocumentProperty class. Each subclass of the DocumentProperty class owned by a Document object defines a characteristic of the document, represented by the Document object.

10.1.5 Record

Record element represents Exhibits that are explicit records of compliance, for example log entries. Record is different from a Document, since a Document element represents some physical object that exists elsewhere in the physical world (even if it is an electronic document), while a Record element exists only in the EvidenceContainer.

Superclass

EvidenceElement

Attributes

- name:Stringthe name of the record
- content:String the content of the record

Semantics

Record is defined as "a thing constituting a piece of evidence about the past, esp. an account of an act or occurrence kept in writing or some other permanent form". In the Evidence Metamodel Record element is such thing. In contrast to a Document element, a Record is not a representative of some other physical object, but the object itself. A Record is therefore similar to an Object, however it is considered a structured element with an informal content rather than a formal element.

10.1.6 FormalElement (abstract)

FormalElement is an abstract class that represents any elements of meaning that are associated with objects presented as evidence or otherwise involved in the evidence collection.

Superclass

EvidenceItem

Semantics

FormalElement is an element of meaning that represents a certain individual concept, a noun concept, verb phrases and propositions. Two subclasses of FormalElement are FormalObject, representing noun concepts, and FormalAssertion, representing verb concepts and propositions.

10.1.7 FormalObject (abstract)

FormalObject is an abstract class that represents any elements of meaning that are noun concepts associated with the objects that are collected as evidence or are otherwise involved in the evidence collection. FormalObject may represent a concept corresponding to an individual concrete physical thing, such as “an axe with stains of blood on it,” or a collection of things, referred to as a whole, or a concept, such a “murder weapon.” Physical things need to be represented as the exhibits. On the other hand, concepts are usually not collected as evidence, rather they are used as the elements of meaning in order to build assertions, as well as other relations describing the items of evidence. For example, in order to describe the abovementioned “axe” as a “murder weapon,” the instance of a FormalObject with the name “murder weapon” is used. This object represents a concept that is involved in making a claim that also involves a concrete physical object. FormalObjects represent concepts in the subject area for which the argument is being developed. Many elements of the Evidence Metamodel are concepts related to evidence. In particular, Exhibit and Document is two key concepts related to evidence.

Superclass

FormalElement

Attributes

- name:String
  Name of the domain concept

Semantics

FormalObject is an element of meaning that represents a certain individual concept (other than a document) or a noun concept.

10.1.8 FormalAssertion (abstract)

FormalAssertion is an abstract class that represents propositions that are involved in evidence collection. In particular, FormalAssertion involves FormalObject that represent a individual concepts corresponding to concrete physical things, collection of things, referred to as a whole, or concepts. FormalAssertions represent propositions about the subject area for which an assurance case is being developed. In contrast, many elements of the Evidence Metamodel are assertions about evidence. In particular, EvidenceEvaluation is one of the key assertions related to evidence.

Superclass

FormalElement

Attributes

- content:String
  The statement that in a selected language that is the expression of the formal assertion (verbalization of the assertion in a natural language).

Semantics

FormalAssertion is an element of meaning that represents a certain proposition. The Assertion subclass, introduced in Chapter 12 “Formal Statements” uses elements of formal statements and a formal reference to an SBVR vocabulary to represent precise meaning of the assertion. ReferencedClaim element represents an informal assertion/claim.

10.1.9 EvidenceGroup

EvidenceGroup asserts a state of affairs that several evidence elements are grouped together and can be referred to collectively.

Superclass

EvidenceItem

Attributes

- name:String
  Name of the evidence group.

Associations

- element:EvidenceElement[0..1]
  Elements of the Evidence Group

Constraints

- EvidenceGroup can not be an element of itself, either directly or indirectly through membership in other Evidence Group.

Semantics

EvidenceGroup asserts a state of affairs that several evidence elements are grouped together and can be referred to collectively. EvidenceGroup is a special subclass of EvidenceItem acting as a named container for evidence items that can be used on both sides of an evidence relation. An EvidenceElement may be a member of more than one EvidenceGroup.

10.2 EvidenceAssertions Class Diagram

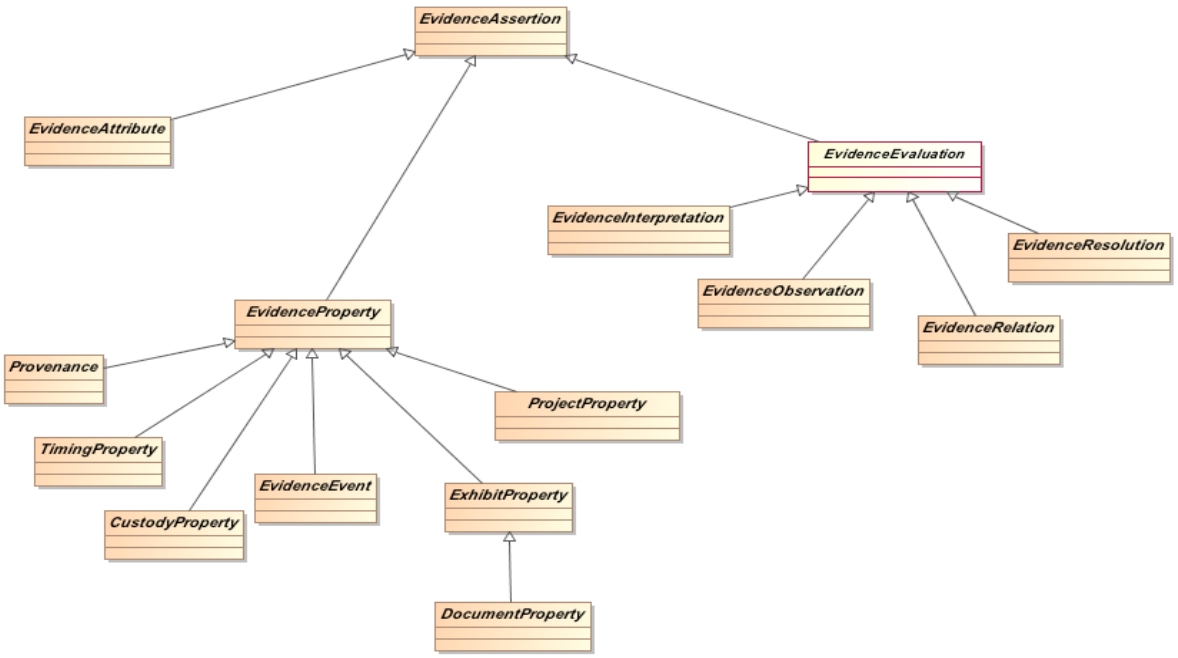

Figure 10.2 EvidenceAssertions class diagram

10.2.1 EvidenceAssertion (abstract)

EvidenceAssertion represents various statements about the evidence items, such as documents and exhibits, and their relations to the subject area claims.

Evidence Assertions are defined within the Evidence Metamodel and include the following categories:

- Statements related to various essential properties of Evidence Items
- Properties of Documents as they are related to the quality of the evidentiary support that may be offered by these documents, such as Primary or secondary, original or derived, Consistency, Completeness, Accuracy.
- Statements related to the Custody, Provenance and Timing of Evidence Elements
- Attributes of the evidentiary support, such as Direct or indirect support, Relevance, Confidence, Strength, Significance.
- Interpretation of Evidence: what an evidence item "Is", what it "means."
- Nature of the evidentiary support: Supports, Challenges.
- Observations and Resolutions.
- Standard of Proof to which the evidence is evaluated.

Superclass

EvidenceElement

Semantics

EvidenceAssertion is an abstract class that represents various assertions related to evidence elements defined in the Evidence Metamodel. More detailed semantics is provided by the concrete subclasses of EvidenceAssertions.

10.2.2 EvidenceProperty (abstract)

EvidenceProperty represents various statements related to the fundamental properties of evidence elements.

Superclass

EvidenceAssertion

Semantics

EvidenceProperty is owned by the subject EvidenceElement. EvidenceProperty is a statement that represents fundamental properties of the EvidenceElement. Such properties are independent of the particular assurance case, for example, the media of a document, the current custodian of the document, or the author of a statement. EvidenceProperty involves one or more objects, specified either as attributes or the associations of the EvidenceProperty element. Each EvidenceProperty represents a relationship between the subject Element that owns it and the corresponding objects.

10.2.3 EvidenceEvaluation (abstract)

Establishing evidentiary support that a set of documents provides to the given claim requires evaluation of the documents and its relations to the claims, including the detection of challenges to the claim, conflicts, and contradictions. Satisfying a certain standard of proof requires analysis of all available evidence items and resolving/explaining conflicts, so that at the end all evidence points in a single direction. Often this requires formulation of a multitude of intermediate claims that are clearly supported by available evidence items and establishing further relations to the target claim.
EvidenceEvaluation is an abstract element that represents relationships between evidence items and assertions, observations regarding conflicts, and resolutions of the conflicts. Navigation through the EvidenceEvaluation elements for the given domain claim allow understanding the exact nature and strength of the evidentiary support provided by the evidence items to the claim. EvidenceEvaluation elements are subjects for additional EvidenceProperty clauses.

Superclass

EvidenceAssertion

Associations

- attribute:EvidenceAttribute[0..*]
  Set of quality attributes of this EvidenceEvaluation element.

Semantics

EvidenceEvaluation establishes relationship between endpoints, such as between EvidenceItems, as well as between EvidenceEvaluation elements themselves. EvidenceAttribute elements owned by the EvidenceEvaluation determine the properties of the relation between the endpoints of the EvidenceEvaluation.

11 Exhibit Properties

11.1 ExhibitProperties Class Diagram

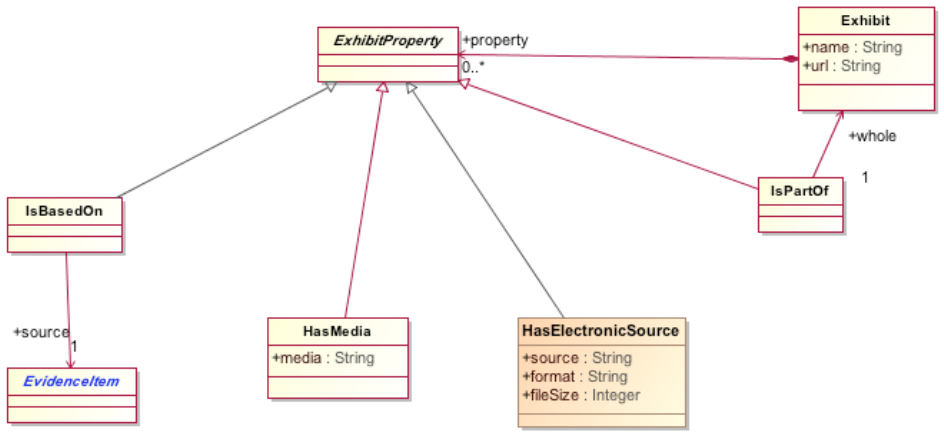

Figure 11.1 - ExhibitProperties class diagram

11.1.1 Exhibit Property

This class defines common physical characteristics of exhibits, including documents.

Superclass

EvidenceProperty

Semantics

Each concrete subclass of ExhibitProperty defines a single characteristic of the exhibit. An instance of a concrete subclass of the ExhibitProperty class that is owned by some Exhibit object defines a characteristic of the exhibit represented by the Exhibit object.

11.1.2 HasElectronicSource

HasElectronicSource represents the expression of an Exhibit in electronic form. Electronic Source is the only way a document may be stored in a computer based Evidence Repository. For example, Electronic Source can be a photograph of an object, a scanned image of a document, a Word document, an XMI representation of a model. In a general case of a non-document exhibit, the electronic source is likely to be some image of the original object. If the physical object existed in electronic form (as specified by the Media property), then the Electronic Source can be considered the “original” representation of the Exhibit. This is often the case with documents. In case of documents as exhibits, the concern is to capture the expression of the meaning represented by the document. If the physical document existed in electronic form as some kind of text (as specified by the Media property), then the Electronic Source can be considered the “original” expression of the Exhibit. In other cases, the Electronic Source is a “derived” expression, which can be a source of errors leading to incorrect interpretation of the meaning of the document. Some arguments involve physical evidence where the transformation between a physical object and its electronic form may be contested, especially if the electronic form is

used to interpret the meaning of the document. For example, if the original document is a handwritten note on a napkin, the original electronic source may be a photographic image of the note. However before the meaning of the note can be analyzed, the text version of the note has to be presented. This may involve some degree of interpretation (was this letter "g" or letter "q"?). In this case the text version of the note is a different electronic source. In most cases related to Software Assurance, electronic source in the form of text is either the original media, or the transformation is reliable.

Superclass

ExhibitProperty

Attributes

- source:String
  The bytestream representing the owner exhibit in electronic form.
- format:String
  The format used by the source.
- fileSize:Integer
  The size of the bytestream (in bytes).

Constraints

- Exhibit shall not have more than one HasElectronicSource property.

Semantics

HasElectronicSource element represents three related properties of the owner Exhibit object, corresponding to the electronic representation of the exhibit. The source property establishes a relationship between the owner Exhibit object and bytestream, which is interpreted as the electronic form of the Exhibit. The source uses the format, and the source has size. We do not make a distinction between single byte character and multi-byte character representations in case of text-based documents. These distinctions shall be made by the format property. The source within the HasElectronicSource property shall represent the entire exhibit, therefore it is not allowed for the exhibit to have more than one electronic source. If an argument requires reference to alternative electronic sources, for example, images at different resolution, the evidence model needs to be more explicit, and include the original exhibit and two derived documents, describing the process of derivation. This allows clear representation of detailed interpretation of each document, unambiguous representation of claims supported by both documents, and evaluation of their contribution to the main claim.

The main characteristic is expressed by a sentential form "Exhibit has electronic source."

11.1.3 IsPartOf

Some exhibits may have complex structure in which different parts render evidentiary support to different claims, and/or have different properties. The SACM Evidence Metamodel allow representing each part of the complex exhibit as a separate Exhibit element, to represent the aggregated whole by another Exhibit element and to represent "part-whole" associations using the "IsPartOf" property.

Superclass

ExhibitProperty

Associations

- whole:Exhibit[1]
  The Exhibit object that represents the "aggregated whole" to which the current Exhibit object is a part of.

Semantics

IsPartOf is a characteristic of Exhibit-1 (instance of a Exhibit class, referred to as the owner of the characteristic), which is defined as a state of affairs that the Exhibit-1 is part from another Exhibit-2.

This characteristic is expressed by a sentential form “Exhibit-1 is part of Exhibit-2.” Exhibit-1 may be part of multiple other exhibits, besides Exhibit-2, and Exhibit-2 may have other exhibits as its parts.

11.1.4 HasMedia

It is often important to identify a particular media of the document or the material of the exhibit. ExhibitProperty HasMedia shall be used for this purpose.

Superclass

ExhibitProperty

Attributes

- media:String
  Designator of the media of the original Exhibit

Semantics

HasMedia element represents a characteristic of the owner Document object that identifies the media of the original exhibit. The version property establishes a relationship between the owner Document object and the designation of the media of the original exhibit.

The main characteristic is expressed by a sentential form “Exhibit is made of media” or “Document is expressed on media.”

11.1.5 IsBasedOn

In Software Assurance documents are often generated by automated process from some sources. For example, the probabilities of Faults are generated from a Fault Tree model through the process of Fault Tree analysis. IsBasedOn element allows to represent the relationship between the owner document and its sources. From the evidentiary quality perspective the fact that the owner document was generated from other documents by means of some automated process does not necessarily make it a “secondary” source, as the transformation usually adds value and generates some primary information, not available in the sources (at least not explicitly). However, this usually makes the document “derived,” rather that “original,” since the transformation is a potential source of errors. A document may be based on multiple sources, each of which shall be represented as a separate IsBasedOn property of the owned document.

Superclass

ExhibitProperty

Associations

- source:EvidenceItem[1]
  The source document that contributes to the content of the owner document.

Semantics

IsBasedOn is a characteristic of Document-1 (instance of a Document class, referred to as the owner of the characteristic), which is defined as a state of affairs that the content of the Document-1 is derived from another Document-2.

This characteristic is expressed by a sentential form “Document-1 is based on Document-2.” Document-1 may be based on multiple other documents, besides Document-2.

Derivation of one Document from another can have various meanings including, but not limited to the following:

- Version derives from prior version
- Version derives from these versions of items
- Copy
- Uses information from
- Conclusion based on
- Change together or should change if other changes
- Uses
- Subsumes
- Compiled from or otherwise results from tool processing of
- Analysis result regarding
- Obtains resources from
- Share contents

This list is by no means exhaustive and not all may apply to a set of exhibits of interest. Apparently, as natures of dependencies could vary multiple relations related to a single dependent element are possible. The SACM Evidence Metamodel does not provide a normative enumeration of the nature of dependency. However, should an author of a SACM document desire so, a TaggedValue mechanism shall be used for this purpose with a tag 'natureofdependency'.

11.2 DocumentProperties Class Diagram

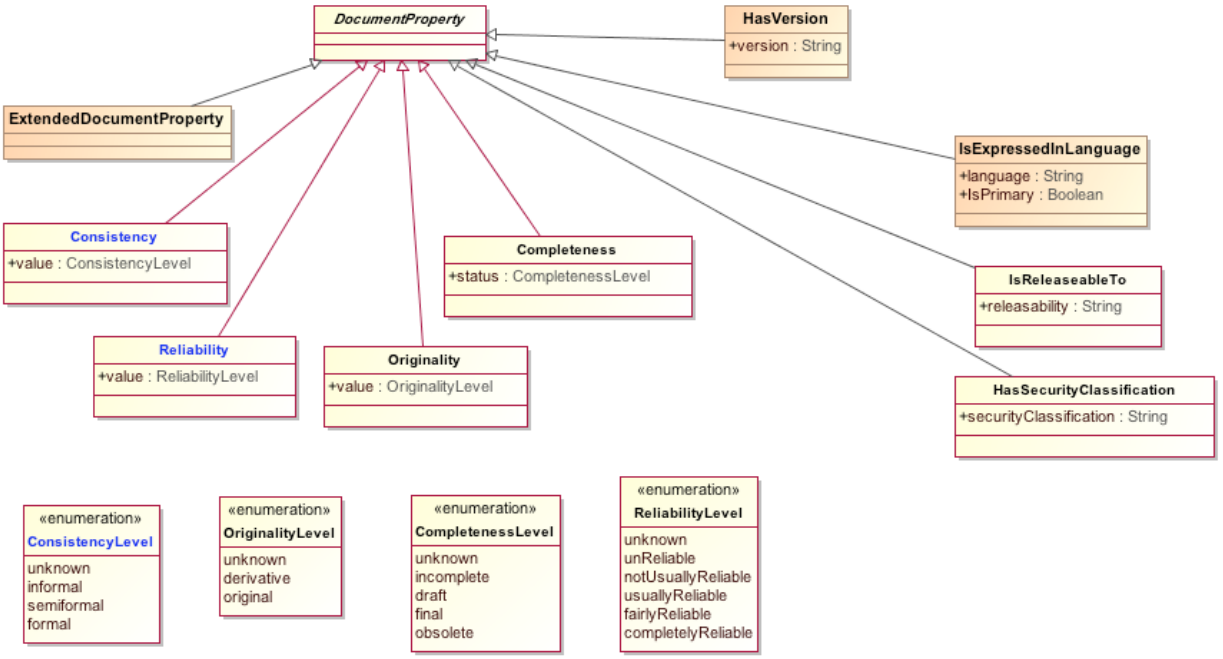

Figure 11.2 Document Properties class diagram

11.2.1 Document Property

This class defines characteristics of documents. Other characteristics common to all Exhibits are defined using ExhibitProperty.

Superclass

ExhibitProperty

Semantics

Each concrete subclass of DocumentProperty defines a single characteristic of the document. An instance of a concrete subclass of the DocumentProperty class that is owned by some Document object defines a characteristic of the document represented by the Document object.

11.2.2 HasVersion

It is often important to identify a particular version of the document. DocumentProperty HasVersion shall be used for this purpose.

Superclass

DocumentProperty

Attributes

- version:String
  Designator of the version of the original Document.

Semantics

HasVersion element represents a property of the owner Document object that identifies the version of the original document. The version property establishes a relationship between the owner Document object and the designation of the version of the original document. The ElectronicSource is a snapshot of the original document captured in electronic form. The version is used to provide full traceability to the original document.

The main characteristic is expressed by a sentential form “Document has version.”

11.2.3 IsExpressedInLanguage

The use of language is one of the essential characteristics of a document. The meaning of the document is expressed as a text that uses a certain vocabulary that is expressed in some language. In the context of the Evidence Metamodel, IsExpressedInLanguage is a document property that established relationship between a document and the language which is essential to understanding the meaning of the document. The language itself is identified as a string attribute of the Language property.

Superclass

DocumentProperty

Attributes

- language:String
  Designation of the language which is used in the owner Document.
- IsPrimary:Boolean
  In case when the document is expressed in multiple languages, this attribute identifies the primary language.

Constraints

- Document should have at least one IsExpressedInLanguage property.
- In case when the Document is expressed in more that one language, the IsPrimary property may be used to identify the primary language.

Semantics

IsExpressedInLanguage element represents a property of the owner Document object that identifies the language of the document. The source property establishes a relationship between the owner Document object and the designation of the language, which is interpreted as the name of a language. A language can be a natural language or an unnatural one, such a computer language, a system of mathematical symbols or a modeling notation. ISO-639-2 provides manes of many languages and provides short language-independent codes. In the scope of the Evidence Metamodel, the language of the each document shall be identified, as this is vital to interpretation of evidence and for exchanging evidence. It is possible that a Document is expressed in more than one language. The SACM Evidence Metamodel allows identifying the primary language by setting the isPrimary attribute to true.

The main characteristic is expressed by a sentential form “Document is expressed in language.” Additional sentential form is “Document is primarily expressed in language.”

11.2.4 HasSecurityClassification

In some contexts of evidence evaluation it is required to track of the security classification of documents. Evidence management tools can use security classification in filters in order to protect sensitive information. HasSecurityClassification property represents security classification of the owner Document.

Superclass

DocumentProperty

Attributes

- securityClassification:String
  Designation of the security classification of the owner document.

Semantics

HasSecurityClassification element represents a property of the owner Document object that identifies the security classification of the original document. The SecurityClassification property establishes a relationship between the owner Document object and the designation of the security property of the original document. SecurityClassification property of the owner Document refers also to all ElectronicSource of the Document. Examples of designations of security classifications are: “Unclassified,” “Secret,” “Top Secret.” When the HasSecurityClassification property is omitted, the Document is assumed to be “Unclassified.”

The main characteristic is expressed by a sentential form “Document has security classification.”

11.2.5 IsReleasableTo

In some contexts of evidence evaluation it is required to track of the releasability of documents. Evidence management tools can use releasability property in filters in order to protect sensitive information. IsReleasableTo property represents security classification of the owner Document.

Superclass

DocumentProperty

Attributes

- releasability:String
  Designation of the releasability of a document.

Semantics

IsReleasableTo element represents a property of the owner Document object that identifies the releasability of the original document. The IsReleasableTo property establishes a relationship between the owner Document object and the designation of the releasability scope of the original document. IsReleasableTo property of the owner Document refers also to all ElectronicSource of the Document. Examples of designations of releasability scope are: “US eyes only,” “Canadian eyes only,” “NATO only.” When the IsReleasableTo property is omitted, the Document is assumed not to have releasability restrictions.

The main characteristic is expressed by a sentential form “Document is releasable to releasability scope.”

Example

11.2.6 Originality

Originality element represents characteristic of documents that is asserted during the course of evaluation and that refers to the originality of the document. This characteristic refers to the document (record) that is the source of evidence. The original source is one that contributes written, oral, or visual information not derived from a prior written or visual record or oral communication. A derivative source is one that contributes information that was copied, transcribed, abstracted, summarized, duplicated, or repeated from information is a previously existing source (that is from the original or another derivative).

Superclass

DocumentAttribute

Attributes

- value:OriginalityLevel
  Originality level, such as derivative or original.

11.2.7 OriginalityLevel (enumeration)

OriginalityLevel enumeration class defines the Originality levels.

Literals

- unknown
  Originality level is unknown.
- derivative
  Document is derivative.
- original
  Document is original.

11.2.8 Consistency

Consistency element represents characteristic of documents that is asserted during the course of evaluation and that refers to the consistency of the document. This characteristic refers to the level of formality of the document and to our capability to interpret the document. Consistency of a document can be informal, semiformal and formal. An informal document uses prose. A semi-formal document uses a template that determines some of its structure, filled in by prose. A form with large amount of prose is an example of a semi-formal document. When the amount of prose becomes limited, the document may be referred to as formal. A multiple-choice questionnaire is an example of a formal document.

Superclass

DocumentAttribute

Attributes

- value:ConsistencyLevel
  Consistency level of the Document, such as informal, semi-formal and formal.

11.2.9 ConsistencyLevel (enumeration)

The ConsistencyLevel enumeration class defines consistency levels.

Literals

- unknown
  Consistency level is unknown
- informal
  Consistency level is informal
- semiformal
  Consistency level is semi-format
- formal
  Consistency level is formal

11.2.10Completeness

Completeness element represents characteristic of documents that is asserted during the course of evaluation and that refers to the completeness of the document. This characteristic refers to the point in the lifecycle of the current version of the document and to our capability to derive useful information from the document. Completeness of a document can be incomplete, draft, final and obsolete. An incomplete document may not be reliable and may contain omissions. A draft document is more reliable and is likely not to contain omissions. A final document is the most reliable state. When the document is obsolete, it may not be a source of high-fidelity information. Evidentiary support from documents that are not final may be contested. Completeness level can be applied to Evidence package.

Superclass

DocumentAttribute

Attributes

- value:CompletenessLevel
  Completeness level, such as incomplete, draft, final, and obsolete.

11.2.11CompletenessLevel (enumeration)

The CompletenessLevel enumeration class defines completeness levels.

Literals

- unknown
  Completeness level is unknown.
- incomplete
  The subject is incomplete.
- draft
  The subject is a draft.
- final
  The subject is final.
- obsolete
  The subject is obsolete.

11.2.12Reliability

Reliability element represents characteristic of documents that is asserted during the course of evaluation and that refers to the reliability of the source of the information contained in the document. This characteristic refers to the level of trust the evaluator confers to the source of the document and therefore to the document itself. Reliability of the document affects the strength of evidentiary support this document provides. The Evidence Metamodel defines 5 levels of reliability.

Superclass

EvidenceAttribute

Attributes

- value:ReliabilityLevel
  Level of reliability of the Document, such as unreliable, not usually reliable, usually reliable, fairly reliable, completely reliable.

11.2.13ReliabilityLevel (enumeration)

The ReliabilityLevel enumeration class defines reliability levels.

Literals

- unknown
  Reliability level is unknown.
- unReliable
  The source is unreliable.
- nonUsuallyReliable
  The source often unreliable.
- usuallyReliable
  The source usually reliable.
- fairlyReliable
  The source is fairly reliable.
- completelyReliable
  The source is completely reliable.

11.2.14ExtendedDocumentProperty

ExtendedDocumentProperty element represents a user-defined characteristic of a document that is asserted during the course of evaluation.

Superclass

DocumentProperty

Constraints

- ExtendedDocumentProperty element shall own at least one TaggedValue describing the meaning of the element.

Semantics

ExtendedDocumentProperty is a user-defined characteristic. Its meaning is represented by the key-value pair of the corresponding TaggedValue element.

ExtendedDocumentProperty characteristic can not be verbalized using the standard vocabulary of the Structured Assurance Case Metamodel. However, the key and value pair may be carefully named to result in meaningful verbalizations for the targeted community in the selected language.

12 Formal Statements

Formal Statements provide the mechanism for representing the elements of meaning involved in the processes of interpretation and evaluation of evidence, and specifically, required for precisely representing assertions and claims.

The two fundamental classes of the Formal Statements are FormalObject and FormalAssertion. A FormalObject is an object of significance, about which information needs to be known or held. Usually a FormalObject corresponds to an Exhibit where the Exhibit element emphasizes the physical object (an instance of the SBVR 'Thing' concept) while a FormalObject emphasizes the associated element of meaning (an instance of the SBVR 'Meaning' concept). A FormalAssertion is a relationship between evidence elements taken as a new assertion/claim that has a distinct, separate existence, a self-contained piece of information that can be referenced as a unit. In the scope of SBVR, such units of information are called facts. However, since the Evidence Metamodel focuses at describing evidentiary support to assurance cases, which involves contestable claims, relationships are interpreted as assertions, rather than facts, which allows contesting them. However, in practice, most of the assertions that may be represented by an evidence model are likely to be within the so-called assumption zone of an assurance case, i.e., be agreed upon facts.

So, an FormalAssertion element represents an assertion involving one or more FormalObjects bound to specific roles associated with the fact type of the assertion. The concepts fact type, role, element is bound to a role are defined in SBVR. In particular, a fact type is defined as a concept that is the meaning of a verb phrase that involves one or more noun concepts and whose instances are all actualities. A role is defined as a noun concept that corresponds to things based on their playing a part, assuming a function or being used in some situation. Specifically, a fact type role characterizes its instances by their involvement in an actuality that is an instance of a given fact type. A RoleBinding element represents an association, linkage, or connection between the FormalObjects that describes their role within the assertion.

Formal Statements are based on some pre-defined conceptual model related to the area for which an assurance case is developed. Such conceptual model can be formally represented as an external ontology or vocabulary. In particular the SACM Evidence Metamodel allows linking an Object element to an SBVR IndividualConcept or SBVR noun concept element and the Assertion element to SBVR fact type element

The Object element is aligned with the SBVR IndividualConcept or the SBVR noun concept while the Assertion element is aligned with the SBVR fact. type. Further, the entire SACM Evidence Metamodel is aligned with the OMG SBVR specification, in such a way that it describes a standard vocabulary related to descriptions of evidence. SBVR rules can be written using this vocabulary to formally describe further properties of evidence. The full SBVR vocabulary for evidence is presented as a non-normative Annex A.

12.1 Formal Objects Class Diagram

The FormalObjects class diagram focuses at objects are they are involved in assertions comprising the fact model underlying an assurance case.

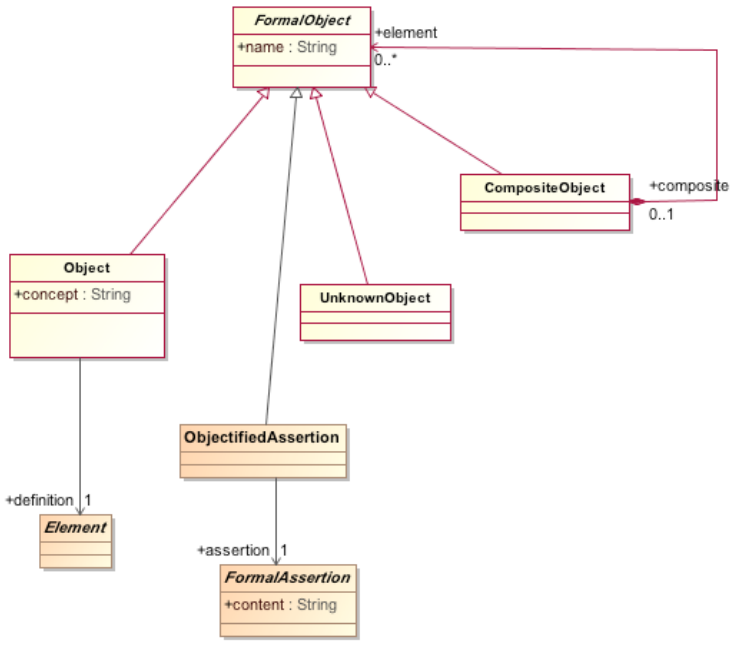

Figure 12.1 - Formal Objects Class Diagram

12.1.1 Object

Object represents a known object that can be involved in assertions constituting the conceptual model underlying an assurance case (formal statements).

Superclass

FormalObject

Attributes

- concept:String
  Designation of the noun concept.

Associations

- definition:MOF::Element
  A link to an entry in an external SBVR vocabulary or an OWL ontology defining the noun concept of the object.

Semantics

Object is an element of meaning. Object shall be used in formal statements underlying an assurance case to represent known subjects of assertions, in particular when more than one assertion refers to the same subject. In some cases, an Object may be accompanied by an Exhibit, which is the only element in the extent of the concept represented by the Object.

12.1.2 UnknownObject

UnknownObject represents an unknown object, existence of which is however is determined by the pattern of relationships in formal statements, and that is involved in assertions constituting the conceptual model underlying an assurance case.

Superclass

FormalObject

Semantics

UnknownObject is an element of meaning. UnknownObject shall be used in formal statementsthe conceptual model underlying an assurance case to represent unknown subjects of assertions, in particular when more than one assertion refers to the same subject. An UnknownObject is not linked to an external noun concept definition (as opposed to an Object element).

12.1.3 CompositeObject

CompositeObject represents a collection of objects that can be involved in assertions constituting the conceptual model underlying an assurance case. CompositeObject can be nested, i.e., a member of a CompositeObject can be another composite object.

Superclass

FormalObject

Associations

- element:FormalObject[0..*]
  Object that is a member of the collection.

Constraints

- CompositeObject shall not be a member of itself, either directly or indirectly through membership in other CompositeObject.

Semantics

CompositeObject is an element of meaning. CompositeObject shall be used in formal statements underlying an assurance case to represent groups of object of assertions, in particular when more than one assertion refers to the same group.

12.1.4 ObjectifiedAssertion

ObjectifiedAssertion represents an objectified assertion, i.e. an assertion that implicitly defines an object that is used in another assertion.

Superclass

FormalObject

Associations

- assertion:FormalAssertionLink to the FormalAssertion being objectified

Semantics

From the formal logic perspective, SACM distinguishes objects from assertions. As a consequence, in order to represent a formal assertion about other assertions the later must be objectified, i.e. represented as a FormalObject that refers to the objectification of the original assertion using the element ObjectifiedAssertion.

12.2 Formal Assertions Class Diagram

The FormalAssertions class diagram focuses at the Assertion as the key element of the formal statements underlying an assurance case.

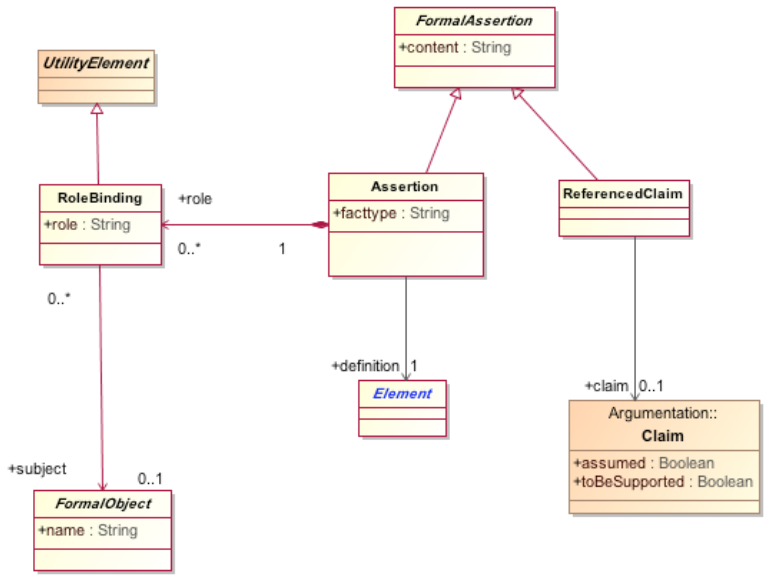

Figure 12.2 - Formal Assertions Class Diagram

12.2.1 Assertion

An Assertion is a relationship involving one or more formal objects, taken as formal proposition that has a distinct, separate existence, a self-contained piece of information that can be referenced as a unit. Assertion is the key constituent of a conceptual model underlying an assurance case. Assertion represents an asserted fact about the subject area for which an assurance case is being developed.

Superclass

FormalAssertion

Attributes

- facctype:String
  Designation of the fact type

Associations

- role:RoleBinding[0..*]
  Set of role bindings that further describe which FormalObject are bound to the roles that are determined by the fact type.
- definition:MOF::Element
  A link to an entry of an external SBVR vocabulary or an OWL ontology defining the fact type of the assertion

Semantics

Assertion is an element of meaning that states existence of a relationship between several individual formal objects. In a formal assurance case, the nature of the relationship is specified through a reference to an external vocabulary, such as an SBVR vocabulary or an OWL ontology. SACM assumes that community of interest for an assurance case will acquire or develop such vocabularies for the corresponding subject area. In a semi-formal assurance case the nature of the relationship can be described informally through a 'content' property. In this case the 'definition' property and the 'facttype' property shall not be used. However the references to the exact FormalObjects through RoleBinding elements can be still stated. The 'content' property of the FormalAssertion element provides the verbalization of the assertion, which is the expression of the assertion in the selected natural language. For informal assurance cases, a ReferencedClaim element can be used, which only contains the verbalization of the claim in a natural language.

12.2.2 ReferencedClaim

ReferencedClaim is an element of meaning that represents an informal assertion about the state of affairs in the subject area about which an assurance case is developed. ReferencedClaim can be linked to a Claim element of the Argumentation part of an assurance case.

Superclass

FormalAssertion

Associations

- claim:Argumentation::Claim[0..1]
  A link to a Claim element in the Argumentation part of an assurance case (if available).

Semantics

ReferencedClaim is an element of meaning that states an assertion about a subject area of an assurance case. ReferencedClaim represents the claim as prose in a selected natural language (formal or informal), without identifying its structure. ReferencedClaim element can represent informal claims (claims not linked to any formal definition of its meaning, such as an ontology developed by some community of meaning) or unstructured claims (where the subjects are not identified).

Usually claims assert existence of a formally defined relationship between several individual subjects and involve several objects bound to specific roles. An Assertion element can be used to capture this structure of a claim in a more formal way. In particular, Assertion element can link the proposition to an external vocabulary or ontology that defines the exact meaning of the proposition, as well as the exact subjects of the proposition.

12.2.3 RoleBinding

A claim usually states existence of a relationship between several individual domain objects and involve several subjects bound to specific roles. RoleBinding element is be used to capture this structure of a claim in a more formal way in the context of an Assurance element representing the claim.

Superclass

UtilityElement

Attributes

- role:String
  Name of the Role in the fact type to which an object is bound.

Associations

- subject:FormalObject[0..1]
  FormalObject that is bound to this Role

Semantics

RoleBinding object is owned by an Assertion object which provides the context, including the definitions of roles and the types of domain objects that can be bound to each role. The formal definition of the relationship represented by an Assertion element is provided by a reference to an external ontology which can be either an SBVR vocabulary of an OWL ontology. This definition shall at a minimum include the definition of roles, to which the RoleBinding elements shall conform. In particular, the 'role' attribute of a RoleBinding shall correspond to a particular role in the formal definition of a relationship. Further, for each role contained in the formal definition of the relationship there shall be exactly one RoleBinding element, in which the 'role' attribute matches the name of the role and the subject matches the allowed type of subject for that role.

SACM allows incremental construction of the conceptual model underlying an assurance case, therefore it allows temporarily unbound roles. A completed Body of Evidence accompanying an Assurance Case shall meet the condition that all RoleBinding element have the corresponding subject of appropriate type.

SACM provides a built-in relation "IsA" between any EvidenceElement and an Object, which states the definition of an EvidenceItem. This mechanism can be used to build the entire formal vocabulary inside the Evidence Model, where the external references can be reduced to a mere handful of meta-meta level concepts (in the extreme case, the only external reference that is needed is the concept "thing," other definitions can, at least in principle, be provided through the "IsA" relationships internal to the Evidence Model. This approach can be used when the external formal vocabulary is not available, and there is a need to use more unified tooling environment.

From the formal logic perspective, SACM distinguishes objects from assertions. As a consequence, in order to represent a formal assertion about other assertions the later must be objectified, i.e. represented as a FormalObject that refers to the of the original assertion using the element ObjectifiedAssertion.

13 Evidence Properties

Evidence Properties defines provenance and timing characteristics of the evidence items and evaluations.

13.1 Custody Class Diagram

The Custody Class Diagram represents various statements related to the Custody of an EvidenceElement. These statements describe the custodians of an evidence element, the locations associated with various events in the lifecycle of the evidence element, as well as the process by which the element was obtained.

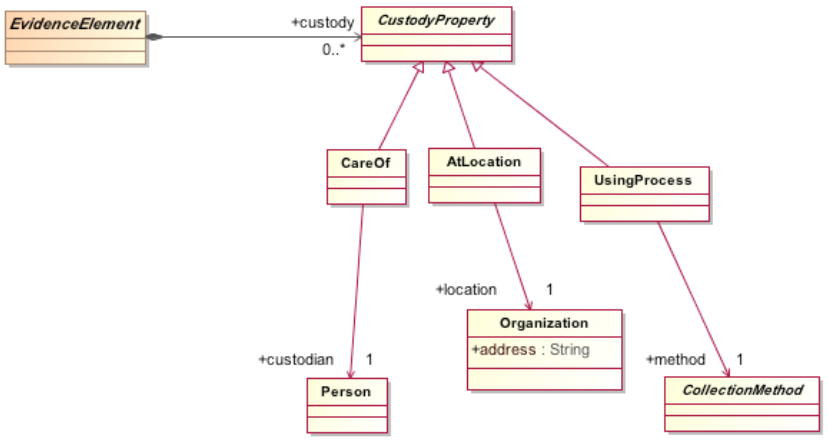

Figure 13.1 Custody class diagram

13.1.1 CustodyProperty (abstract)

CustodyProperty is an abstract class that represents a custody property of an evidence event. Concrete custody properties are defined by subclasses of CustodyProperty.

Superclass

EvidenceProperty

Semantics

CustodyProperty element represents a property of the owner EvidenceEvent object. CustodyProperty element is an abstract class that establishes a relationship between the owner evidence event object and the particular custody property, defined by a particular concrete subclass of the CustodyProperty element and further interpreted by the context of a particular event (as described by a property meaning table of a particular evidence event).

13.1.2 CareOf

CareOf is a characteristic of an EvidenceEvent that specifies the custodian of the associated evidence element.

Superclass

CustodyProperty

Associations

- custodian:Person[1]
  Custodian of the evidence element associated with the subject EvidenceEvent.

Semantics

CareOf element represents a property of the subject EvidenceEvent and its associated EvidenceElement. CareOf element represents the state of affairs that the person identified in the ‘custodian’ attribute of the CareOf object is the custodian of the owner EvidenceElement object (with the additional constraints imposed by the semantics of the owned EvidenceEvent).

13.1.3 AtLocation

AtLocation is a characteristic of an EvidenceEvent that specifies the location of the associated evidence element.

Superclass

CustodyProperty

Associations

- location:Organization[1]
  Location of the evidence event or the associated owner EvidenceElement.

Semantics

AtLocation element represents a property of the owner EvidenceEvent and its associated EvidenceElement. AtLocation element represents the state of affairs that the location identified in location attribute of the AtLocation object is the location of the owner EvidenceElement object (with the additional constraints imposed by the semantics of the owned EvidenceEvent).

13.1.4 UsingProcess

UsingProcess is a characteristic of an EvidenceEvent that specifies the method by which the event was performed.

Superclass

CustodyProperty

Associations

- method:CollectionMethod[1]
  CollectionMethod involved at the owner EvidenceEvent

Semantics

UsingProcess element represents a property of the owner EvidenceEvent. UsingProcess element represents the state of affairs that the CollectionMethod identified in method attribute of the UsingProcess object is the method involved at the owner EvidenceEvent object (with the additional constraints imposed by the semantics of the owned EvidenceEvent).

13.2 EvidenceEvents Class Diagram

The EvidenceEvents Class Diagram describes evidence statements related to the Events that determine the lifecycle of an evidence element. EvidenceEvents set the context for additional timing, provenance and custody properties associated with the subject evidence element. Therefore EvidenceEvents allow representing the entire Chain of Custody of the evidence element. EvidenceEvents statements are owned by the subject evidence element.

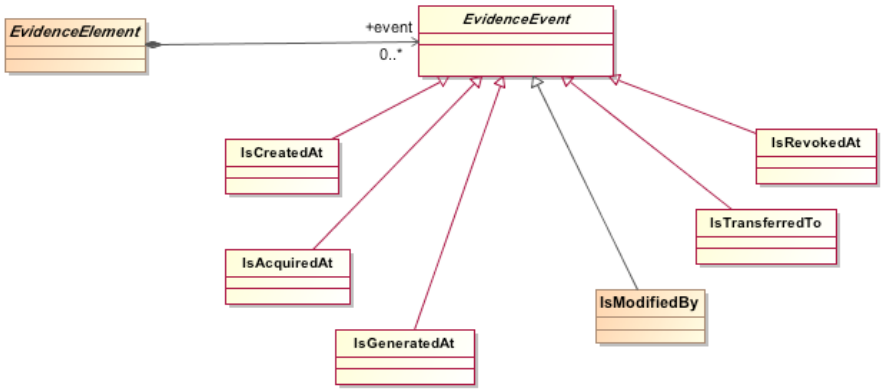

Figure 13.2 - EvidenceEvent Class Diagram

13.2.1 EvidenceEvent (abstract)

EvidenceEvent represents statements related to the events in the lifecycle of an evidence element. The lifecycle of an evidence element is determined by several events, such as Creation, Acquisition or Derivation of the evidence element; Transfer of the evidence element; Modification of the evidence element; Evaluation of the evidence element; and Revocation of the evidence element. Semantics of concrete evidence events is defined for the subclasses of EvidenceEvent element. An EvidenceEvent statement describes a certain characteristic of the subject evidence element. More complex Event statements can be constructed by adding further Timing, Provenance and Custody clauses to EvidenceEvents of the subject evidence element. In particular, the mechanism of EvidenceEvents allows making statements about the time-dependent characteristics of the subject evidence element, since each EvidenceEvent can be the subject of its own timing clause. The entire chain of custody of an evidence element can be established by analyzing the EvidenceEvents of the element. On the other hand, the Timing, Provenance and Custody clauses of the subject evidence element itself (EvidencePropery objects that are directly owned by the EvidenceElement object) state essential characteristics of the EvidenceElement that do not change over time.

Statements about evidence elements can be revoked and updated statements can be made. The ModifiedBy event statement can be used to provide record of the modification elements.

Superclass

EvidenceProperty

Semantics

EvidenceEvent represents statements related to the lifecycle events of the subject EvidenceItem. Further detail of the event are provided by the EvidenceProperty elements owned by the EvidenceEvent. The set of EvidenceEvent owned by an EvidenceItem establishes the chain of custody for the EvidenceItem.

The EvidenceEvent element is an abstract class that establishes a relationship between the subject evidence item and the particular event description with its associated characteristics, defined by a particular concrete subclass of the EvidenceEvent element and its owned properties, such as CustodyProperty, Provenance, and TimingProperty.

13.2.2 IsAcquiredAt

IsAcquiredAt is an Evidence Event that describes an acquisition of an evidence element and thus initiates the lifecycle of the evidence element. Other evidence events that initiate the lifecycle of evidence element are creation of an evidence element and generation of an evidence element. Acquisition emphasizes an event at which custody is established over a pre-existing item.

Superclass

EvidenceEvent

Semantics

IsAcquiredAt element represents a property of the owner EvidenceElement object. IsAcquiredAt element represents the state of affairs that the owner object is acquired. IsAcquiredAt may own further properties establishing additional details about the acquisition event.

Property	Meaning	Verbalization
AtTime	Time of the acquisition	Element *is acquired at* time
EffectiveTime	N/A	
CreatedBy	N/A	
PerformedBy	The stakeholder who acquired the evidence element	Element *is acquired by* stakeholder
ApprovedBy	The person or organization who approved the acquisition.	*Acquisition of* element *is approved by* stakeholder
OwnedBy	Organization which executed acquisition of the evidence element and has custody of the evidence element.	Element *is owned by* stakeholder
CareOf	The custodian of the evidence element within the owner organization.	Person *is custodian of* element
AtLocation	The location of the evidence document at which it was acquired.	Element *is acquired at* location
UsingProcess	The reference to a CollectionMethod object that provides a definition of the process involved in the acquisition.	Element *is acquired using* method

13.2.3 IsCreatedAt

IsCreatedAt is an Evidence Event that describes creation of an evidence element and thus initiates the lifecycle of the evidence element. Other evidence events that initiate the lifecycle of evidence element are acquisition of an evidence element and generation of an evidence element. Creation emphasizes an event by which a primary evidence item comes to existence. Generation emphasizes event by which a secondary (derived) evidence element comes to existence.

Superclass

EvidenceEvent

Semantics

IsCreatedAt element represents a property of the owner EvidenceElement object. IsCreatedAt element represents the state of affairs that the owner object is created. This usually applied to primary evidence elements. IsCreatedAt may own further properties establishing additional details about the creation event.

Property	Meaning	Verbalization
AtTime	Time of creation	Element *is created at* time
EffectiveTime	Effective time of the evidence element	
CreatedBy	N/A	
PerformedBy	The source of the evidence element	Element *is created by* stakeholder
ApprovedBy	The person or organization who approved the creation of the evidence element.	*Creation of* element *is approved by* stakeholder
OwnedBy	Organization which created the evidence element.	Element *is owned by* stakeholder
CareOf	The custodian of the evidence element within the owner organization.	Person *is custodian of* element
AtLocation	The location of the evidence document at which it was created; this location may be different from the location of the organization that created the event.	Element *is created at* location
UsingProcess	The reference to a CollectionMethod object that provides a definition of the process involved in the creation of the document.	Element *is created using* method

13.2.4 IsTransferredTo

IsTransferedTo is an Evidence Event that describes a transfer of an already established evidence element and thus continues the lifecycle of the evidence element. Transfer emphasized change of custody.

Superclass

EvidenceEvent

Semantics

IsTransferedTo element represents a property of the owner EvidenceElement object. IsTransferedTo element represents the state of affairs that the owner object is transferred to a different custody. IsTransferedTo element may own further properties establishing additional details about the transfer event.

Property	Meaning	Verbalization
AtTime	Time of the transfer	Element *is transferred at* time
EffectiveTime	N/A	
CreatedBy	N/A	
PerformedBy	The stakeholder who transferred the evidence element	Element *is transferred by* stakeholder
ApprovedBy	The person or organization who approved the transfer of the evidence element.	*Transfer of* element *is approved by* stakeholder
OwnedBy	Organization which established custody over the evidence element.	Element *is owned by* stakeholder
CareOf	The custodian of the evidence element.	Person *is custodian of* element
AtLocation	The new location of the evidence document after the transfer; this location may be the same as the location of the organization that took custody of the document, however these two locations may be different.	Element *is transferred to* location
UsingProcess	The reference to a CollectionMethod object that provides a definition of the process involved in the transfer of the document.	Element *is transferred using* method

13.2.5 IsModifiedBy

IsModifiedBy is an Evidence Event that describes a modification of an evidence element throughout its lifecycle. Modification event emphasizes changes to the original exhibit or changes in the meaning of the FormalAssertion or EvidenceAssertion, or changes to the ProjectElement. The IsModifiedBy element can be the subject of additional Timing, Provenance and Custody clauses.

Superclass

EvidenceEvent

Semantics

IsModifiedBy element represents a unique modification event throughout its lifecycle of the subject EvidenceElement object. IsModifiedBy element represents the state of affairs that the owner object is modified. IsModifiedBy may include additional clauses that provide further details about the modification event. In particular, an Annontation clause can be used to describe the nature of the modification.

Property	Meaning	Verbalization
AtTime	Time of the modification	Element *is modified at* time
EffectiveTime	N/A	
CreatedBy	N/A	
PerformedBy	The stakeholder who modified the evidence element	Element *is modified by* stakeholder
ApprovedBy	The stakeholder who approved the modification of the evidence element.	*Modification of* element *is approved by* stakeholder
OwnedBy	N/A	
CareOf	The custodian of the evidence element.	Person *is custodian of* element
AtLocation	The location oat which the modification of the evidence element is performed	Element *is modified at* location
UsingProcess	The reference to a method by which the evidence element is modified	Element *is modified using* method

13.2.6 IsRevokedAt

IsRevokedAt is an Evidence Event that describes revocation of an already established evidence element and thus describes the end of the lifecycle of the evidence element. Revocation of an evidence document means that the evidence element is no longer admissible for supporting arguments while it is still available e.g. as an item in an evidence repository. A revoked element may still remain as the subject of assertions stating evidentiary support to some claims. Such relations may need to be evaluated and explicitly negated based on the revocation event. Revocation of an evidence element is stronger than the end of the validation period of an evidence element.

Superclass

EvidenceEvent

Semantics

IsRevokedAt element represents a property of the subject EvidenceElement object. IsRevokedAt element represents the state of affairs that the subject has been revoked. IsRevokedAt element may be the subject of additional properties describing further details about the revocaction event.

Property	Meaning	Verbalization
AtTime	Time of the revocation	Element *is revoked at* time
EffectiveTime	N/A	
CreatedBy		
PerformedBy	The stakeholder who revoked the evidence element	Element *is revoked by* stakeholder
ApprovedBy	The person or organization who approved the revocation of the evidence element.	*Revocation of* element *is approved by* stakeholder
OwnedBy	Organization which established custody over the evidence element, if applicable.	Element *is owned by* stakeholder
CareOf	The custodian of the evidence element.	Person *is custodian of* element
AtLocation	N/A	
UsingProcess	The reference to a CollectionMethod object that provides a definition of the process involved in the revocation of the document.	Element *is revoked using* method

13.2.7 IsGeneratedAt

IsGeneratedAt is an Evidence Event that describes generation of a derived evidence element and thus initiates the lifecycle of the evidence element. Other evidence events that initiate the lifecycle of evidence element are acquisition of an evidence element and creation of an evidence element. Creation emphasizes an event by which a primary evidence item comes to existence. Generation emphasizes event by which a secondary (derived) evidence element comes to existence. Acquisition emphasizes taking custody of a pre-existing item.

Superclass

EvidenceEvent

Semantics

IsGeneratedAt element represents a property of the owner EvidenceElement object. IsGeneratedAt element represents the state of affairs that the owner object is generated. This usually applies to primary evidence elements. IsGeneratedAt may own further properties establishing additional details about the creation event.

Property	Meaning	Verbalization
AtTime	Time of generation	Element *is generated at* time
EffectiveTime	Effective time of the generated evidence element	
CreatedBy	N/A	
PerformedBy	The stakeholder who generated the evidence element	Element *is generated by* stakeholder
ApprovedBy	The person or organization who approved the generation of the evidence element.	*Generation of* element *is approved by* stakeholder
OwnedBy	Organization which executed generation of the evidence element.	Element *is owned by* stakeholder
CareOf	The custodian of the evidence element within the owner organization.	Person *is custodian of* element
AtLocation	The location of the evidence document at which is was generated.	Element *is generated at* location
UsingProcess	The reference to a CollectionMethod object that provides a definition of the process involved in the generation of the document.	Element *is transferred using* method

13.3 Provenance Class Diagram

The Provenance Class Diagram focuses at the Provenance characteristics: who create the evidence element, or who evaluated it, who approved it, and what organization owns the evidence element.

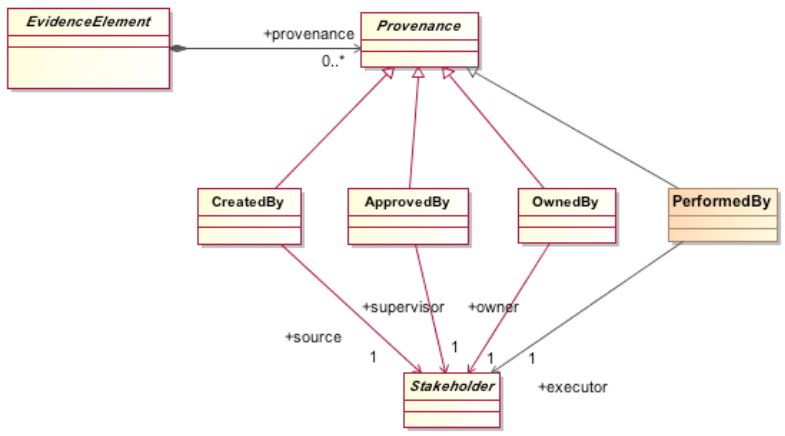

Figure 13.3 - Provenance Class Diagram

13.3.1 Provenance (abstract)

Provenance element is an abstract class that represents any provenance characteristic. In the SACM Evidence Metamodel this element is utilized to specify which elements can have provenance properties. Specific provenance characteristics extend Provenance element.

Superclass

EvidenceProperty

Semantics

Provenance element represents a property of the owner EvidenceElement object or EvidenceAttribute object. This element is an abstract class that establishes a relationship between the owner object and the particular provenance characteristic, defined by a particular concrete subclass of the Provenance element.

13.3.2 CreatedBy

CreatedBy element represents the source of the owner object. The source can be a person or an organization, collectively referred to as an stakeholder.

Superclass

Provenance

Associations

- source:Stakeholder[1]
  The source of the owner object.

Semantics

CreatedBy element represents a property of the owner EvidenceElement object or EvidenceAttribute object. CreatedBy element represents the state of affairs that the owner object was created by the particular stakeholder, defined by stakeholder object. Stakeholder of an evidence object can be a person or an organization.

The characteristic of CreatedBy is expressed by a sentential form "Element *is created by* stakeholder."

13.3.3 ApprovedBy

ApprovedBy element represents the supervisor of the owner object. The subervisor can be a person or an organization, collectively referred to as an stakeholder.

Superclass

Provenance

Associations

- supervisor:Stakeholder[1]
  The supervisor of the owner object.

Semantics

ApprovedBy element represents a property of the owner EvidenceElement object or EvidenceAttribute object. ApprovedBy element represents the state of affairs that the owner object has been approved by the particular stakeholder, defined by stakeholder object. Stakeholder of an evidence object can be a person or an organization.

The characteristic of ApprovedBy is expressed by a sentential form "Element *is approved by* stakeholder."

13.3.4 OwnedBy

OwnedBy element represents the owner the evidence object. The owner can be a person or an organization, collectively referred to as an stakeholder, however in practice, the owner is usually an organization.

Superclass

Provenance

Associations

- owner:Stakeholder[1]
  The owner of the evidence object.

Semantics

OwnedBy element represents a property of the owner EvidenceElement object or EvidenceAttribute object. OwnedBy element represents the state of affairs that the owner object (which is the technical term referring to the fact that the OwnedBy property is owned by some object of EvidenceElement or EvidenceAttribute class) is owned by the particular subject, defined by Stakeholder object. Stakeholder of an evidence object can be a person or an organization.

The characteristic of OwnedBy is expressed by a sentential form "Element *is owned by* stakeholder."

13.3.5 PerformedBy

PerformedBy element represents the provenance clause that states the stakeholder who executes an evidence object. The clause can refer to a person or an organization, collectively referred to as a stakeholder.

Superclass

Provenance

Associations

- executor:Stakeholder[1]
  The executor of the evidence event.

Semantics

PerformedBy element represents a clause of an evidence statement related to the subject EvidenceElement. PerformedBy element represents the state of affairs that the subject event is executed by the particular stakeholder, defined by 'executor' object. Executor of an evidence event can be a person or an organization.

The characteristic of PerformedBy is expressed by a sentential form "Event *is performed by* executor."

13.4 Timing Class Diagram

The Timing Class Diagram focuses at the Timing characteristics: when the evidence element was created, what is its effective date, and until when it is valid.

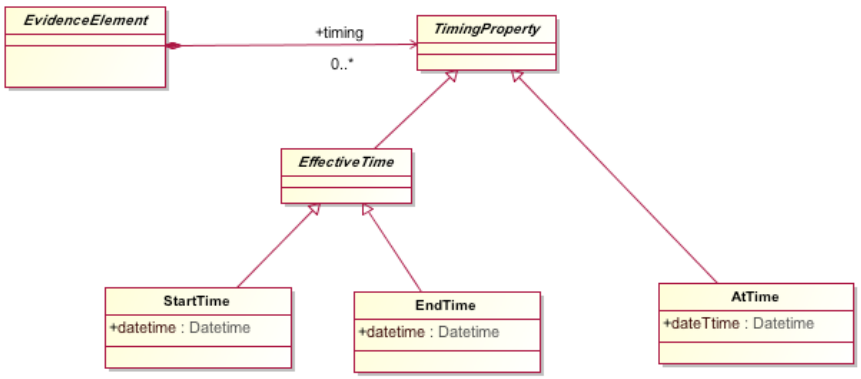

Figure 13.4 - Timing Class Diagram

13.4.1 TimingProperty (abstract)

TimingProperty element is an abstract class that represents any timing characteristic. In the SACM Evidence Metamodel this element is utilized to specify which elements can have timing properties. Specific timing characteristics extend TimingProperty element.

Superclass

EvidenceProperty

Semantics

TimingProperty element represents a property of the owner EvidenceElement object or EvidenceAttribute object. This element is an abstract class that establishes a relationship between the owner object and the particular timing characteristic, defined by a particular concrete subclass of the TimingProperty element.

13.4.2 EffectiveTime (abstract)

EffectiveTime element represents various compound statements that involve a certain time interval during which a certain proposition is asserted to be true (time-dependent assertions involving an “effective “time period). Specific characteristics related to the effective time interval are defined by concrete subclasses of EffectiveTime element.

Superclass

TimingProperty

Semantics

EffectiveTime element represents a statement about the owner EvidenceElement (an object that owns the instance of one of the concerete subclasses of this element). The EffectiveTime element specifies a time interval associated with the subject, during which the subject is asserted to be “effective”. For example, in case of an EvidenceAssertion or a FormalAssertion, this element specifies a time interval at which the corresponding statement is asserted to be true. In case of an EvidenceItem this element specifies the relevant time context in which the element shall be considered.

13.4.3 StartTime

This element represents the start of the effective time interval of the owner evidence object.

Superclass

EffectiveTime

Attributes

- datetime:EDate[1]
  Date starting from which the owner object becomes valid.

Constraints

- One object shall not own more than one StartTime property.
- When object owns StartTime and EndTime, the datetime of the StartTime property shall be earlier than or equal to the datetime of the EndTime property.

Semantics

StartTime element represents a property of the owner EvidenceElement object or EvidenceAttribute object. StartTime element represents the state of affairs that the owner object is valid starting from the datetime stated by the StartTime property.

13.4.4 EndTime

This element represents the end of the effective time interval of the owner evidence object.

Superclass

EffectiveTime

Attributes

- datetime:EDate[1]
  Date after which the owner object ceases to be valid.

Constraints

- One object shall not own more than one EndTime property.
- When object owns StartTime and EndTime, the datetime of the EndTime property shall be later than or equal to the datetime of the StartTime property.

Semantics

EndTime element represents a property of the owner EvidenceElement object or EvidenceAttribute object. EndTime element represents the state of affairs that the owner object is not valid after from the datetime stated by the EndTime property.

13.4.5 AtTime

This element represents the time stamp for the owner evidence object. The context for the timestamp is given by the owner object.

Superclass

TimingProperty

Attributes

- datetime:EDate[1]
  The timestamp associated with the owner object.

Semantics

AtTime element represents a property of the owner EvidenceElement object or EvidenceAttribute object. AtTime element represents the state of affairs that involves an association between the owner object and the datetime stated by the AtTime property.

14 Evidence Evaluation

Evaluation of Evidence involves making certain assertions about evidence items and their relations to the subject area claims.

Evidence Assertions are defined within the Evidence Metamodel and include the following categories:

- Properties of Documents as they are related to the quality of the evidentiary support that may be offered by these documents, such as Primary or secondary document, original or derived document, Consistency, Completeness, Accuracy of the document. These properties are independent on an assurance case for which the evidence is collected.
- Attributes of the evidentiary support, such as Direct or indirect, Relevance, Confidence, Strength and Significance.
- Interpretation of Evidence: what an evidence item "Is", what it "means."
- Nature of evidentiary support: Supports, Challenges.
- Observations and Resolutions.
- Standard of Proof to which evidence is evaluated.

14.1 Evidence Relations Class Diagram

The Evidence Relations Class Diagram provides elements that represent statements of evidentiary support relations between an EvidenceItem, such as an Exhibit and a FormalAssertion.

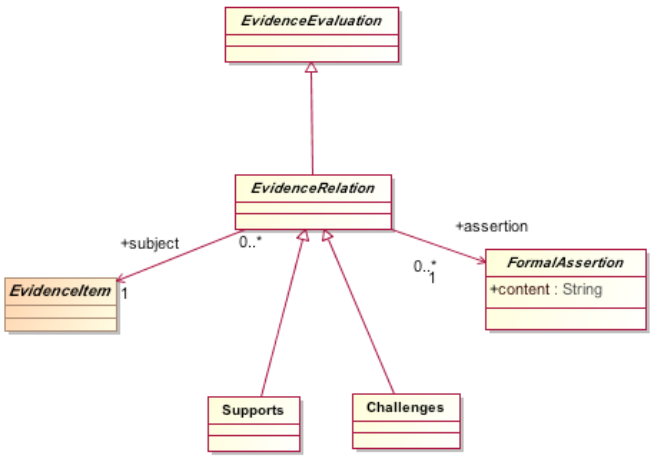

Figure 14.1 - EvidenceRelations Class Diagram

14.1.1 EvidenceRelation (abstract)

EvidenceRelation is an abstract class that represents an evidence relation between one EvidenceItem and one FormalAssertion element. Concrete nature of these relations is defined by the subclasses of the EvidenceRelation element.

Superclass

EvidenceEvaluation

Associations

- subject:EvidenceItem[1]
  The EvidenceItem object, such as an Exhibit or a Document that is the subject of an evidentiary relation to a FormalAssertion object such as a ReferencedClaim.
- assertion:FormalAssertion[1]
  FormalAssertion object that receives an evidentiary relation from the EvidenceItem object.

Constraints

- FormalAssertion shall not receive evidence relation from self.

Semantics

EvidenceRelation is a unit of information generated during evidence evaluation. It represents a relationship between an EvidenceItem and a FormalAssertion objects that is asserted during the evidence evaluation.

14.1.2 Supports

Supports element represents an evidence relation between one EvidenceItem and one FormalAssertion element where the EvidenceItem confers evidentiary support to the FormalAssertion.

Superclass

EvidenceRelation

Semantics

Supports relation is generated during evidence evaluation. It represents a relationship between an EvidenceItem and FormalAssertion objects where the EvidenceItem confers evidentiary support on the claim represented by FormalAssertion. This relationship is verbalized as: "EvidenceItem *supports* FormalAssertion."

14.1.3 Challenges

Challenges element represents an evidence relation between one EvidenceItem and one FormalAssertion element where the EvidenceItem challenges the validity of the FormalAssertion.

Superclass

EvidenceRelation

Semantics

Challenges relation is generated during evidence evaluation. It represents a relationship between an EvidenceItem and a FormalAssertion objects where the EvidenceItem is the so-called counter evidence to the claim represented by the FormalAssertion object, i.e., the EvidenceItem challeges the validity of the domain claim represented by the FormalAssertion. This relationship is verbalized as: "EvidenceItem *challenges* FormalAssertion."

14.2 Evidence Attributes Class Diagram

The EvidenceAttribute Class Diagram defines several concrete characteristics of evidence, introduced during the process of evidence evaluation.

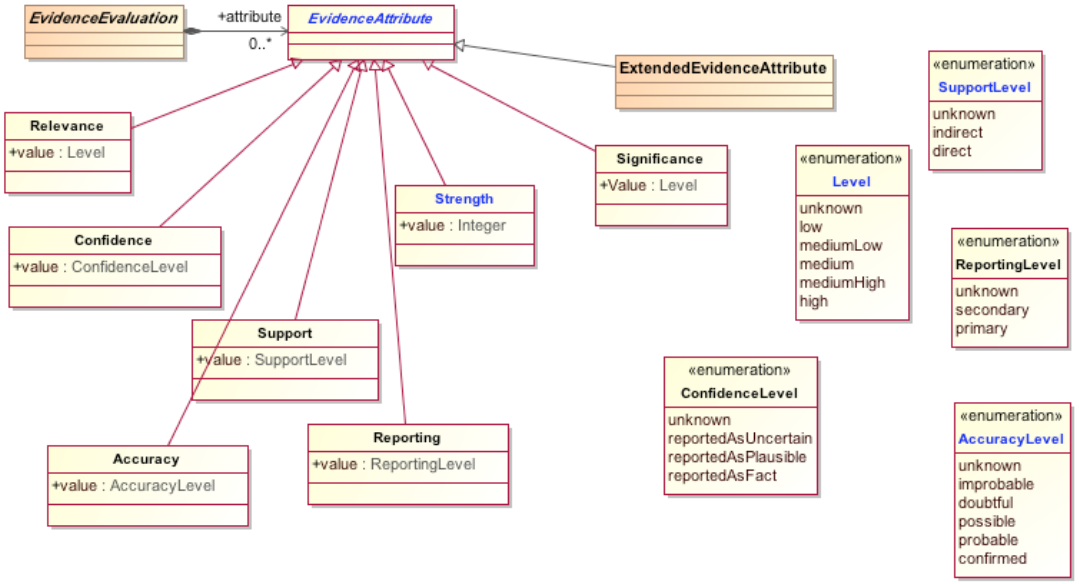

Figure 14.2 - EvidenceAttribute Class Diagram

14.2.1 Support

Support element represents characteristic of the evidence relations that is asserted during the course of evaluation and that refers to the nature of support - direct support vs. indirect support - provided by evidence item to the corresponding claim.

Superclass

EvidenceAttribute

Attributes

- value:SupportLevel
  Level of support (e.g., indirect or direct).

Constaints

- Support element shall not be owned by elements other than EvidenceRelation.

Semantics

Support is an asserted characteristic that potentially can be disputed. Support attribute adds a quality modifier to the EvidenceRelation. To be considered “direct evidence,” an evidence item must be sufficient on its own to make a statement without the necessity of introducing other records. Direct evidence specifically makes a statement. Indirect evidence (or

circumstantial evidence as it is often called) requires introduction of other pieces of information to complete a statement. Direct evidence has more weight than indirect. Whenever additional records are drawn to supply missing information there is a chance for error. Because of that, less weight is assigned to indirect evidence.

Support characteristic is verbalized as follows:

- “EvidenceItem directly supports FormalAssertion,”
- “EvidenceItem indirectly supports FormalAssertion,”
- “EvidenceItem directly challenges FormalAssertion,”
- “EvidenceItem indirectly challenges FormalAssertion.”

14.2.2 SupportLevel (enumeration)

SupportLevel enumeration specifies the support level.

Literals

- unknown
  The directness is unknown.
- indirect
  Evidence relation provides indirect support the Assertion.
- direct
  Evidence relation provides direct support the Assertion.

14.2.3 Reporting

Reporting element represents characteristic of the evidence relations that is asserted during the course of evaluation and that refers to the reporting level of the relationship - primary or secondary reporting - provided by evidence item to the corresponding claim.

Superclass

EvidenceAttribute

Attributes

- value:ReportingLevel
  Reporting level of the evidence relation, such as secondary or primary.

Constaints

- Reporting element shall not be owned by elements other than EvidenceRelation.

Semantics

Reporting level is an asserted characteristic that potentially can be disputed. Reporting level attribute adds a quality modifier to the EvidenceRelation. This characteristic refers to the quality of information provided as evidence. For example, the record is primary if it was made at or near the time of the event, by someone in a position to know firsthand (such as an eyewitness). Alternatively, a record is considered primary if it was made in writing by an officer charged by law, canon, or bylaws with creating an accurate record. Primary information carries more weight than secondary

information. Various communities disagree on whether primary information remains primary when copied. For example, the legal community states that a primary record becomes secondary when copied. Other communities focus at the information rather than the record, from which standpoint the primary information remains primary when copied.

Reporting characteristic is verbalized as follows: "EvidenceItem *is a primary record* **of** FormalAssertion," "EvidenceItem *is a secondary record* **of** FormalAssertion."

14.2.4 ReportingLevel (enumeration)

ReportingLevel enumeration specifies the reporting levels.

Literals

- unknown
  The level of reporting is unknown.
- secondary
  EvidenceItem is a secondary record of FormalAssertion.
- primary
  EvidenceItem is a primary record of FormalAssertion.

14.2.5 Accuracy

Accuracy element represents characteristic of evidence relations that is asserted during the course of evaluation and that refers to the perceived accuracy of the information contained in the document. This characteristic refers to the level of trust the evaluator confers to the information contained in the document. Accuracy of the information affects the strength of evidentiary support this document provides. The Evidence Metamodel defines 5 levels of accuracy.

Superclass

DocumentAttribute

Attributes

- value: Level
  Accuracy level of the Document, such as improbable, doubtful, possible, probable, confirmed.

14.2.6 AccuracyLevel (enumeration)

The AccuracyLevel enumeration class defines accuracy levels.

Literals

- unknown
  Accuracy level is unknown.
- improbable
  The information is improbable.
- doubtful
  The information is doubtful.
- possible
  The information is possible.

- probable
  The information is probable.
- confirmed
  The information is confirmed.

14.2.7 Confidence

Confidence element represents characteristic of the evidence relations that is asserted during the course of evaluation and that refers to the confidence level of the relationship - whether information is reported as uncertain, plausible or as a fact. Confidence affects the strength of evidentiary support provided by evidence item to the corresponding claim.

Superclass

EvidenceAttribute

Attributes

- value:ConfidenceLevel
  Condidence level of the evidence relationship, such as reportedAsUncertain, reportedAsPlausible, reportedAsFact.

Semantics

Confidence element is owned by EvidenceEvaluation as appropriate. Confidence characteristic is owned by EvidenceEvaluation object as appropriate. Each subclass of EvidenceEvaluation defines specific constraints regarding the meaning of Confidence in this context. Relevance is an asserted characteristics that potentially can be disputed as opposed to EvidenceProperty, which represents fundamental properties of the EvidenceElement, and AdministrativeElement. Confidence element includes the relevance level.

14.2.8 ConfidenceLevel (enumeration)

The ConfidenceLevel enumeration class defines confidence levels.

Literals

- unknown
  Accuracy level is unknown.
- reportedAsUncertain
  The information is reported as uncertain.
- reportedAsPlausible
  The information is reported as plausible.
- reportedAsFact
  The information is reported as Fact.

14.2.9 Significance

Significance element represents characteristic of the evidence relations that is asserted during the course of evaluation and that refers to the significance level of the relationship - whether information that is reported as indirect support of the claim is significant to establish the truth of the claim. Significance affects the strength of evidentiary support provided by evidence item to the corresponding claim.

Superclass

EvidenceAttribute

Attributes

- value:Level
  Significance level, such as low, mediumLow, medium, mediumHigh, or high.

14.2.10 Relevance

Relevance element represents characteristic of the evidence relations that is asserted during the course of evaluation and that refers to the relevance level of the relationship - whether information that is reported as indirect support of the claim is relevant to establish the truth of the claim. Relevance affects the strength of evidentiary support provided by evidence item to the corresponding claim.

Superclass

EvidenceAttribute

Attributes

- value:Level
  Relevance level, such as low, mediumLow, medium, mediumHigh, or high.

14.2.11 Level (enumeration)

Level enumeration provides generic 5-level qualitative measure. Level enumeration is utilized to evaluate relevance and significance of evidentiary support.

Literals

- unknown
  The level is unknown.
- low
  The level is low.
- mediumLow
  The level is medium low.
- medium
  The level is medium.
- mediumHigh
  The level is medium high.
- high
  The level is high.

14.2.12 Strength

Strength element represents characteristic of the evidence relations that is asserted during the course of evaluation and that refers to the reporting level of the relationship - the strength of the support relation - provided by evidence item to the corresponding claim.

Superclass

EvidenceAttribute

Attributes

- value:Integer
  The strength of support: 0 to 100

Constraints

- Strength value shall be an integer value that is greater than or equal to 0 and less than or equal to 100.

Semantics

Strength is an asserted characteristic that potentially can be disputed. Strength attribute adds a quality modifier to the EvidenceRelation. This characteristic refers to the quality of information provided as evidence. Strength can be a primary characteristic provided during the evaluation, or can be derived from other qualitative characteristics.

Strength characteristic is verbalized as follows: "EvidenceItem *supports* FormalAssertion *with strength* 50," "EvidenceItem *challenges* FormalAssertion *with strength* 10."

14.2.13ExtendedEvidenceAttribute

ExtendedEvidenceAttribute element represents a user-defined characteristic of the evidence relations that is asserted during the course of evaluation.

Superclass

EvidenceAttribute

Constraints

- ExtendedEvidenceAttribute element shall own at least one TaggedValue describing the meaning of the element.

Semantics

ExtendedEvidenceAttribute is a user-defined characteristic. Its meaning is represented by the key-value pair of the corresponding TaggedValue element.

ExtendedEvidenceAttribute characteristic can not be verbalized using the standard vocabulary of the Structured Assurance Case Metamodel. However, the key and value pair may be carefully named to result in meaningful verbalizations for the targeted community in the selected language.

14.3 EvidenceInterpretation Class Diagram

The EvidenceInterpretation Class Diagram defines several EvidenceEvaluation elements that allow assertions regarding the interpretation of EvidenceElements.

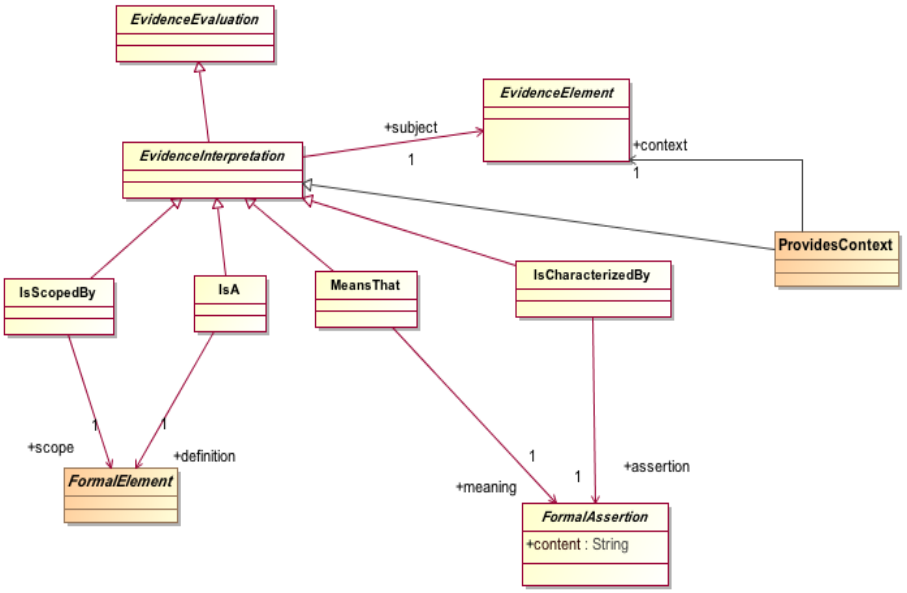

Figure 14.3 - EvidenceInterpretation Class Diagram

14.3.1 EvidenceInterpretation (abstract)

EvidenceInterpretation is an abstract class that represents a relation between one EvidenceElement and one FormalElement. Concrete nature of these relations is defined by the subclasses of the EvidenceInterpretation element. The subtypes of EvidenceInterpretation are: “IsA,” “MeansThat,” “IsCharacterizedBy,” and “IsScopedBy.” The following statements are examples of evidence interpretation:

- “This document is a test report.”
- “This document is characterized by the fact that it was produced by an independent testing laboratory.”
- “This metric is scoped by the client subsystem.”
- “This metric means that the architecture quality of the Client subsystem is high.”

Superclass

EvidenceEvaluation

Associations

- subject:EvidenceElement[1]
  The EvidenceElement that is the subject of interpretation.

Semantics

EvidenceInterpretation is a unit of information generated during evidence evaluation. It represents a relationship between an EvidenceItem and a FormalElement object that is asserted during the evidence evaluation.

14.3.2 IsA

IsA statement represents a fundamental relation between one EvidenceElement and one FormalElement which defines the general concept for the subject EvidenceElement. The actual concept can be given by reference to an external formal vocabulary or ontology. The following statements are examples of IsA statements:

- "This metric is a McCabe's Cyclomatic Complexity Metric."
- "This report is a penetration testing report."

Superclass

EvidenceInterpretation

Associations

- definition:FormalElement[1]
  The formal FormalElement that is the general concept of the subject of the relation.

Constraints

- The subject of the IsA relation shall not be its definition.

Semantics

The IsA element asserts a state of affairs that the EvidenceElement, identified as the subject element of the IsScopedBy element, has a general concept represented by the FormalElement that is identified as the definition of the IsA element.

This characteristic is verbalized as follows: "EvidenceElement *is a* FormalElement."

14.3.3 MeansThat

MeansThat represents a fundamental relation between one EvidenceElement and one FormalAssertion element which defines the meaning of the source EvidenceElement. The actual assertion is given by reference to an external formal vocabulary or ontology. The Evidence Metamodel limits the scope of meaning to a single fact type instance. Alternatively an informal ReferencedClaim can be used. The following statements are examples of Means:

- "This metric means that the quality of the system is medium-low."
- "This report means that the preliminary hazard list has been identified correctly."

Superclass

EvidenceInterpretation

Associations

- meaning:FormalAssertion[1]
  FormalAssertion element

Constraints

- The subject of the MeansThat relation shall not be its meaning.

Semantics

The MeansThat element asserts a state of affairs that the EvidenceElement, identified as the 'subject' of the MeansThat element, has meaning represented by the FormalAssertion that is identified as the 'meaning' of the MeansThat element.

This characteristic is verbalized as follows: "EvidenceElement *means that* FormalAssertion is true."

14.3.4 IsCharacterizedBy

IsCharacterizedBy represents a relation between one EvidenceElement and one FormalAssertion element which defines a characteristic of the subject EvidenceElement. The actual fact type is given by reference to an external formal vocabulary or ontology. The following statements are examples of IsCharacterizedBy:

- "This metric is characterized by its accuracy being confirmed," or alternatively
- "The accuracy of this metric is confirmed."

Superclass

EvidenceInterpretation

Associations

- assertion:FormalAssertion[1]
  The FormalAssertion that characterizes the subject EvidenceElement.

Semantics

The IsCharacterizedBy element asserts a state of affairs that the EvidenceElement, identified as the 'subject' of the IsCharacterizedBy element, is characterized by an assertion, in which the subject is bound to one of the roles, and which is represented by the FormalAssertion that is identified as the 'assertion' of the IsCharacterizedBy element.

This characteristic is verbalized as follows: "EvidenceElement *is characterized by* FormalAssertion."

14.3.5 IsScopedBy

IsScopedBy statement represents a relation between one EvidenceElement and one FormalElement that defines the scope of the subject EvidenceElement. The actual concept is given by reference to an external formal vocabulary or an ontology. The following statements are example of IsScopedBy: "This metric is scoped by the client subsystem."

Superclass

EvidenceInterpretation

Associations

- scope:FormalElement[1]
  The FormalElement that is the scope of the subject of the relation.

Constraints

- The subject of the IsScopedBy relation shall not be its scope.

Semantics

“Scope” is defined as the area covered by a given activity or subject, which can be interpreted in either physical or logical sense. The IsScopedBy element asserts a state of affairs that the EvidenceElement, identified as the ‘subject’ of the IsScopedBy element, is delimited by the FormalElement that is identified as the ‘scope’ of the IsScopedBy element. The FormalElement may represent an individual concept, an abstract concept or an assertion.

This characteristic is verbalized as follows: “EvidenceElement *is scoped by* FormalElement.”

14.3.6 ProvidesContext

ProvidesContext element represents statements that assert that a certain evidence element provides a context for the interpretation of another evidence element.

Superclass

EvidenceInterpretation

Associations

- context:EvidenceElement[1]The element that is asserted to represent the context for the subject

Semantics

ProvidesContext element establishes a relationship between two evidence elements where the ‘context’ evidence element (usually an EvidenceGroup) provides a context for the ‘subject’ evidence element (usually a FormalAssertion, or an EvidenceAssertion). A 'context' is defined as the set of evidence elements (including evidence items, evidence assertions and even project elements) that are important for understanding of the ‘subject’ evidence element. The concept of a context is more informal than the related concept of 'scope' (see 'IsScopedBy' assertion).

14.4 Evidence Observations Class Diagram

The EvidenceObservations Class Diagram defines several EvidenceEvaluation elements that allow assertions regarding the dependencies between EvidenceRelation elements or conflicts between FormalAssertions.

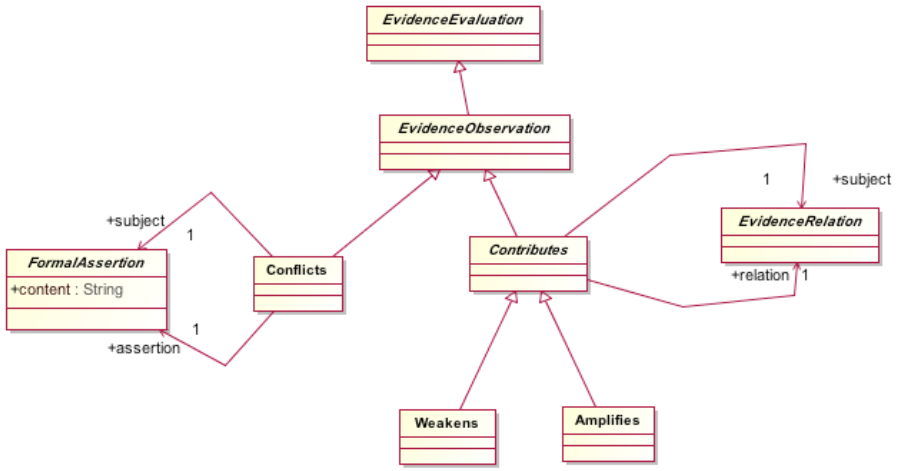

Figure 14.4 - EvidenceObservations Class Diagram

14.4.1 EvidenceObservation (abstract)

EvidenceObservation is an abstract class that asserts existence of a dependency between two evidence relations or conflict between two domain assertions. These conflicts need to be further addressed during the rest of the evidence evaluation process.

Superclass

EvidenceEvaluation

Semantics

The EvidenceObservation element asserts existence of a conflict in evidentiary support. The concrete subclasses of the EvidenceObservation element define the exact nature of the conflict.

14.4.2 Conflicts

Conflicts element asserts existence of a conflict between two domain assertions. For example, one may assert that the claim that “Bob is married to Alice” conflicts the claim that “Bob is single” and conflicts the claim that “Bob is married to Eve.” These conflicts need to be further addressed during the rest of the evidence evaluation process.

Superclass

EvidenceObservation

Associations

- subject: FormalAssertion[1]
  The subject FormalAssertion
- assertion: FormalAssertion[1]
  The object FormalAssertion

Semantics

The Conflicts element asserts a state of affairs that the FormalAssertion-1, identified as the assertion1 of the Conflicts element, is in conflict with FormalAssertion that is identified as the assertion2 of the Conflicts element. Conflict here is defined as a state of doubt that both assertion can be true at the same time. The conflict needs to be resolved by clarifying the meaning of the assertions, negating or refuting the supporting evidence to one of the assertion, etc.

This characteristic is verbalized as follows: "FormalAssertion-1 *conflicts* FormalAssertion-2"

14.4.3 Contributes (abstract)

Contributes element asserts dependency between two EvidenceRelation elements. For example, let's assume the following evidentiary relationships:

Exhibit A *supports* (referenced) claim that "Bob *is married to* Alice"

Exhibit A *challenges* claim "Bob *is single*"

We can observe that **the** claim "Bob *is married to* Alice" *conflicts with* **the** claim "Bob *is single*"

Let's further assume the following evidentiary relationship:

Exhibit C *supports* claim Exhibit A *is likely a forgery*

We can observe that:

The evidence assertion Exhibit C *supports* claim "Exhibit A *is likely a forgery*" *weakens* support *given by* **the** Exhibit A *to* **the** claim "Bob *is married to* Alice"

At the same time we do not directly assert that:

Exhibit C *challenges* **the** claim "Bob *is married to* Alice"

Evidence observations help capture dependencies between related claims and thus facilitate evaluation of evidence.

Superclass

EvidenceObservation

Associations

- subject: EvidenceRelation[1]
  The subject EvidenceRelation
- relation: EvidenceRelation[1]
  The object EvidenceRelation

Constraints

- The subject and object EvidenceRelation elements shall not be the same.

Semantics

The Contributes element asserts existence of a dependency in evidentiary support. The concrete subclasses of the Contributes element define the exact nature of the dependency.

14.4.4 Weakens

Weakens element asserts that the subject EvidenceRelation weakens another EvidenceRelation2. This statement has a different meaning than a statement about existence of an evidence item that (directly) challenges the FormalAssertion involved in the EvidenceRelation2. Weakens relation may imply a conflict between the subject FormalAssertion that is involved in the subject EvidenceRelation and FormalAssertion2. In that case the evidence in support of the subject FormalAssertion is not relevant to FormalAssertion2.

Superclass

Contributes

Semantics

The Weakens element asserts a state of affairs that the EvidenceRelation-1, identified as the 'subject' of the Weakens element, weakens EvidenceRelation-2 that is identified as the 'relation' of the Weakness element. The Weakens statement asserts a negative contribution made by one EvidenceEvaluation to another EvidenceEvaluation.Weakens may imply a conflict between the 'subject' FormalAssertion-1 that is identified as assertion of EvidenceRelation-1 and FormalAssertion-2 that is identified as assertion of EvidenceRelation-2.

This characteristic is verbalized as follows: "Evidentiary support *to* FormalAssertion-1 *weakens* evidentiary support *to* FormalAssertion-2", where the statement "Evidentiary support *to* a FormalAssertion C1" is an objectified assertion **that** there is an evidence item E1 **that** *supports* the FormalAssertion C1".

14.4.5 Amplifies

Amplifies element asserts that the subject EvidenceRelation amplifies another EvidenceRelation2. This statement has a different meaning than the statement asserting existence of an evidence item that (directly) supports the FormalAssertion2 that is involved in the EvidenceRelation2. Amplifies relation may imply a coupling between the subject FormalAssertion and the FormalAssertion2. In that case the evidence in support of the subject FormalAssertion may be relevant to the FormalAssertion.

Superclass

Contributes

Semantics

The Amplifies element asserts a state of affairs that the EvidenceRelation-1, identified as the subject, amplifies EvidenceRelation-2 that is identified as the relation of the Amplifies element. The Amplifies statement asserts a positive contribution made by one EvidenceEvaluation to another EvidenceEvaluation. Amplifies may imply a coupling between FormalAssertion-1 that is identified as assertion of EvidenceRelation-1 and FormalAssertion-2 that is identified as assertion of EvidenceRelation-2.

This characteristic is verbalized as follows: "Evidentiary support *to* the subject FormalAssertion *amplifies* evidentiary support *to* FormalAssertion2"

14.5 Evidence Resolutions Class Diagram

The EvidenceResolutions Class Diagram defines several EvidenceEvaluation elements that allow assertions regarding the resolutions to EvidenceEvaluation elements for the purpose of explaining the conflicts between FormalAssertions. The Evidence Metamodel provides three options: Negate EvidenceRelation, Refute a FormalAssertion, and Resolve

EvidenceObservation (which implies existence of conflicting claims). The purpose of EvidenceResolutions is to provide necessary clarifications explaining the existence of counterevidence to the key domain claims. At the end of evidence evaluation EvidenceResolutions should build a clear picture showing that the preponderance of evidence to the required domain claims in case of real conflicts, and resolving the conflicts that are determined by imprecise formulation of claims and incorrect interpretation of evidence.

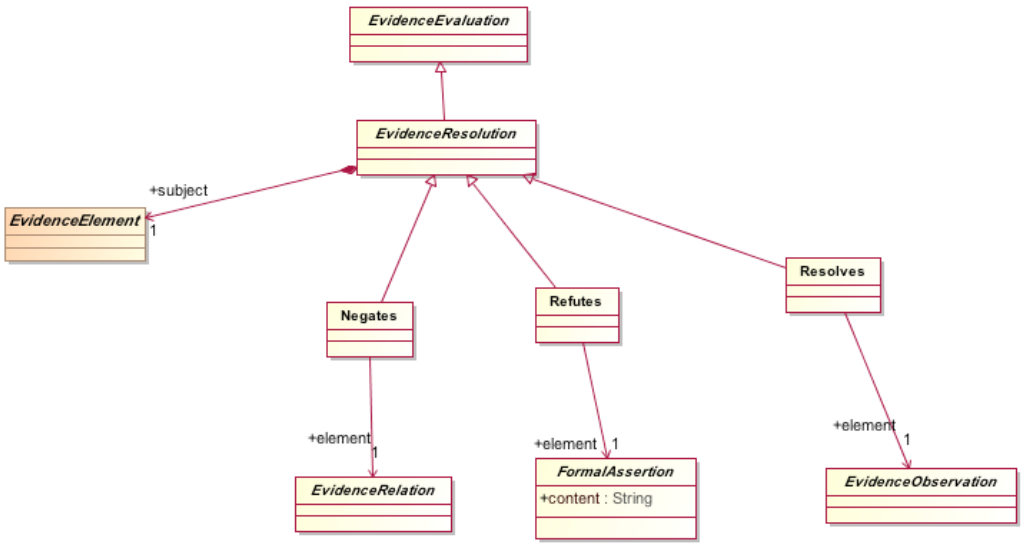

Figure 14.5 - EvidenceResolutions Class Diagram

14.5.1 EvidenceResolution (abstract)

EvidenceResolution represents statements that assert resolution to the conflics between two evidence assertions either directly or indirectly by refuting some evidence assertion or negating some evidence relation.

Superclass

EvidenceEvaluation

Associations

- subject:EvidenceElement[1]
  The subjectevidence element for the resolution, i.e. the evidence element negates, resoles or refutes other evidence elements.

Constraints

- The EvidenceElement that is resolved by the EvidenceResolution (as defined by one of the concrete subclasses of the EvidenceResolution class) shall not be a member of the context either directly or indirectly through membership in other contexts.

Semantics

The EvidenceResolution element asserts resolution of a conflict in evidentiary support. The concrete subclasses of the EvidenceResolution element define the exact nature of the resolution.

14.5.2 Negates

Negates element asserts negation of an EvidenceRelation. For example, one may want to assert that “there is insufficient evidence to support the fact that the weakness in line 256 can be exploited by an outside attacker.” Negation indirectly refutes the FormalAssertion by claiming that the evidentiary support to the FormalAssertion is indirect, weak, unreliable, not coming from credible sources.

Superclass

EvidenceEvaluation

Associations

- element:EvidenceRelation[1]
  The EvidenceRelation being negated.

Semantics

The Negates element asserts negation of evidentiary support to a certain FormalAssertion. The Rationale element that is owned by the Negates object provides a readable explanation to the negation. The context property may refer to a particular set of EvidenceAttribute or Document that describes the context for negation. Negates element addresses the existing evidentiary support to a certain FormalAssertion.

14.5.3 Refutes

Refutes element asserts direct refutation of a FormalAssertion. For example, one may want to assert that “the weakness in line 256 cannot be exploited by an outside attacker because of the existence of proper architecture controls.” Refutes element asserts direct refutation of a FormalAssertion. Context of the refutation is important, because the conflicting claims with strong evidentiary support need to de identified.

Superclass

EvidenceEvaluation

Associations

- element:FormalAssertion[1]
  The FormalAssertion being refuted.

Semantics

The Refutes element asserts direct refutation of a certain FormalAssertion. The Rationale element that is owned by the Refutes object provides a readable explanation to the refutation. The context property may refer to a particular set of EvidenceAttribute or Document that describe the context for refutation. Refutes element emphasizes the claims with strong evidentiary support conflicting to the FormalAssertion being refuted.

14.5.4 Resolves

Resolves element asserts resolution of a conflict between two FormalAssertion. For example, one may want to assert that “the fact that Bob is married to Alice is not in conflict with the fact that Bob is single because they refer to non-overlapping time intervals.” Resolves element asserts resolution to a conflict between two FormalAssertion. Context of the resolution is important, because the precise interpretation of the seemingly conflicting claims with strong evidentiary support need to de identified.

Superclass

EvidenceEvaluation

Associations

- element:EvidenceObservation[1]
  The EvidenceObservation being resolved (usually a Conflicts relation between two FormalAssertion).

Semantics

The Resolves element asserts resolution of a conflict between two FormalAssertion. The Rationale element that is owned by the Resolves object provides a readable explanation to the resolution. The context property may refer to a particular set of EvidenceAttribute or EvidenceInterpretation that describe the context for resolution. Resolves element emphasizes the claims with strong evidentiary support are not conflicting after precise interpretation.

15 Administration

This chapter describes the elements of the SACM Evidence Metamodel that are involved in managing evidence, exchanging units of evidence and related concerns. The elements described in this chapter organize instances on Evidence Metamodel, which can be referred to as an Evidence Model. In particular, this chapter defines the root object of Evidence Models - the EvidenceContainer. This element contains other objects in an evidence project and constitutes a unit of exchange using the Evidence Metamodel as the protocol.

15.1 Project Class Diagram

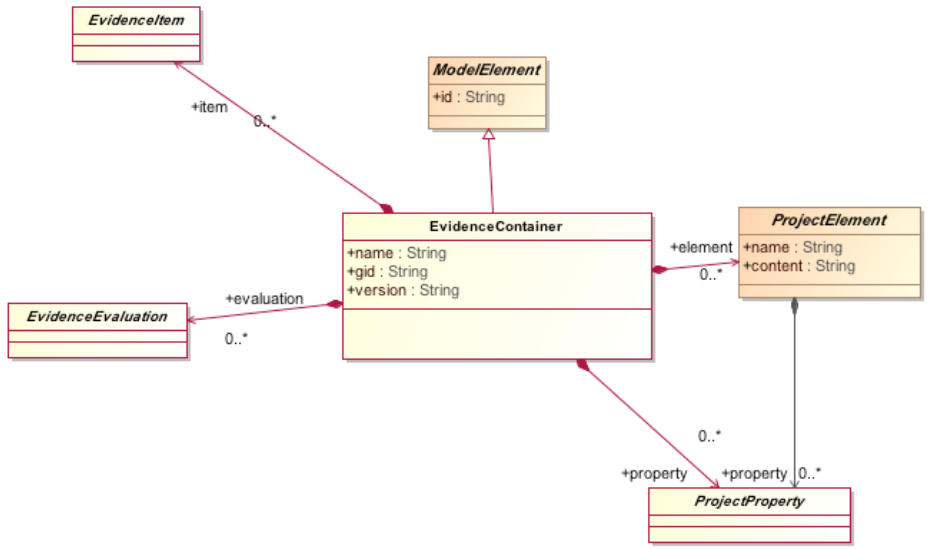

Figure 15.1 - Project Class Diagram

15.1.1 ProjectElement (abstract)

ProjectElement represents the auxiliary elements of the Evidence Metamodel that are involved in the statements related to managing evidence collection, interpretation, evaluation, and exchange processes.

Superclass

EvidenceElement

Attributes

- name:String
  Name of the ProjectElement.
- content:String
  Statement in a selected language that is the description of the content of the element.

Associations

- property:ProjectProperty[0..*]
  Properties of the ProjectElement - zero or more predicates to the main clause in which the current element is the subject.

Semantics

The properties of a ProjectElement make assertions regarding the current element (use the current element as the subject of the corresponding clauses). Therefore, the following properties for a ProjectElement can be readily interpreted in the above way:

- *DependsOn* when a subject element is an Activity (for example, verbalized as " Activity A2 *depends on* Activity A1")
- *HasRoleIn* when the subject element is a Stakeholder (for example, verbalized as " Bob is president *of* organization SupplierCorporation")
- *Satisfies* when a subject element is an Activity (for example, verbalized as" Activity A2 *satisfies* project objective Perform Search")

All ProjectProperties clauses directly owned by a ProjectElement shall be interpreted with the ProjectElement as the main subject. For example, "Person Researcher *depends on* activity Perform Search **and** *satisfies* project objective Find evidence"

15.1.2 EvidenceContainer

EvidenceContainer element is the root object of the SACM Evidence Metamodel instances. This object owns EvidenceItem, and EvidenceEvaluation elements, as well as other ProjectElement related to the processes of evidence identification, collection, interpretation, evaluation, and management.

Superclass

EvidenceElement

Attributes

- name:String name of the EvidenceContainer.
- gid:String Globally unique identifier of the EvidenceContainer.
- version:String version of the EvidenceContainer.

Association

- item:EvidenceItem[0..*] List of evidence items.
- evaluation:EvidenceEvaluation[0..*] List of evaluations.
- element:ProjectElement[0..*] List project elements (objectives, activities, requests, methods, stakeholders).
- property:ProjectProperty[0..*] List of project property clauses.

Constraints

- EvidenceContainer shall not be the object of the requiresContainer relation owned by the EvidenceContainer, either directly or indirectly through requiresContainer of other EvidenceContainers.

- Any EvidenceContainer that is the object of the requiresContaienr relation shall be available for exchange.
- [Completeness of the evidence container with respect to required evidence containers] Any Element that is referenced by any of the Element defined in the package (i.e., that are members of the lists item, evaluation, or element of the EvidenceContainer) shall be also defined in the EvidenceContaienr or in one of the EvidenceContainers that are referred to as objects of the requiresContaienr relation either directly or indirectly. An Element is referenced if it is an object of an EvidenceProperty or an EvidenceEvaluation.
- EvidenceProperty, EvidenceEvaluation, EvidenceRequest, EvidenceAction, ProjectObjective elements shall not be referenced across evidence containers.

Semantics

EvidencePackage element is the root object of an instance of the Evidence Metamodel (which can be referred to as Evidence Model). A single EvidenceContainer is a unit of exchange of evidence information. All Element defined in an EvidenceContainer are exchanged together as part of the EvidenceContainer. Elements that are referenced shall be either present in the EvidenceContainer or in one of the EvidenceContaienrs that is specified as required for the EvidenceContainer. The Evidence Metamodel does not require completeness of the closure of all required packages.

The properties of the EvidenceContainer element make assertions regarding the current container (use the current container as the subject of the corresponding clauses). Therefore, the following properties for an EvidenceContainer can be readily interpreted in the above way:

- RequiresContainer (for example, verbalized as "**the** EvidenceContainer *requires* EvidenceContainer X1")
- ContainerConsistency (for example, verbalized as "elements **of the** EvidenceContainer *are interpreted formally*")
- ContainerCompleteness (for example, verbalized as" **the** EvidenceContainer *is in draft state*")
- CompliesTo (for example, verbalized as "**the** EvidenceContainer *complies to* Resolved Counter Evidence proof standard")

All ProjectProperties clauses directly owned by an EvidenceContainer shall be interpreted with the EvidenceContainer as the main subject. For example, "**the** EvidenceContainer *depends on* evidentiary support *rendered by* Exhibit E1 *to* Claim Testing is completed"

15.2 ProjectElements Class Diagram

ProjectElements Class Diagram defines several auxiliary elements that are used in various statements as predicate clauses for some main clause in which the subject. is some evidence element. The elements defined at this class diagram are collectively referred to as the project elements. They are required to express various evidence statements related to evidence collection, evaluation and evidence management.

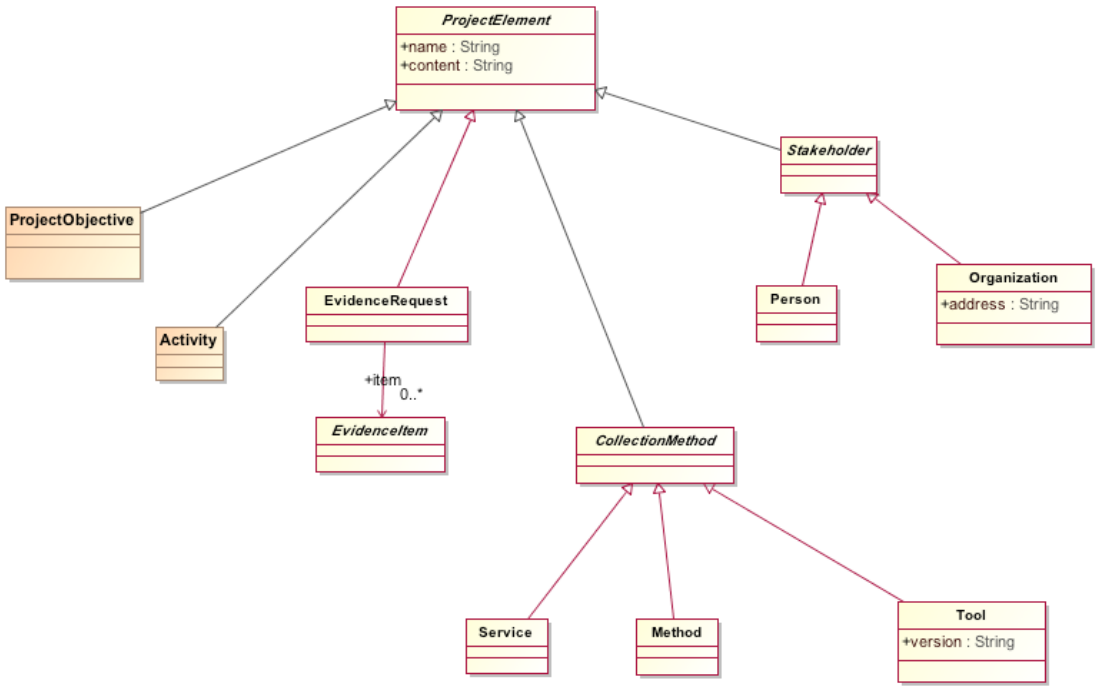

Figure 15.2 - ProjectActivities Class Diagram

15.2.1 Activity

Activity element represents an individual task that either needs to be performed during an evidence-related effort (planning purposes), or has been performed during the effort (tracking purposes). Activity element may own several properties which define its relationship to other Activities (dependencies), to ProjectObjective elements (motivation), to required CollectionMethods (required resources) and to associated EvidenceRequest elements (for the purpose of planning collection of certain exhibits). Activity element may also own Provenance and Timing properties.

Superclass

AdministrativeElement

Associations

- property:ActivityProperty[0..*]
  Additional properties of this activity
- provenance:Provenance[0..*]
  Provenance of this activity
- timing:TimingProperty[0..*]
  Timing properties of this activity

ProjectObjective

ProjectObjective element represents an individual project requirement of an evidence-related effort. Specific activities can be added that satisfy there requirements.

Superclass

AdministrativeElement

Attributes

- text:String
  Text of the project objective (prose)

Semantics

The text attribute of the ProjectObjective element specifies the project objective. In addition, the ProjectObjective element may own Description element.

15.2.2 EvidenceRequest

EvidenceRequest represents a placeholder for an EvidenceItem to be collected during the evidence-related effort.

Superclass

ProjectElement

Associations

- item:EvidenceItem[0..*]
  Evidence items that satisfy the request.

15.2.3 CollectionMethod (abstract)

CollectionMethod is an abstract class that represents evidence collection methods as elements of meaning in the Evidence Model.

Superclass

Object

Semantics

Defined by concrete subclasses and further through a reference to an external vocabulary of ontology.

15.2.4 Service

Service element represents an evidence collection capability that can be provided by a person or an organization.

Superclass

CollectionMethod

Associations

- tool:RequiresTool[0..*]
  Tool that is required by the service.

Semantics

RequiresTool is an owned property of Service. This property represents a state of affairs that the tool identified as tool attribute of the RequiresTool object owned by Service object, is required by the Service object. Further detail may be provided through the Provenance and Timing attribute. Multiple OwnedBy attribute specify multiple providers of the Service.

15.2.5 Method

Method element represents an evidence collection method that can be applied by a person or an organization. The scope of a Method may be creation, acquisition, and generation of evidence elements, transfer of evidence element, revocation of evidence elements, evaluation of evidence elements.

Superclass

CollectionMethod

Associations

- tool:RequiresTool[0..*]
  Tool that is required by the method.

Semantics

RequiresTool is an owned property of Method. This property represents a state of affairs that the tool identified as tool attribute of the RequiresTool object owned by Method object, is required by the Method object. Further detail may be provided through the Provenance and Timing attribute. Multiple OwnedBy attribute specify multiple providers of the Method.

15.2.6 Tool

Tool element represents an automated evidence collection or evidence generation capability that can be licensed by a person or an organization.

Superclass

CollectionMethod

Attibutes

- version:String[1]
  Designation of the version of the tool

15.2.7 Stakeholder (abstract)

Stakeholder is an abstract class that represents a Person or an Organization as they participate in the statements related to evidence.

Superclass

ProjectElement

Semantics

The Evidence Metamodel indirectly defines several roles in which stakeholders are involved in evidence statements, such as Provenance statements and Custody statements. These roles include the "source" of an evidence item or an evidence assertion, the "supervisor" of an evidence assertion, the "owner" of an evidence item, the ‘executor’ of an evidence event and the "custodian" of an evidence item. This vocabulary facilitates exchange of structured statements related to evidence. Additional roles related to the affiliation of a stakeholder in some Organization can be defined by the corresponding community of interest. These roles can be used in HasRoleIn statements and exchanged informally, as the value of the ‘role’ attribute. On the other hand, formal statements related to stakeholders and their roles can be represented using the mechanism of Formal Statements. The fact type “stakeholder *has* role *with respect to* evidence item” can be formally defined outside of the Evidence Metamodel and then referred to for the purpose of constructing formal statements related to stakeholders.

15.2.8 Person

An individual that can be the source of evidence items in various roles defined by the Evidence Metamodel. A person may be affiliated with an Orgnaization.

Superclass

Stakeholder

Associations

- affiliation:HasRoleIn[0..1]
  Affiliation of the Person with an Organization

Semantics

HasRoleIn is an owned property of Person. This property represents a state of affairs that the Person identified as organization attribute of the HasRoleIn object owned by Person object, is the organization with which the Person is affiliated in the role identified as the ‘role’ attribute of the HasRoleIn object. Further detail may be provided through the Provenance and Timing attribute. For example, EffectiveTime property is added specifies the effective period of affiliation. Person may be affiliated with multiple organizations.

15.2.9 Organization

An organization that can be the source of evidence items in various roles defined by the Evidence Metamodel. Organization may be affiliated with another Organization.

Superclass

Stakeholder

Attributes

- address:String
  The address of the Organization

Associations

- affiliation:HasRoleIn[0..1]
  Affiliation of the Organization with parent Organization

Constraints

- Organization shall not be affiliated with self, either directly or indirectly.

Semantics

HasRoleIn is an owned property of Organization. This property represents a state of affairs that the Organization-2 identified as organization attribute of the HasRoleIn object owned by Organization-1 object, is the organization with which the Organization-1 is affiliated in the role identified as the 'role' attribute of the HasRoleIn object. Further detail may be provided through the Provenance and Timing attribute. For example, EffectiveTime property is added specifies the effective period of affiliation. Organization may be affiliated with multiple other organizations.

15.3 ProjectProperties Class Diagram

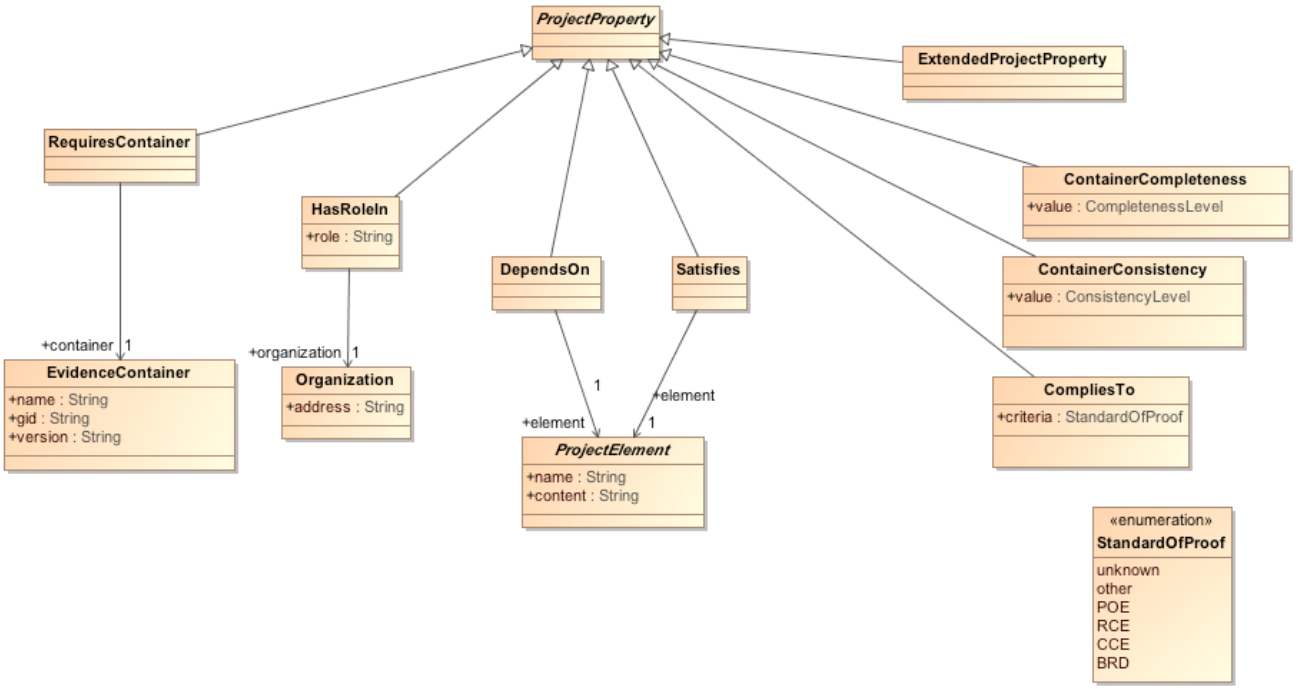

Figure 15.3 ProjectProperties class diagram

15.3.1 ProjectProperty (abstract)

ProjectProperty represents statements related to the structure of ProjectElement. These statements are predicate clauses where the main clause describes some project element. The subject of the ProjectProperty clause is a ProjectElement.

Superclass

EvidenceProperty

Semantics

Defined by concrete subclasses

15.3.2 Satisfies

Satisfies element represents a relationship between the owner project element and another project element that is identified as the element attribute of the Satisfies element. The Satisfies element is a clause where the main subject is the ProjectElement that owns the current element. For example, this clause can be used to specify that a certain Activity satisfies a certain ProjectObjective in an evidence-related effort.

Superclass

ProjectProperty

Associations

- element:ProjectElement[1]
  Project element (such as a ProjectObjective) that is satisfied by the subject project element.

Semantics

Satisfies element represents a state of affairs that the subject project element object satisfies another ProjectElement (such as a ProjectObjective) identified as the 'element' attribute of the Satisfies element.

15.3.3 HasRoleIn

An owned property of Person and Organization

Superclass

ProjectProperty

Attributes

- role:String
  The role in which Person or Organization is affiliated with another Organization.

Associations

- organization:Organization[1]
  Organization with which the subject ProjectElement (such as Person or Organization) is affiliated in the given role.

Constraints

- ProjectElement shall not be affiliated with self, either directly or indirectly.

15.3.4 DependsOn

DependsOn element represents a relationship between the owner project element and another project element that is identified as the element attribute of the DependsOn element. DependsOn element is a clause where the main subject is the ProjectElement that owns the current element. For example, this clause can be used to specify dependencies between Activities in an evidence-related effort.

Superclass

ProjectProperty

Associations

- element:ProjectElement[1]
  Project element that the subject element depends on.

Constraints

- ProjectElement shall not depend on self, either directly or indirectly.

Semantics

DependsOn element represents a state of affairs that the subject project element depends on another project element identified as the ‘element’ attribute of the DependsOn element.

Dependency of one ProjectElement on another can have various meanings. The SACM Evidence Metamodel does not provide a normative enumeration of the nature of dependency. However, should an author of a SACM document desire so, a TaggedValue mechanism shall be used for this purpose with a tag 'natureofdependency'

15.3.5 StandardOfProof (enumeration)

The StandardOfProof enumeration defines the values of the standard of proof criteria for evidence evaluation.

Literals

- unknown
  Standard of Proof unknown
- other
  Standard of proof other than those explicitly enumerated
- POE
  Preponderance of Evidence
- RCE
  Resolved Counter Evidence
- CCE
  Clear and Convincing Evidence
- BRD
  Beyond Reasonable Doubt

Semantics

There are well-defined “Standards of proof,” such as:

- Preponderance of evidence (POE), also known as the balance of the probabilities. The standard is met if the proposition is more likely to be true than not true. This standard is required in most civil cases.
- Resolved Counter Evidence (RCE) this standard is met if all the evidence points in the same direction and anything to the contrary must be resolved. This is a stricter standard than the preponderance of evidence, where even a slight tipping of the scale is sufficient.
- Clean and Convincing Evidence (CCE) this standard is met if it is substantially more likely than not that the proposition is in fact true. This is a lesser requirement than “proof beyond a reasonable doubt,” which requires that the proposition be close to certain of the truth, but a stricter requirement than proof by “preponderance of the evidence,” which merely requires that the proposition asserted seem more likely true than not.

- Beyond the reasonable doubt (BRD) this standard is met if the proposition being presented is proven to the extent that there is no “reasonable doubt” in the mind of a reasonable person that the proposition is true. There can still be a doubt, but only to the extent that it would not affect a “reasonable person’s” belief that the proposition is true.

15.3.6 RequiresContainer

RequiresContainer is an owned property of EvidenceContainer element. This element represents a statement asserting that the subject EvidenceContainer requires another evidence container for the resolution of some references.

Superclass

ProjectProperty

Associations

- container:EvidenceContainer[1]
  EvidenceContainer that is required for the resolution of some references in the subject evidence container.

Constraints

- RequiresContainer element shall not be owned by any ProjectElement object
- subject EvidenceContainer shall not be the ‘container’ of the requiresContainer relation, either directly or indirectly.

Semantics

RequiresContainer property represents a state of affairs that the subject EvidenceContainer requires another evidence container for the resolution of some references. This property contributes to the completeness constraint of the EvidenceContainer. This is a commitment to the set of evidence containers that need to be processed together.

15.3.7 ContainerConsistency

ContainerConsistency element is a counterpart of the Consistency property of Documents. ContainerConsistency clause makes an assertion about the subject EvidenceContainer regarding the level of formality of the element of the container. In combination with other container properties, such as ContainerCompleteness and CompliesTo, this clause determines capability to interpret the elements of this container. Consistency of an EvidenceContainer can be informal, semiformal and formal.

Superclass

ProjectProperty

Attributes

- value:ConsistencyLevel
  asserted Consistency level of the elements of the EvidenceContainer, such as informal, semi-formal, and formal.

15.3.8 ContainerCompleteness

ContainerCompleteness element is a counterpart of the Completeness property of Documents. ContainerCompleteness clause makes an assertion about the subject EvidenceContainer regarding the level of completeness of the element of the container. In combination with other container properties, such as ContainerConsistency and CompliesTo, this clause determines capability to interpret the elements of this container. Completeness of an EvidenceContainer can be incomplete, draft, final and obsolete.

Superclass

ProjectProperty

Attributes

- value:CompletenessLevel
  asserted Completeness level of the elements of the EvidenceContainer, such as incomplete, draft, final and obsolete.

15.3.9 CompliesTo

CompliesTo clause makes an assertion about the subject EvidenceContainer regarding the standard of proof used for the evaluation of evidence in the EvidenceContainer. In combination with other container properties, such as ContainerConsistency and ContainerCompleteness, this clause determines capability to interpret the elements of this container. Completeness of an EvidenceContainer can be incomplete, draft, final and obsolete.

Attributes

- criteria:StandardOfProof
  Standard of Proof used for evaluation of evidence in the subject container.

15.3.10ExtendedProjectProperty

ExtendedProjectProperty element represents a user-defined characteristic documents that is asserted during the course of evaluation for the project elements in the subject container.

Superclass

ProjectProperty

Constraints

- ExtendedProjectProperty element shall own at least one TaggedValue informally describing the meaning of the element.

Semantics

ExtendedProjectProperty is a user-defined characteristic. Its meaning is represented by the key-value pair of the corresponding TaggedValue element.

ExtendedProjectProperty characteristic can not be verbalized using the standard vocabulary of the Structured Assurance Case Metamodel. However, the key and value pair may be carefully named to result in meaningful verbalizations for the targeted community in the selected language.

Annex A - SBVR Vocabulary for Evidence

(non-normative)

This chapter presents the full concepts catalog for the SACM Evidence Metamodel as a business vocabulary represented in SBVR Structured English which is described in the OMG's specification for SBVR.

A.1 Key concepts

This section defines the key concepts of the SACM Evidence Metamodel.

Evidence Element

General concept: Element

Definition: *identifiable* element **of the** body of knowledge collected as part of an evidence-related effort.

Note: Three categories of Evidence Element are Evidence Item (things provided as evidence and their meanings, such as claims), Evidence Event (an occurrence in the life cycle of an Evidence Item) and Evidence Evaluation (various asserted relations between Evidence Element, and asserted characteristics of Evidence Element, including Evidence Evaluation).

Reference schema: *global* id **of** Evidence Element

Evidence Property

General concept: Element

Definition: *essential* characteristic **of an** evidence element.

Note: evidence property represents fundamental characteristics of evidence elements

Note: some evidence property are indirectly associated with evidence element via evaluation attribute

Concept type: Characteristic

Reference schema: *global* **id of the** Evidence Element **that** *is* **the** subject **of the** Evidence Property

Evaluation Attribute

General concept: Element

Definition: *asserted* state of affairs *related to* **the** evidence element

Concept type: Characteristic

Reference schema: *global* id **of the** Evidence Element **that** *is* **the** subject **of the** Evaluation Attribute

Evidence Item

General concept: Evidence Element

Definition: Thing **that** *confers evidentiary support to* claim

Note: Evidence Item represents material things, including documents and records, as well as elements of meaning, such as propositions, that confer evidentiary support to claims (which are propositions).

Note: Evidence Item is a category of Evidence Element. Other categories include Evidence Event and Evidence Evaluation

Reference schema: id **of** Evidence Item

Exhibit

General concept: Evidence Item

Definition: *Material* Thing **that** *confers evidentiary support to* claim

Note: The main category of an exhibit is a document which is a direct expression of some meaning. Other exhibits are representations of various material objects that are not direct expressions of meaning, and their meaning and relation to claim is usually subject to interpretation (and may require additional backing)

Source: American Heritage Dictionary [‘Exhibit’]

Concept type: thing

Reference schema: name **of** Exhibit

Exhibit *is called* Name

Definition: state of affairs **that an** exhibit *has* **a** Name.

Concept type: state of affairs

Reference schema: name **of** Exhibit

Exhibit *has* url

Definition: state of affairs **that an** exhibit *is represented by* **a** url.

Synonym: url **of** Exibit.

Note: this property assumes that the exhibit is a web resource

Concept type: state of affairs

Document

General concept: Evidence Item

Definition: **A** thing **that** *is* **a** *direct expression of* meaning

Description: 1. A written or printed paper that bears the original, official, or legal form of something and can be used to furnish decisive evidence or information

2. A writing that contains information

3. a piece of work created with an application, as by a word processor

4. something, especially a material substance such as a coin bearing a revealing symbol or mark, that serves as proof or evidence (American Heritage Dictionary)

Source: American Heritage Dictionary [‘Document’]

Concept type: thing

Reference schema: name **of** Document

Meaning

General concept: Evidence Item

Definition: what is meant by a word, sign, statement, or description; what someone intends to express or what someone understands

Note: any elements of meaning that are associated with objects presented as evidence or otherwise involved in the evidence collection.

Source: **based on** Semantics of Business Vocabularies and Business Rules [‘Meaning’]

Formal Object

General concept: Meaning

Definition: Meaning **that** *is* **a** noun concept

Note: any elements of meaning that is a noun concept associated with objects presented as evidence or otherwise involved in the evidence collection.

Note: Formal Object corresponds to things in the subject area of the evidence-related effort

Reference schema: name **of a** Formal Object

Formal Assertion

General concept: Meaning

Definition: Meaning **that** *is* **a** proposition

Note: An evidence assertion can be defined in an informal way or can be a formal meaning.

Note: Usually Formal Assertion involves Formal Objects and corresponds to state of affairs in the subject area of the evidence-related effort

Source: **based on** Argumentation Metamodel ['Claim']

Concept type: claim

Reference schema: content **of** Formal Assertion

Evidence Event

General concept: Evidence Element

Definition: Event **that** *determines the life cycle of* **an** Evidence Item

Description: Evidence Events are: Creation, Acquisition, Derivation, Transfer, Evaluation, and Revocation

Reference schema: id **of an** Evidence Event

Evidence Evaluation

General concept: Evidence Element

Definition: Assertion **that** *establishes characteristics* **of** Evidence Element

Note: Establishing evidentiary support that a set of documents provides to the given claim requires evaluation of the documents and its relations to the claims, including the detection of challenges to the claim, conflicts, and contradictions.

Note: Evidence Evaluation corresponds to an Event in the life-cycle of Evidence Element

Reference schema: id **of an** Evidence Evaluation

A.2 Exhibits

This section defines properties of exhibits and documents.

Exhibit_1 *is part of* Exhibit_2

Definition: state of affairs **that** exhibit_1 *is part of* exihibit_2.

Concept type: state of affairs

Exhibit *is expressed in* Media

Definition: state of affairs **that** exhibit *is expressed using* Media.

Example: tablet is expressed in stone

Concept type: state of affairs

Exhibit *is electronically represented as* Bytestream

Definition: state of affairs **that** exhibit *is electronically represented as* stream **of** bytes.

Electronic representation **of** Exhibit *has format* Format

Definition: state of affairs **that** exhibit *is electronically represented using* format.

Electronic representation **of** Exhibit *has size* Size

Definition: state of affairs that the electronic representation of an exhibit has given size.

Document *has* Title

Definition: state of affairs that the string Title is the full title of the Document.

Concept type: state of affairs

Document *is based on* Evidence Item

Definition: state of affairs that Document is derived from Evidence Item.

Synonym: Evidence Item is the source of Document.

Concept type: state of affairs

Document *has* Version

Definition: state of affairs that string Version is the designation of the version of Document

Note: This assumes certain life-cycle of a document and existence of one or more artifacts with the same name and title, but with different content (and therefore expressing different meaning). Within the Evidence Metamodel, each Document has a unique id, so the version allows identification of the physical document and represents the situation where several Document items represent the snapshots of the same physical document at different phases of the life-cycle.

Concept type: state of affairs

Document *is expressed in* Language

Definition: state of affairs that the meaning of the document is expressed in vocabulary that is expressed in Language.

Concept type: state of affairs

Language *is primary in* Document

Definition: state of affairs that Language is primary in Document.

Note: This assumes that document is expressed in multiple languages. Primary language is one used to express the key parts of the document

Document *is releasable to* Community

Definition: state of affairs that Document can be released to members of the Community.

Note: this property is an element of governance: it is permitted that the document is released to the set designated as Community

Concept type: element of governance

Document *is classified as* Security Classification

Definition: state of affairs that Document is marked with Security Classification.

Concept type: state of affairs

A.3 Formal Assertions

Domain Claim

Definition:

Source: **based on** Software Assurance Evidence Metamodel (10.1.2) [‘ReferencedClaim’]

Concept type: Concept

Reference schema: id **of an** Evidence Element

Formal Object

Definition:

Source: **based on** Software Assurance Evidence Metamodel (10.2.1) [‘Formal Object’]

Concept type: Concept

Reference schema: id **of an** Evidence Element

Object

Definition:

Source: **based on** Software Assurance Evidence Metamodel (10.2.2) [‘Object’]

Concept type: Concept

Reference schema: id **of an** Evidence Element

Unknown Subject

Definition: A KDM model that represents facts about the user interface of the existing software system

Source: **based on** Software Assurance Evidence Metamodel (10.2.3) [‘Unknown Subject’]

Concept type: Concept

Reference schema: id **of an** Evidence Element

Composite Subject

Definition:

Source: **based on** Software Assurance Evidence Metamodel (10.2.4) [‘Composite Subject’]

Concept type: Concept

Reference schema: id **of an** Evidence Element

Composite Subject *includes* Domain Object

Definition:

Concept type: Facttype

Assertion

Definition: A proposition that is related to the area for which an assurance case is developed.

Description: A formal assertion is a proposition that describes a state of affairs for which an assurance case is developed. This proposition uses the vocabulary that is imported from the semantic community involved in the subject area within which the evidence is collected. Formal assertions for evidence collection represent the asserted facts as part of the fact model corresponding to the body of evidence. Fact model is an SBVR term.

American Heritage Dictionary: Something declared or stated positively, often with no support or attempt at proof

Note: The term 'fact' is avoided because of the connotation with 'real' occurrences. Formal assertions can represent contradicting or conflicting propositions. The goal of the evidence-related effort is to establish the truth of certain propositions. During the course of the evidence collection and analysis project, various assertions may be considered.

Note: Formal assertion is an instance of a fact type, a proposition that is formalized as an atomic formulation that binds to individual things

Source: **based on** Semantics of Business Vocabularies and Rules ['Fact']

Concept type: meaning

Assertion *involves* Domain Object *in role* Subject Role

Definition:

Concept type: Facttype

Subject Role

Definition:

Concept type: Concept

A.4 Evidence Evaluation

A.4.1 Evidence Relations

Evidence Item *supports* Subject Assertion

Definition: state of affairs **that** evidence item *supports* formal assertion.

Concept type: state of affairs

Evidence Item *challenges* Subject Assertion

Definition: an evidence judgment that an evidence item contradicts a formal assertion.

Concept type: Evidence judgment

Support

Definition: An objectification of an evidence judgment that an evidence item supports a formal assertion

General concept: evidence relation

Contradiction

Definition: An objectification of an evidence judgment that an evidence item contradicts a formal assertion

Concept type: evidence relation

Evidence Relation

Definition: An objectification of an evidence judgment that an evidence item supports a formal assertion

Source: **based on** Software Assurance Evidence Metamodel (10.2.2) ['Evidence Relation']

General concept: evidence judgment

Reference schema: id **of an** Evidence Element

A.4.2 Evidence Observations

Subject Assertion$_1$ *conflicts with* Subject Assertion$_2$

Definition:

Concept type: evidence observation

Evidence Relation$_1$ *contributes to* Evidence Relation$_2$

Definition:

Concept type: evidence observation

Evidence Relation$_1$ *weakens* Evidence Relation$_2$

Definition:

Concept type: evidence observation

Evidence Relation$_1$ *amplifies* Evidence Relation$_2$

Definition:

Concept type: evidence observation

Conflict

Definition: objectification of the state of affairs that a Subject Assertion conflicts with another Subject Assertion

General concept: evidence observation

Contribution

Definition: objectification of the state of affairs that a Subject Assertion contributes to another Subject Assertion

General concept: evidence observation

Evidence Observation

Definition:

Source: based on Software Assurance Evidence Metamodel (10.2.2) ['Evidence Observation']

General concept: evidence judgment

Reference schema: id **of an** Evidence Element

A.4.3 Evidence Resolutions

Rationale *negates* Evidence Relation

Definition:

Concept type: evidence resolution

Rationale *refutes* Subject Assertion

Definition:

Concept type: evidence resolution

Rationale *resolves* Evidence Observation

Definition:

Concept type: evidence resolution

Evidence Resolution

Definition:

General concept: evidence evaluation

A.4.4 Document Attributes

Originality

Definition:

Concept type: Document Attribute

Document *is original*

Definition:

Concept type: Originality

Document *is derivative*

Definition:

Concept type: Originality

Document *is of unknown originality*

Definition:

Concept type: Originality

Consistency

Definition:

Concept type: Document Attribute

Document *has formal consistency*

Definition:

Concept type: Consistency

Document *has semi-formal consistency*

Definition:

Concept type: Consistency

Document *has informal consistency*

Definition:

Concept type: Consistency

Document *has unknown consistency*

Definition:

Concept type: Consistency

Reliability Level

Definition:

Concept type: Document Attribute

Document *is completely reliable*

Definition:

Concept type: Reliability Level

Document *is fairly reliable*

Definition:

Concept type: Reliability Level

Document *is usually reliable*

Definition:

Concept type: Reliability Level

Document *is not usually reliable*

Definition:

Concept type: Reliability Level

Document *is unreliable*

Definition:

Concept type: Reliability Level

Document *is of unknown reliability*

Definition:

Concept type: Reliability Level

Completeness

Definition:

Concept type: Document attribute

Document *is final*

Definition:

Concept type: Completeness

Document *is obsolete*

Definition:

Concept type: Completeness

Document *is draft*

Definition:

Concept type: Completeness

Document *is incomplete*

Definition:

Concept type: Completeness

Document *is of unknown completeness*

Definition:

Concept type: Completeness

Document Attribute

Definition:

Concept type: Concept

Document *has* Document Attribute

Definition:

Concept type: Facttype

A.4.5 Evidence Attributes

Reporting Level

Definition:

Concept type: Evidence Attribute

Evidence Evaluation *is primary*

Definition:

Concept type: Reporting Level

Evidence Evaluation *is secondary*

Definition:

Concept type: Reporting Level

Evidence Evaluation *is of unknown reporting level*

Definition:

Concept type: Reporting Level

Support Level

Definition:

Concept type: Evidence Attribute

Evidence Evaluation *is direct*

Definition:

Concept type: Support Level

Evidence Evaluation *is indirect*

Definition:

Concept type: Support Level

Evidence Evaluation *is of unknown support level*

Definition:

Concept type: Support Level

Significance

Definition:

Concept type: Evidence Attribute

Evidence Evaluation *has high significance*

Definition:

Concept type: Significance

Evidence Evaluation *has medium high significance*

Definition:

Concept type: Significance

Evidence Evaluation *has medium significance*

Definition:

Concept type: Significance

Evidence Evaluation *has medium low significance*

Definition:

Concept type: Significance

Evidence Evaluation *has low significance*

Definition:

Concept type: Significance

Evidence Evaluation *has unknown significance*

Definition:

Concept type: Significance

Relevance

Definition:

Concept type: Evidence Attribute

Evidence Evaluation *has high relevance*

Definition:

Concept type: Relevance

Evidence Evaluation *has medium high relevance*

Definition:

Concept type: Relevance

Evidence Evaluation *has medium relevance*

Definition:

Concept type: Relevance

Evidence Evaluation *has medium low relevance*

Definition:

Concept type: Relevance

Evidence Evaluation *has low relevance*

Definition:

Concept type: Relevance

Evidence Evaluation *has unknown relevance*

Definition:

Concept type: Relevance

Accuracy Level

Definition:

Concept type: Evidence Attribute

Evidence Evaluation *has high accuracy*

Definition:

Concept type: Accuracy Level

Evidence Evaluation *has medium high accuracy*

Definition:

Concept type: Accuracy Level

Evidence Evaluation *has medium accuracy*

Definition:

Concept type: Accuracy Level

Evidence Evaluation *has medium low accuracy*

Definition:

Concept type: Accuracy Level

Evidence Evaluation *has low accuracy*

Definition:

Concept type: Accuracy Level

Evidence Evaluation *has unknown accuracy*

Definition:

Concept type: Accuracy Level

Confidence

Definition:

Concept type: Evidence Attribute

Evidence Evaluation *is reported as fact*

Definition:

Concept type: Confidence

Evidence Evaluation *is reported as plausible*

Definition:

Concept type: Confidence

Evidence Evaluation *is reported as uncertain*

Definition:

Concept type: Confidence

Evidence Evaluation *is reported with unknown confidence*

Definition:

Concept type: Confidence

Strength

Definition:

Concept type: Facttype

Evidence Evaluation *has* Strength

Definition:

Concept type: Facttype

Evidence Attribute

Definition:

Concept type: evidence attribute

Reference schema: id **of an** Evidence Element

Evidence Evaluation *has* Evidence Attribute

Definition:

Concept type: Facttype

Evidence Attribute *has* Provenance Property

Definition:

Concept type: Facttype

A.4.6 Evidence Interpretation

Evidence Element *is* **an** Object

Definition:

Concept type: FactType

Evidence Element *means that* Domain Assertion

Definition:

Concept type: FactType

Evidence Element *is characterized by* Domain Assertion

Definition:

Concept type: FactType

Evidence Element *is scoped by* Object

Definition:

Concept type: FactType

Evidence Interpretation

Definition:

Concept type: FactType

A.4.7 Evaluation Context

Evidence Context

Definition:

Concept type: FactType

Evidence Context *includes* Element

Definition:

General concept: Evidence Evaluation

Concept type: FactType

Evidence Context *provides context to* Evidence Element

Definition:

General concept: Evidence Evaluation

Concept type: FactType

Evidence Attribute$_1$ *supercedes* Evidence Attribute$_2$

Definition:

General concept: Evidence Evaluation

Concept type: FactType

A.5 Properties

A.5.1 Provenance Properties

Evidence Element *is created by* Stakeholder

Definition:

General concept: Provenance

Concept type: FactType

Evidence Element *is approved by* Stakeholder

Definition:

General concept: Provenance

Concept type: FactType

Evidence Element *is owned by* Organization

Definition:

General concept: Provenance

Concept type: FactType

Provenance

Definition:

General concept: Evidence Property

Concept type: FactType

A.5.2 Timing Properties

Evidence Element *is reported at* Datetime

Definition:

General concept: Timing

Concept type: FactType

Effective Time

Definition:

General concept: Evidence Property

Concept type: FactType

Evidence Element *is effective starting at* Datetime

Definition:

General concept: Effective Time

Concept type: FactType

Evidence Element *is effective ending at* Datetime

Definition:

General concept: Effective Time

Concept type: FactType

Timing

Definition:

General concept: Evidence Property

Concept type: FactType

A.5.3 Evidence Events

Evidence Item *is acquired at* Location

Definition:

General concept: Evidence Event

Concept type: FactType

Evidence Item *is created at* Location

Definition:

General concept: Evidence Event

Concept type: FactType

Evidence Item *is generated at* Location

Definition:

General concept: Evidence Event

Concept type: FactType

Evidence Item *is transferred to* Location

Definition:

General concept: Evidence Event

Concept type: FactType

Evidence Item *is revoked at* Location

Definition:

General concept: Evidence Event

Concept type: FactType

Evidence Event

Definition:

General concept:Evidence Element

Concept type: Concept

Custody Property

Definition:

General concept:Evidence Property

Concept type: FactType

Evidence Event *is transferred in care of* Person

Definition:

General concept:Evidence Event

Concept type: FactType

Evidence Event *using* Collection Method

Definition:

General concept:Evidence Event

Concept type: FactType

A.5.4 Description

Evidence Item *has* Description

Definition:

Concept type: FactType

Description

Definition: An informal text accompanying an evidence item

Concept type: text

Reference schema:Description of an Evidence Item

A.6 Stakeholders

Stakeholder

Definition:

Concept type: Concept

Reference schema: id of an Evidence Element

Organization

Definition:

Source: based on Merriam-Webster Dictionary [‘Organization’]

Concept type: Concept

Reference schema:id of an Evidence Element

Person

Definition:

Source: based on Merriam-Webster Dictionary [‘Person’]

Concept type: Concept

Reference schema:id of an Evidence Element

Person *is affiliated with* Organization *in* Afficiation

Definition:

Concept type: FactType

Organization *is affiliated with* Organization *in* Afficiation

Definition:

Concept type: FactType

Affiliation

Definition:

Concept type: Concept

A.7 Methods

Collection Method

Definition:

Concept type: Concept

Reference schema: id of an Evidence Element

Method

Definition:

Concept type: Concept

Reference schema: id of an Evidence Element

Tool

Definition:

Concept type: Concept

Reference schema: id of an Evidence Element

Collection Method *derives* Evidence Item *from* Evidence Item

Definition:

Concept type: FactType

Method *requires* Tool

Definition:

Concept type: FactType

A.8 Project

Administrative Element

Definition:

Concept type: Concept

Reference schema: id of an Evidence Element

Administrative Element *is called* Name

Definition:

Concept type: FactType

Reference schema:Name of an Administrative Element

Evidence Package

Definition:

Concept type: Concept

Reference schema: id of an Evidence Element

Evidence Package *contains* Evidence Element

Definition:

Concept type: FactType

Evidence Package *contains* Evidence Request

Definition:

Concept type: FactType

Evidence Package *contains* Tool

Definition:

Concept type: FactType

Evidence Package *contains* Method

Definition:

Concept type: FactType

Evidence Package *contains* Contributor

Definition:

Concept type: FactType

Project Objective

Definition:

Concept type: Concept

Reference schema: id of an Administrative Element

Activity

Definition:

Concept type: Concept

Reference schema: id of an Administrative Element

Evidence Package *contains* Project Objective

Definition:

Concept type: FactType

Evidence Package *contains* Activity

Definition:

Concept type: FactType

Activity *depends on* Activity

Definition:

Concept type: FactType

Stakeholder *is responsible for* Activity

Definition:

Concept type: FactType

Activity *requires* Collection Method

Definition:

Concept type: FactType

Activity *is associated with* Evidence Request

Definition:

Concept type: FactType

Activity *satisfies* Project Objective

Definition:

Concept type: FactType

Rationale

Definition: Informal text that explains evidence resolution

Concept type: Concept

Annex B - Examples

(non-normative)

The section provides two examples of argument from the safety and the security domain. The safety argument refers to an industrial press, whereas the security example is a fragment from a Bluetooth security case.

B.1 Industrial Press Safety Argument

<?xml version="1.0" encoding="ASCII"?>

<ARM:Argumentation xmi:version="2.1"

xmlns:xmi="http://schema.omg.org/spec/XMI/2.1"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:ARM=" www.omg.org/spec/SACM/20120501/Argumentation"

xmi:id="0" id="IPSA">

<xsd:import namespace=http://schema.omg.org/spec/XMI/2.1" schemaLocation="http://www.omg.org/spec/XMI/20071213/XMI.xsd"/>

<xsd:import namespace="www.omg.org/spec/SACM/20120501/Argumentation" schemaLocation=" http://www.omg.org/spec/SACM/20120501/Argumentation.xsd" />

<argumentElement xsi:type="ARM:Claim" xmi:id="1" id="C1" description="" content="C/S logic is fault free"/>

<argumentElement xsi:type="ARM:ArgumentReasoning" xmi:id="2" id="RC1.1" content="Argument by omission of all identified software hazards" describes="5 6"/>

<argumentElement xsi:type="ARM:ArgumentReasoning" xmi:id="3" id="RC1.2" content="Argument by satisfaction of all C/S safety requirements" describes="7 8 9"/>

<argumentElement xsi:type="ARM:InformationElement" xmi:id="4" id="IRC1.1" description="Identified software hazards"/>

<argumentElement xsi:type="ARM:Claim" xmi:id="5" id="C1.1" description="" content="Unintended opening of press (after PoNR) can only occur as a result of component failure"/>

<argumentElement xsi:type="ARM:Claim" xmi:id="6" id="C1.2" description="" content="Unintended closing of press can only occur as a result of component failure"/>

<argumentElement xsi:type="ARM:Claim" xmi:id="7" id="C2.1" content="Press controls being 'jammed on' will cause press to halt"/>

<argumentElement xsi:type="ARM:Claim" xmi:id="8" id="C2.2" content="Release of controls prior to press passing physical PoNR will cause press operation to abort"/>

<argumentElement xsi:type="ARM:Claim" xmi:id="9" id="C2.3" description="" content="C/S fails safe (halts on) and annunciates (by sounding Klaxon) all component failures" toBeSupported="true"/>

<argumentElement xsi:type="ARM:Claim" xmi:id="12" id="C2.1.1" content="Failure 1 of PLC state machine includes BUTTON_IN remaining true"/>

<argumentElement xsi:type="ARM:Claim" xmi:id="13" id="C2.2.1" content="Abort transition of PLC state machine includes BUTTON_IN going false"/>

<argumentElement xsi:type="ARM:InformationElement" xmi:id="10" id="S1.1" content="Fault tree analysis cutsets for event 'Hand trapped in press due to command error'"/>

<argumentElement xsi:type="ARM:InformationElement" xmi:id="11" id="S1.2" content="Hazard directed test results"/>

<argumentElement xsi:type="ARM:InformationElement" xmi:id="14" id="S2.1" description="" content="black box testing"/>

<argumentElement xsi:type="ARM:InformationElement" xmi:id="15" id="S2.2.1" content="C/S state machine"/>

<argumentElement xsi:type="ARM:AssertedInference" xmi:id="16" id="C1.1.1" description="" source="5" target="1"/>

<argumentElement xsi:type="ARM:AssertedInference" xmi:id="17" id="C1.1.2" source="6" target="1"/>

<argumentElement xsi:type="ARM:AssertedInference" xmi:id="18" id="C1.2.1" source="7" target="1"/>

<argumentElement xsi:type="ARM:AssertedInference" xmi:id="19" id="C1.2.2" source="8" target="1"/>

<argumentElement xsi:type="ARM:AssertedInference" xmi:id="20" id="C1.2.3" source="9" target="1"/>

<argumentElement xsi:type="ARM:AssertedContext" xmi:id="21" id="CIRC1.1" source="4" target="2"/>

<argumentElement xsi:type="ARM:AssertedEvidence" xmi:id="22" id="S1.1" source="10" target="5 6"/>

<argumentElement xsi:type="ARM:AssertedEvidence" xmi:id="23" id="S1.2" source="11" target="5 6"/>

<argumentElement xsi:type="ARM:AssertedEvidence" xmi:id="24" id="SC2.1" source="14" target="7"/>

<argumentElement xsi:type="ARM:AssertedEvidence" xmi:id="25" id="SC2.1.1" source="15" target="12"/>

<argumentElement xsi:type="ARM:AssertedEvidence" xmi:id="26" id="SC2.2.1" source="15" target="13"/>

<argumentElement xsi:type="ARM:AssertedInference" xmi:id="27" id="DI C2.1" source="12" target="7"/>

<argumentElement xsi:type="ARM:AssertedInference" xmi:id="28" id="DI C2.2" source="13" target="8"/>

<argumentElement xsi:type="ARM:AssertedContext" xmi:id="29" id="AR29" source="2" target="16 17"/>

</ARM:Argumentation>

B.2 Bluetooth Security Case

<?xml version="1.0" encoding="ASCII"?>

<ARM:Argumentation xmi:version="2.1"

xmlns:xmi="http://schema.omg.org/spec/XMI/2.1"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:ARM=" www.omg.org/spec/SACM/20120501/Argumentation"

xmi:id="0" id="BSC11">

<xsd:import namespace=http://schema.omg.org/spec/XMI/2.1" schemaLocation="http://www.omg.org/spec/XMI/20071213/XMI.xsd"/>

<xsd:import namespace="www.omg.org/spec/SACM/20120501/Argumentation" schemaLocation=" http://www.omg.org/spec/SACM/20120501/Argumentation.xsd" />

<argumentElement xsi:type="ARM:Claim" xmi:id="1" id="Bluetooth secure" content="A bluetooth enabled network provides adequate security"/>

<argumentElement xsi:type="ARM: Claim" xmi:id="2" id="Availability" content="A bluetooth enabled network is adequately available [1] Section 1 para 3"/>

<argumentElement xsi:type="ARM: Claim" xmi:id="3" id="Access" description="" content="A bluetooth enabled network provides adequate control for access to services and data [1] Section 1 para 3"/>

<argumentElement xsi:type="ARM: Claim" xmi:id="4" id="Confidentiality" content="A bluetooth enabled network provides adequate levels of confidentiality [1] Setion 1 para 3"/>

<argumentElement xsi:type="ARM: Claim" xmi:id="5" id="Integrity" content="A bluetooth enabled network provides adequate levels of integrity [1] Section 1 para 3"/>

<argumentElement xsi:type="ARM: InformationElement" xmi:id="6" id="Context: security policy and scenario for use" content="Definitions are required of the intented security policy and the scenario of use for the system, including what is regarded as 'adequate'"/>

<argumentElement xsi:type="ARM: InformationElement" xmi:id="7" id="References" content="[1] Bluetooth security white paper 19/4/ 02"/>

<argumentElement xsi:type="ARM: InformationElement" xmi:id="8" id="Definition: Availability" content="The system is capable of providing requested services to authorised users, in an acceptable/defined time"/>

<argumentElement xsi:type="ARM: InformationElement" xmi:id="9" id="Definition: Access" content="Only users permitted by the defined security policy have access to services and data"/>

<argumentElement xsi:type="ARM: InformationElement" xmi:id="10" id="Define: Confidentiality" content="Unauthorised persons cannot intercept and understand information to which they are not entitled"/>

<argumentElement xsi:type="ARM: InformationElement" xmi:id="11" id="Define: Integrity" description="" content="Services and data are provided to authorised users as intended and without corruption"/>

<argumentElement xsi:type="ARM: AssertedContext" xmi:id="12" id="AC1" source="7" target="1"/>

<argumentElement xsi:type="ARM: AssertedContext" xmi:id="13" id="AC2" source="6" target="1"/>

<argumentElement xsi:type="ARM: AssertedContext" xmi:id="14" id="AC3" source="8" target="2"/>

<argumentElement xsi:type="ARM: AssertedContext" xmi:id="15" id="AC4" source="9 " target="3"/>

<argumentElement xsi:type="ARM: AssertedContext" xmi:id="16" id="AC5" source="10" target="4"/>

<argumentElement xsi:type="ARM: AssertedContext" xmi:id="17" id="AC6" source="11" target="5"/>

<argumentElement xsi:type="ARM: AssertedInference" xmi:id="18" id="AI1" source="5 4 3 2" target="1"/>

<argumentElement xsi:type="ARM: ArgumentReasoning" xmi"id="19" id="Argue over vulnerabilities" description="" content="Argue for each security requirement identified in the security white paper" describes="18"/>

</ARM:Argument>

B.2.1 Goal Structuring Notation (GSN) Examples

This section contains examples of arguments using the Goal Structuring Notation. The following table explains the relationship from the example to the modeling elements of SACM Argumentation Metamodel.

GSN element	SACM Argumentation Metamodel counterpart
Rectangle	Claim
Rounded rectangle	InformationElement
Parallelogram	ArgumentReasoning
Circle	InformationElement linked using an AssertedEvidence instance
Filled arrow	AssertedInference (or AssertedEvidence when linked to circle). The arrow head attaches to the source element.

Empty arrow	AssertedContext. The arrow head attaches to the source element.
Diamond decorator	ToBeSupported = true
Shaded triangle decorator	The current element is a citation element.

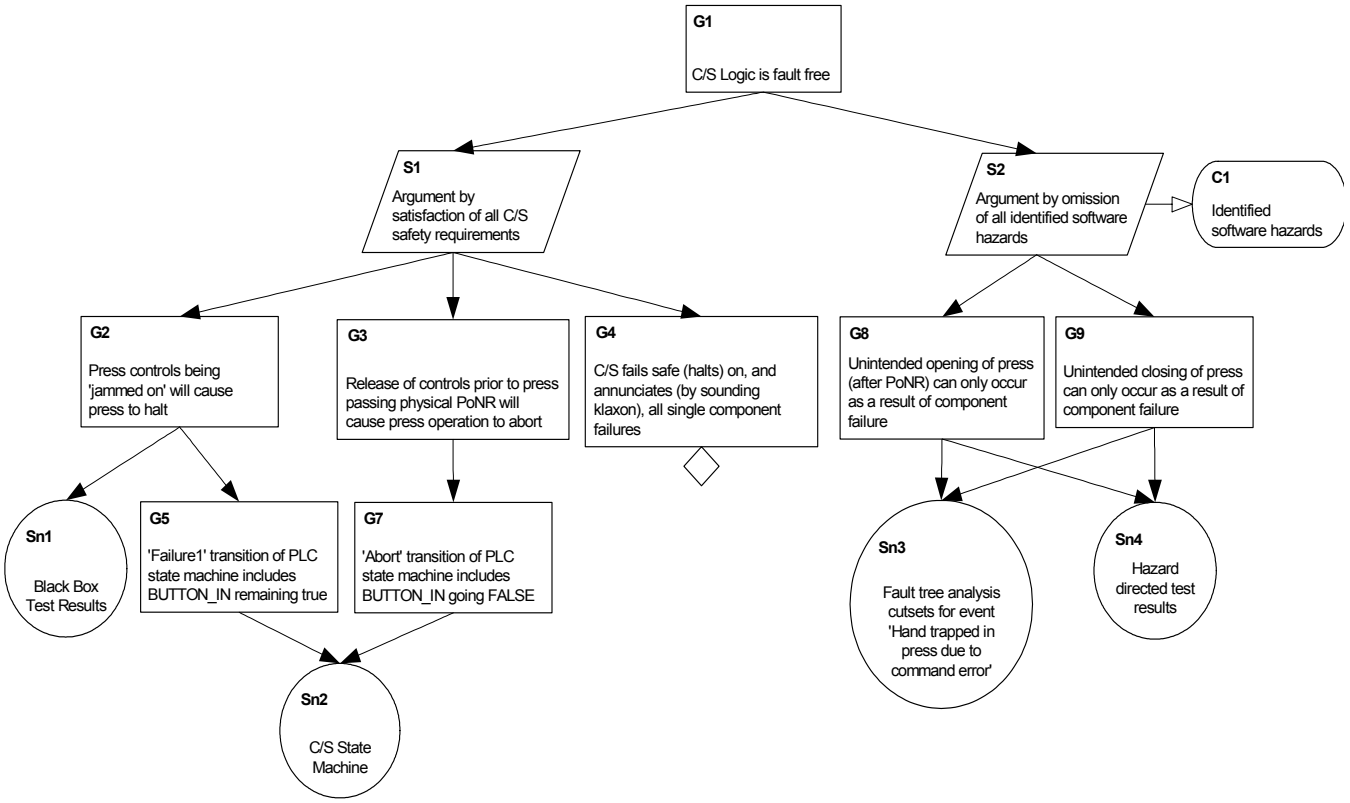

Figure B.1 - Industrial Press Safety argument (§8.3.1)

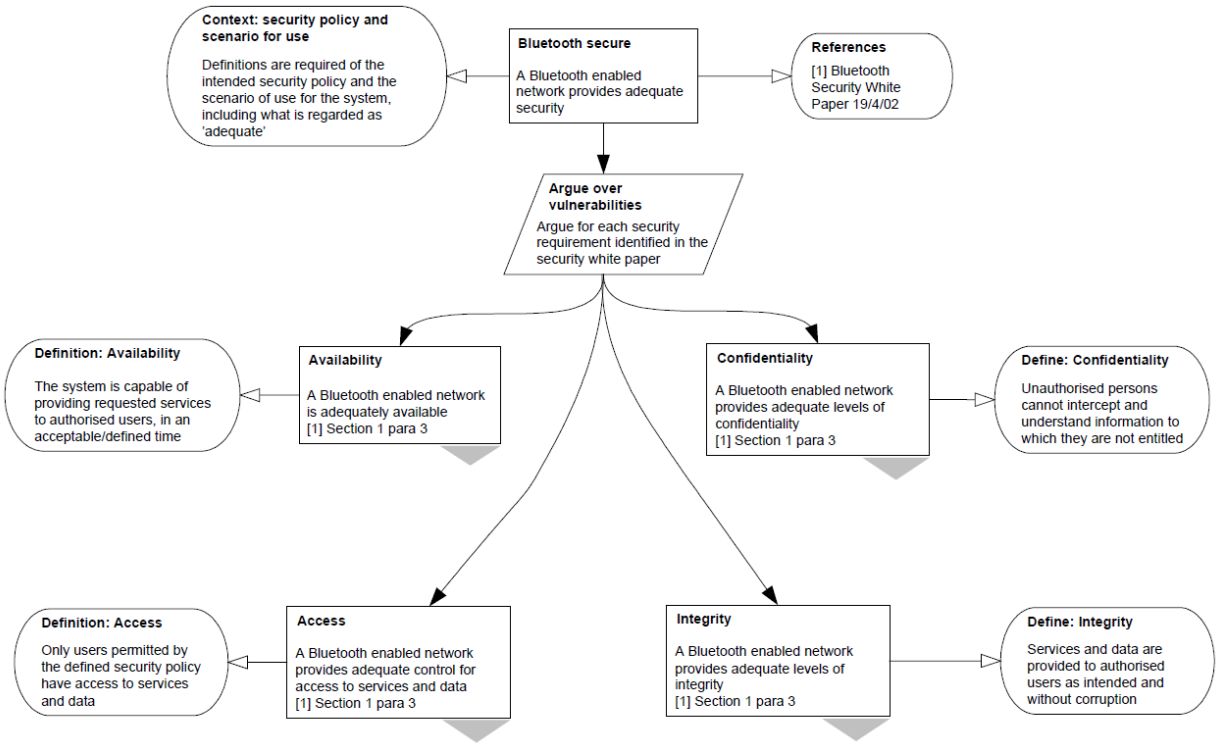

Figure B.2 - GSN Bluetooth Security Case (§8.3.2)

B.2.2 Claims-Arguments-Evidence (CAE) Example

In CAE, contextual information can be represented either as visual nodes in a similar manner to GSN (see Figure B.3), or alternatively as rich text associated with the node (see Figure B.4).

The following table explains the relationship from the example to the modeling elements of the SACM Argumentation Metamodel.

CAE element	SACM Argumentation Metamodel counterpart
Blue elipse	Claim
Green rounded box	ArgumentReasoning
Element with no border	InformationElement
Blue arrow	AssertedInference
Green arrow	AssertedInference (unless from InformationElement, in which case AssertedContext)
Rich narrative text	InformationElement attached using AssertedContext to the current element.

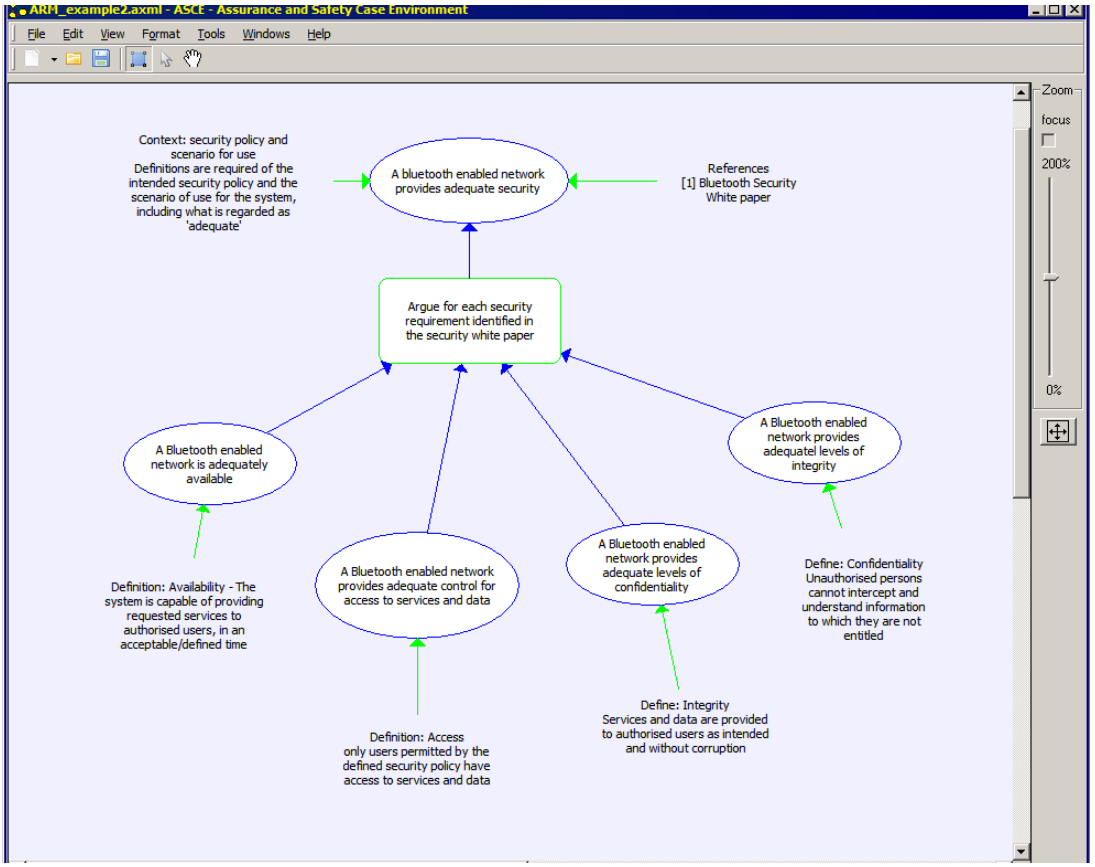

Figure B.3 - CAE of Bluetooth example - showing contextual information as visual nodes

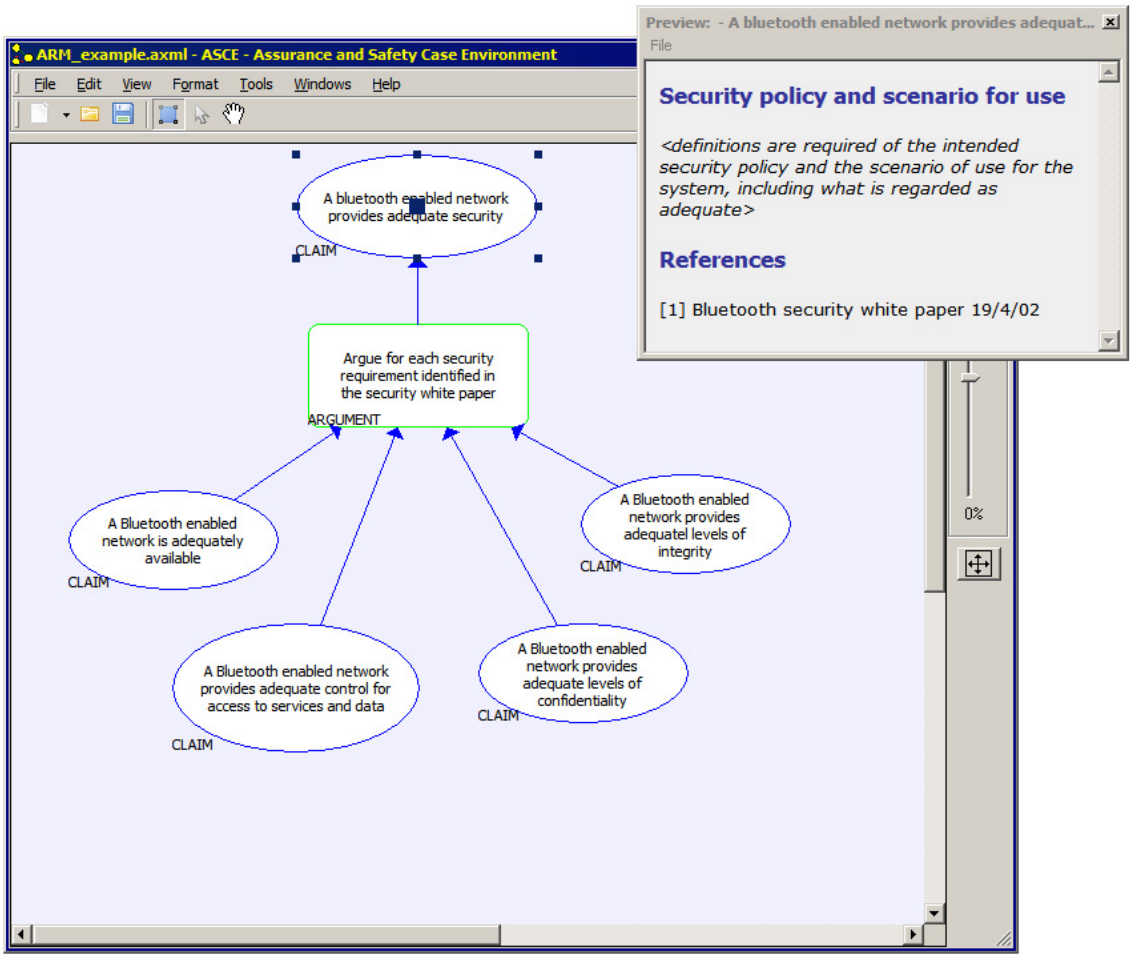

Figure B.4 - CAE representation of the Bluetooth example where contextual information held as rich text (top claim is selected)