Date: April 2012

Structured Assurance Case Metamodel (SACM)

FTF - Convenience Document 1

OMG Document Number: ptc/2012-04-04
Standard document URL: http://www.omg.org/spec/SACM
Associated Schema Files:

sysa/2010-03-16 -- http://www.omg.org/spec/ARM/20100301
sysa/2010-03-17 -- http://www.omg.org/spec/ARM/20100302
sysa/2010-02-02 -- http://www.omg.org/spec/SAEM/20100201
sysa/2010-02-03 -- http://www.omg.org/spec/SAEM/20100202
sysa/2010-02-04 -- http://www.omg.org/spec/SAEM/20100203
sysa/2010-02-05 -- http://www.omg.org/spec/SAEM/20100204

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

The SACM Evidence Metamodel is the first specification in the series of System Assurance specifications that will enable creation of new type of Software Assurance tools related to assurance of safety and security of software-intensive systems, and bring automation to the processes of regulatory compliance and risk assessments.

The SACM Evidence Metamodel establishes the necessary fine grained models of evidence elements required for detailed compliance and risk analysis.

The structure of the Evidence Metamodel provides the basis for logical design of easily-constructed tools for storing, cross-referencing, evaluating and reporting the elements of evidence for systems during the Software Assurance.

1.3 Why Software Assurance Evidence?

While organizations are becoming increasingly dependent on software, targeted attacks against software are on the rise, causing harm to the infrastructure and disrupting business operations. There is a growing consensus in the industry and government that the software industry needs to actively address the root causes of exploitable vulnerabilities and implement methods to improve software resilience to attacks from the onset, thereby enhancing **software trustworthiness**. Unfortunately, it is becoming increasingly difficult to establish or verify whether or not software is sufficiently trustworthy, due to a variety of factors such as: scope, volume and complexity of software systems, software interconnectivity, networks and net-centricity, rate of change of technology, globalization, use of open source & COTS, security issues in legacy code, etc.

Level of confidence that given software is trustworthy is called Software Assurance (SwA) and process that establishes level of confidence is called SwA process.

Currently, SwA process is mostly informal, subjective and manual. As complications to assessing trustworthiness will continue to evolve, formalized and standardized mechanisms/approaches must be developed that increase software assurance. However, to make software assurance 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 that allows:

- Different participants to initiate collaboration and activities in areas of SwA through common assurance standards,
- Enabling of a new generation of supporting solutions that benefit all participants, and
- Enhancement/improvement in automation of SwA activities by enabling interoperability between different supporting solutions (toolsets).

Over the past few years, a strong cross-section of SwA participants (software intensive organizations, users, consumers, and regulators) has emerged with the objective of promoting software assurance within the community. Until recently, these participants have been working mostly in isolation. Standardization of SwA concepts and methodologies builds upon a combination of prior experiences, domain knowledge, and best practices, and ultimately facilitates interoperability for the creation, exchange, and use of assurance-related information among community participants.

The result of SwA is a clear, comprehensive and defensible argument that a system is acceptably safe and secure to operate within a particular context. Several industries have already recognized the value of a structured argument as a means of assurance of other dependability properties such as safety and reliability and regulatory controls have been instantiated in some countries that require safety of critical system be demonstrated through the development of assurance cases. An assurance case presents the safety argument in which the claims are supported by evidence.

Certain claims are supported through evidence, i.e., relies on external documented facts to confer evidentiary support to claims. Evidence is collected by applying systematic methods and procedures In the software assurance context, evidence is often collected by 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, we must weigh the source we are introducing.
- 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

Software that conforms to the specification 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 specification.

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
- XMI 2.1 Specification
- SBVR
- ISO 15026

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.

Structured argument

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

Evidence

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

Claim

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

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

Term	Definition
Evidence	A document or other exhibit that provides justification to a certain claim.
Assertion	A proposition that is suggested to be taken as true, a claim.
Fact	A proposition that is agreed to be taken as true.
Argument	Collection of claims and their justification.
Assurance	Clear and convincing communication of the safety and security posture of a system.
Evidence repository	A software service providing access to, and information about a collection of evidence items, such as documents and other exhibits together with their attributes.

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 attribute	A characteristic of an evidence item.
Safeguard	Actions or measures taken to offset a particular safety or security concern or threat.
Software Assurance	The level of confidence that safety and security safeguards of the software function as intended and that the software is free of vulnerabilities, either intentionally or unintentionally designed or inserted as part of the software.
Software Assurance Process	The process of establishing level of confidence in safety and security safeguards of the software and communicating the safety and security posture of the software.
Software Assessment	The process of obtaining facts about the adequacy of the safety and security safeguards of the software.
Assurance Case	A set of auditable claims, arguments and evidence created to support the contention that a defined system/service will satisfy the particular requirements through supporting arguments and evidence.
Software Artifact	Source code, executable, software module, software architecture.

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.

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

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, the 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.

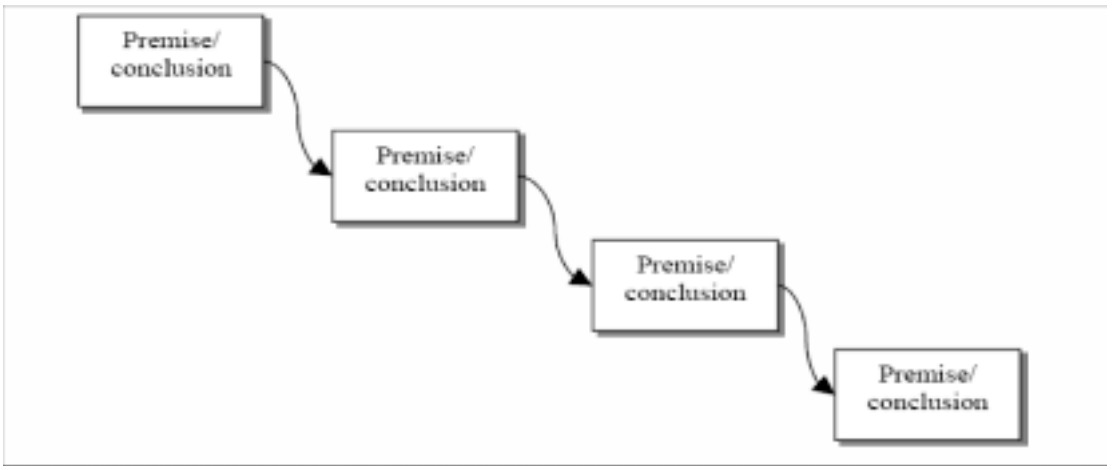


Figure 7.1 - Argument Chain Structure

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 approach that allows 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 Evidence Metamodel

In the simplest form, evidence consists of a collection of documents 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. We will differentiate subject claims from evidence claims, which are claims about the evidence items that help establish the exact nature of the evidentiary support they provide to subject claims in a crea, comprehensive and defensible way. Evidence arguments are reused as opposed to subject domain claims and arguments, which are specific to each subject domain. The evidence vocabulary describes claims made about evidence. Evidence vocabulary is reused in every argument for various diverse domains.

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 Exchange 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 evidence collection projects, including

- Collection of evidence
- Management of evidence
- Interpreting evidence
- Evaluation of evidence

Collection of Evidence includes activities of identifying evidence items, and recording various information about them, including their origin, timing, custody. In Software Assurance, evidence items are often generated by tools. Evidence Metamodel provides means to represent the full pedigree of an evidence item, including evidence collection method used.

In the Evidence Metamodel Management of Evidence is addressed by providing several model elements that represent Evidence Items. The primary items of the Evidence Metamodel are Documents, Assertions and Objects. A Document is an Object, however this is a primary Evidence Item, because by design the Evidence Metamodel is Document-oriented, so Document is explicitly represented in the Evidence Metamodel together with its several essential attributes. In general, it is possible to represent a Document as an Object, and represent its attributes as specialized Assertions (for example, same

as Quality attributes). On the other hand, an Object does not have any attributes, only identity. Object in Evidence Metamodel is same as SBVR “Thing”. Usually Objects correspond to real things (an SBVR “Res”), however in theory one also can make assertions about Concepts not than real things. Objects in Evidence Metamodel use external vocabulary to refer to their types. It is assumed that the type of object is specified by a reference to an SBVR vocabulary or an OWL ontology, however less formal evidence packages may describe types of objects informally as prose. Documents may have Properties, such as one Document may be part of another Document.

Properties in the Evidence Metamodel include the following:

- Provenance, stating the Provenance Attributes of Evidence Elements
  - Who created
  - Who approved
  - Who owns
- Custody, stating the Custody Attributes of Evidence Items
  - Where
- Timing, stating the Timing Attributes of Evidence Elements
  - When
  - Effective Time

Assertions in the Evidence Metamodel represent candidate facts (i.e., assertions are meanings that are suggested to be taken as true). The goal of collecting documents as evidence is to provide justification for the claims. The nature of argument, of which the evidence is essential part, involves dialog during which the validity of certain claims is established. Not all claims are justified through evidentiary support though. Some claims are justified through reasoning. Real things like documents and, in general, objects, do not have truth value, they simply exist. Assertions correspond to propositions involving objects (and documents), therefore assertions have truth value. Evidence packages list Documents and Assertions and describe the evidentiary support the Documents provide to Assertions. This relation can be that of evidentiary support or evidentiary challenge. Usually the goal of the evidence package is to provide evidentiary support to certain assertions. However addressing counterevidence in the course of evaluating evidentiary findings is important too and to that end it is important to represent assertions that are in certain sense complimentary to the intended claims (assertion that we want to justify as being true facts). Not all complimentary assertions, corresponding to counterevidence have to false though.

Assertions in an evidence package are suggested to be taken as true (or false), as they tend to be rather primitive and are assumed to represent mutually accepted facts. However in general, it is possible to contest the validity of the evidence assertions.

Assertions in the Evidence Metamodel are of two major categories: Evidence Assertions and Domain Assertions. Domain Assertions are claims about the domain of the argument. Domain Assertions involve Domain Objects. Domain Assertions and Objects refer to external vocabularies to determine their exact type. It is assumed that the type of assertion is specified by a reference to an SBVR vocabulary or an OWL ontology, however less formal evidence packages may describe assertions informally as prose. Evidence Assertions are made in the course of evaluating evidence. Evidence Assertions are defined within the Evidence Metamodel itself.

Management of Evidence compliments evidence collection activities with associated planning and tracking activities. Important to management of evidence is the set of Administrative Elements, including a Package, 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 provenance and timing properties, these administrative elements are powerful enough to support management of evidence collection projects and exchange of the managerial data as part of evidence packages.

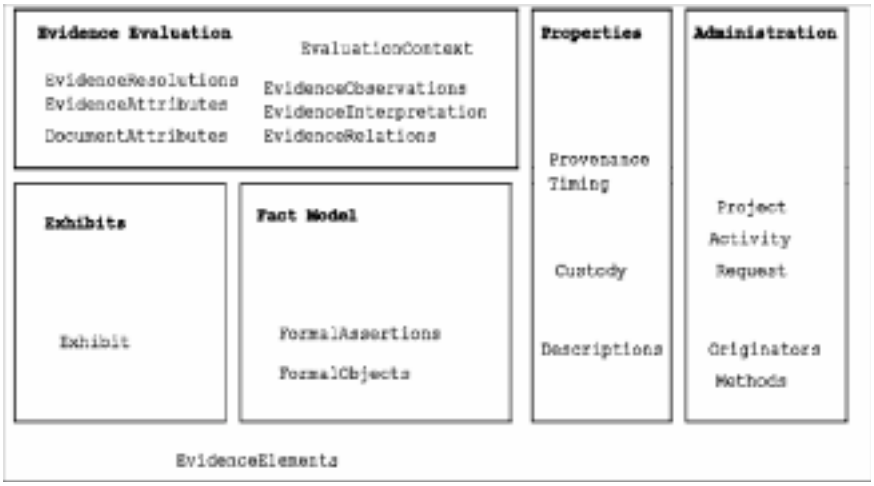
Evaluation of Evidence includes the activities of making certain assertions about evidence items and their relation to domain claims.

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

- Quality Attributes of Documents, such as
  - Primary or secondary
  - Document: original or derived
  - Consistency
  - Completeness
  - Accuracy
- Quality Attributes of Evidentiary Support
  - Direct or indirect
  - Relevance
  - Confidence
  - Strength
  - Significance
- Nature of 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)
    - Genealogical Proof Standard (GPS)
    - Beyond the reasonable doubt (BRD)

7.5.1 Example Evidence Evaluation Process (non normative)

The following diagram is related to the so-called Genealogical Proof Standard, which illustrates the recommended steps for evaluating evidence.



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 material object is entered 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

In addition to evidence items and evidence assertions, the Evidence Metamodel supports Evidence Events, which are statements recording the chain of custody of the evidence items. Evidence Events are not assertions, such as domain assertion and domain claims or evidence assertions that can be challenged. Instead, they are records concerning the evidence items.

7.6 The Key Elements of Evidence

Relationships between the key elements of the evidence metamodel are illustrated in Figure 6.1.

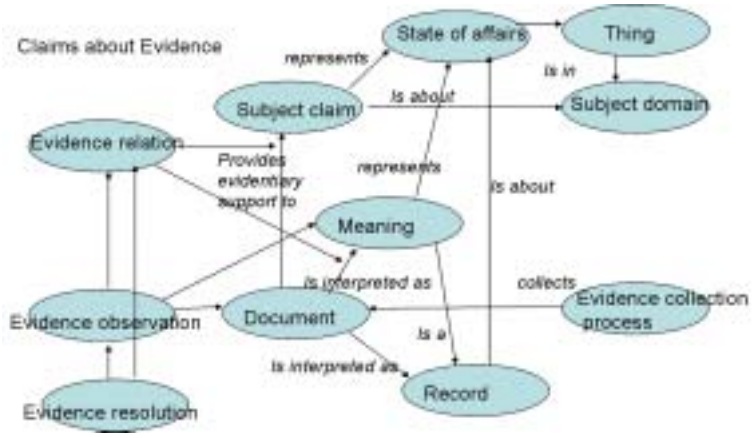


Figure 6.1 - The key elements of the SACM Evidence Metamodel (non-normative)

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. Document is interpreted as a Meaning that is a state of affairs involving several Things in the Subject domain (for which Subject claims are being made). Subject claims are about Subject domain, that is to say that a Subject claim represents a state of affairs between Things that are in the Subject domain. Often, a Document is interpreted as a Record of the state of affairs involving things in the Subject domain. 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 quality 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 shall have a timestamp and provenance, so the entire history of the evidence collection and evaluation activities can be generated. Figure 6.2 summarizes the life cycle of an Evidence Item (A Document or an Object).

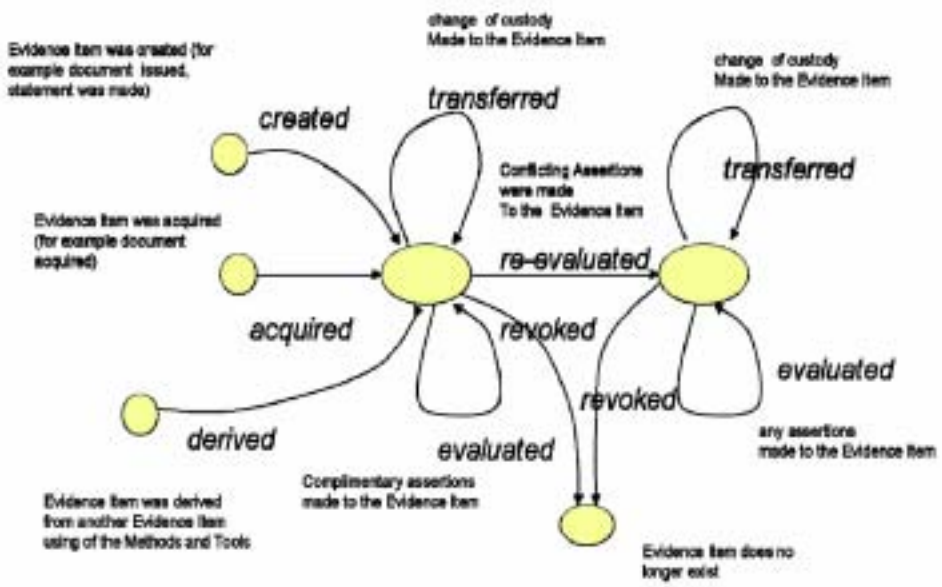


Figure 6.2 - The life-cycle of an evidence item (non-normative)

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.

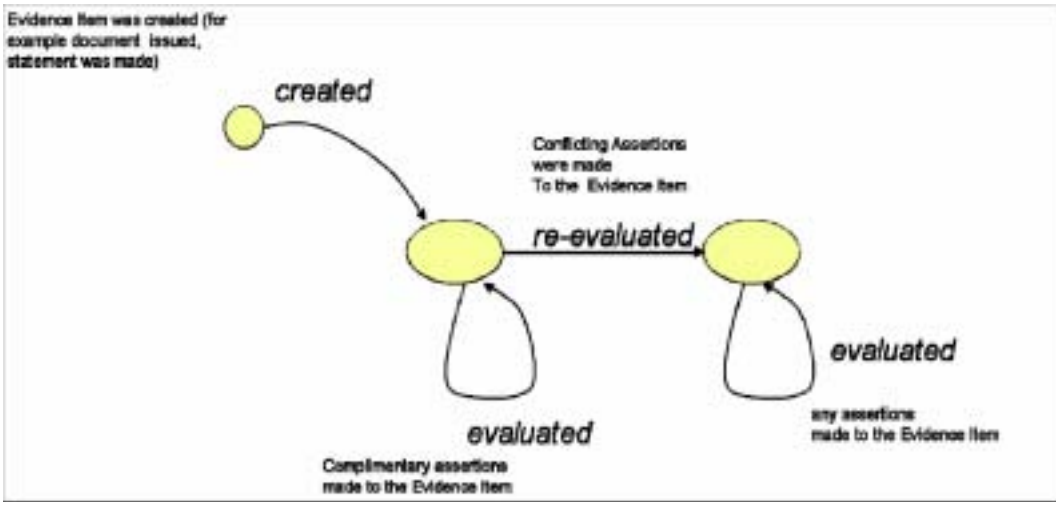


Figure 6.3 - Life-cycle of an Assertion or Evaluation (non-normative)

Assertions and Evaluations have simpler life cycle, where they are created and evaluated or re-evaluated, see Figure 6.3. Assertions and Evaluations can not be acquired, derived, transferred or revoked, since they are statements involving objects.

Attributes are objects bound to roles.

	Document, Exhibit	DomainObject, Domain Assertion	Evaluation
IsCreatedAt	At location By source (person) Approved by supervisor At time Effective time Owned by organization	By source Approved by supervisor At time Effective time Owned by organization	By source Approved by supervisor At time Owned by organization
IsAcquiredAt	At location By source (person) At time Owned by organization	N/A	N/A
IsGeneratedAt	At location By source (person) Approved by supervisor At time Owned by organization	N/A	N/A
Evidence Evaluation (*Supports, Challenges, Weakens, Amplifies, Conflicts, Refutes, Negates, Resolves as well as Document and Evidence attributes*)	By source Approved by supervisor At time Owned by organization	By source Approved by supervisor At time Owned by organization	N/A
IsTransferredTo	At location To curator By source At time Approved by supervisor Owned by organization	N/A	N/A
IsRevokedAt	By source Approved by supervisor At time Owned by organization (?)	N/A	N/A

7.8 Evaluation of Evidence

The following table summarizes the EvidenceEvaluation relationships that can be added during the process of evaluating evidence.

EvidenceEvaluation	Source	Target
Supports	Exhibit, domain assertion	Domain assertion
Contradicts	Exhibit, domain assertion	Domain assertion
weakens	Evidence relation (supports, contradicts)	Evidence relation (supports, contradicts)
amplifies	Evidence relation (supports, contradicts)	Evidence relation (supports, contradicts)
conflicts	Domain assertion	Domain assertion
Refutes	EvidenceContext, Rationale	Domain assertion
Negates	EvidenceContext, Rationale	Evidence relation (supports, contradicts)
Resolves	EvidenceContext, Rationale	Observation (weakens, amplifies)

7.9 Design Characteristics of The Evidence Metamodel

The following are key design characteristics of the SACM Evidence Metamodel:

- The Evidence Metamodel is a MOF model.
- The Evidence Metamodel is aligned with RDF: Design of the Evidence Metamodel facilitates transformation of the Evidence Metamodel documents as RDF triples in a more straightforward way than arbitrary MOF models. In particular, the Evidence Metamodel elements can be represented as RDF resources (Exhibit, Description, AdministrativeElement, Object and Assertion), and EvidenceEvent, EvidenceEvaluation, Properties, Attrbiutes and RoleBinding - as RDF triples.
- The Evidence Metamodel can be extended to capture additional information.
- The Evidence Metamodel defines multiple hierarchies of entities via containers.

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.

Part 2 Argumentation Metamodel

This part of the specification defines the Argumentation Metamodel.

9 SACM Argumentation Metamodel

This section 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 Overview

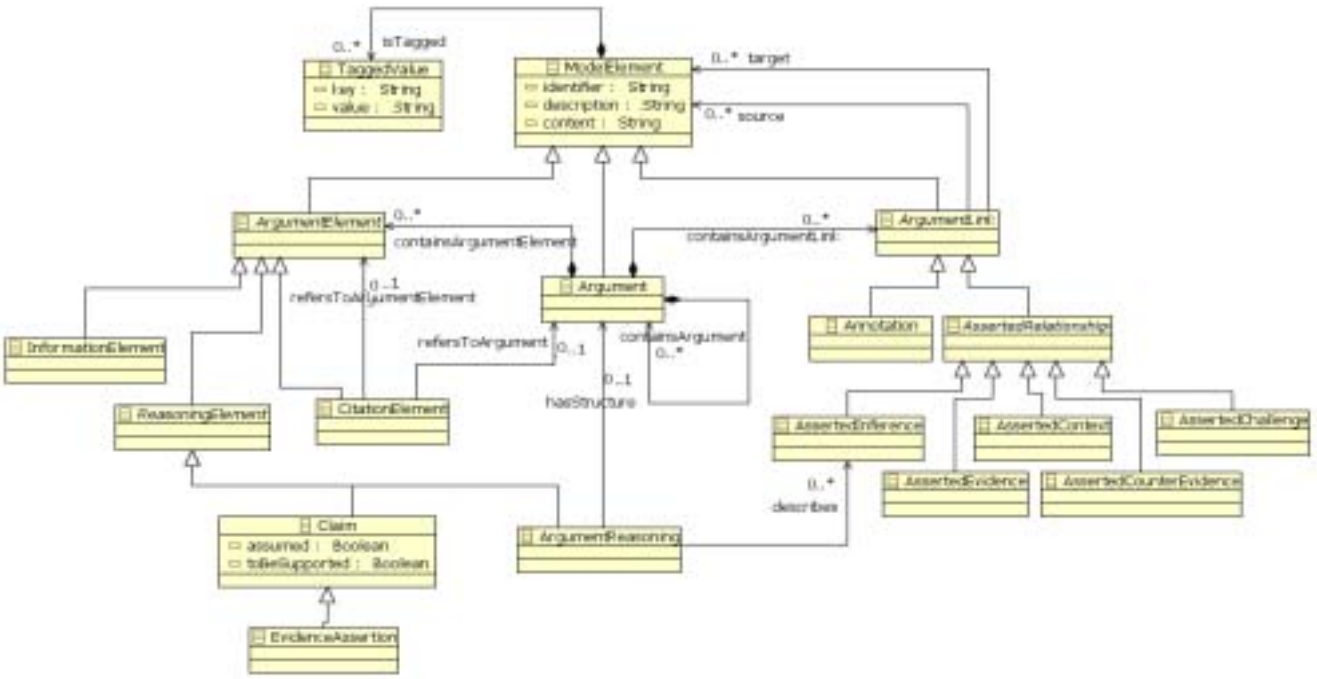


Figure 9.1 - Metamodel overview

9.2 Class Definitions

In the following sections we describe the model elements.

9.2.1 ModelElement Class (Abstract)

A ModelElement is an atomic constituent of a structured argument represented using the Argumentation Metamodel. In the meta-model, ModelElement is the top meta-element in the SACM Argumentation Metamodel class hierarchy. ModelElement is an abstract meta-model element.

Attributes

- identifier: String
  A unique identifier for the Argumentation Metamodel entity.

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

Associations

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

Semantics

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

Invariants

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

9.2.2 TaggedValue Class

A TaggedValue is an annotation that can be provided on any ModelElement in the Argumentation 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 Argumentation Metamodel instance.

9.2.3 Argument Class

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

Superclass

ModelElement

Associations

- containsArgumentElement:ArgumentElement[0..*]
  The ArgumentElements contained in a given instance of an Argument.
- containsArgumentLink:ArgumentLink[0..*]
  The ArgumentLinks (between ArgumentElements) contained in a given instance of an Argument.

Semantics

Structured arguments represented using The Argumentation Metamodel are composed of ArgumentElements and ArgumentLinks between ArgumentElements.

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>
```

9.2.4 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.

9.2.5 ArgumentLink Class (Abstract)

The ArgumentLink Class is the abstract association class that enables the ArgumentElements of any structured argument to be linked together.

Superclass

ModelElement

Associations

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

Semantics

In the Argumentation Metamodel, the structure of an argument is declared through the linking together of primitive ArgumentElements. For example, links between ArgumentElements allow the declaration of the structure of the inferences of the argument, and the association between claims and evidence. In addition argument links can be used to associate ModelElements to ArgumentLinks (e.g., it could be necessary to add a supporting claim – backing – to an AssertedInference between two claims).

9.2.6 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 – namely, Claims and ArgumentReasoning (the description of inferential reasoning that exists between Claims.

Superclass

ArgumentElement

Attributes

- toBeSupported: Boolean
  An attribute recording whether further reasoning has yet to be provided to support the ReasoningElement (e.g., further evidence to be cited).

Semantics

The core of any argument is the reasoning that exists to connect individual claims of that argument. Reasoning is captured in the Argumentation Metamodel through the linking of fundamental claims, and the description of the links between claims. ReasoningElements represent these two elements.

When building an argument it can often be useful to create a ReasoningElement yet not have fully defined the element. When building an argument it is may be useful to denote that further (supporting) reasoning or evidence is necessary to support a ReasoningElement. This is done using the toBeSupported attribute.

9.2.7 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

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.

Example

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

9.2.8 CitationElement Class

The CitationElement Class cites an Argument, or an ArgumentElement within another Argument, for use within an Argument.

Superclass

ArgumentElement

Associations

- refersToArgumentElement:ArgumentElement[0..1]
  References an ArgumentElement within another Argument.
- refersToArgument:Argument[0..1]
  References an Argument.

Semantics

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

9.2.9 Claim Class

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

Superclass

ReasoningElement

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.

Semantics

Structured arguments are declared by stating claims, and asserting how those claims relate to each other. 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 reasoning (in the recorded Argument) 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.

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"/>

9.2.10 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"/>

9.2.11 ArgumentReasoning Class

ArgumentReasoning can be used to provide additional description or explanation of the asserted inference that connect one or more Claims (premises) to another Claim (conclusion). ArgumentReasoning elements are therefore related to AssertedInferences. 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

- describes:AssertedInference[0..*]
  Reference to the AssertedInference being described by the ArgumentReasoning.
- hasStructure: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"/>

9.2.12 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

ArgumentLink

Semantics

In the SACM Argumentation Metamodel, the structure of an argument is declared through the linking together of primitive ArgumentElements (and potentially ArgumentLinks). 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

9.2.13 AssertedInference Class

The AssertedInference association class records the inference that a user declares to exist between one or more Claims (premises) and another Claim (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 primitive Claims. For example, a AssertedInference can be said to exist between two claims (“Claim A implies Claim B”). A 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))
```

9.2.14 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 this Claim and the evidence can be asserted by a 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))

9.2.15 AssertedChallenge Class

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

Superclass

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))

9.2.16 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))

9.2.17 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))

9.2.18 Annotates Class

The Annotates association class declares that the information cited by an InformationElement provides annotation of a ModelElement.

Superclass

ArgumentLink

Semantics

The Annotates association class allows the (informal) association of InformationElements to *any* ModelElement of the structured argument (i.e., both elements and links). For example, it can be useful to attach citations of information such as review comments, descriptive documents, and the text of assurance standards. Importantly: use of the Annotates association class does not assert any *logical* relationship between the InformationElement and the ModelElement (unlike AssertedRelationship and its subclasses), it is merely adding additional ancillary to the structured argument.

Part 3 Evidence Metamodel

This part of the specification defines the Evidence Metamodel.

SACM Evidence Metamodel consists of 20 class diagrams. The logical organization of the SACM Evidence Metamodel is illustrated at Figure 10.1. This figure illustrates the main logical parts of the SACM Evidence Metamodel and shows all 20 class diagrams. SACM Evidence Metamodel is delivered as a single UML package.

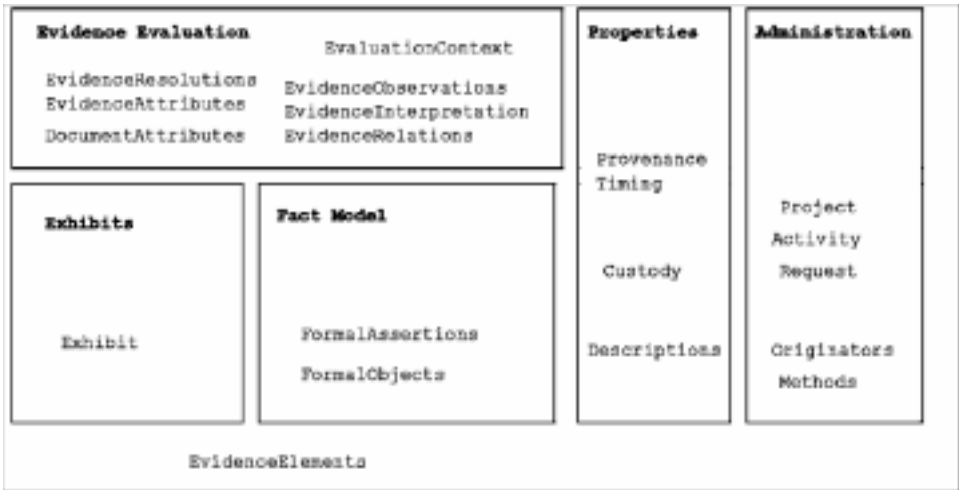


Figure 10.1 - Logical Organization of the SACM Evidence Metamodel

The SACM Evidence Metamodel consists of the following logical parts:

- Exhibits
- Fact Model
- Properties
- Evidence Evaluation
- Administration

The **Exhibits** part defines the coarse-grained evidence, provided in the form of documents and sometimes other exhibits. The Exhibits part also defines the properties of exhibits and documents. The **Fact Model** part defines the fine-grained assertions, provided in the form of individual propositions. These propositions use an external vocabulary of the domain for which an argument is being provided. The Fact Model part defines a subset of an SBVR fact model in the form of atomic formulations based on fact types with role bindings to individual concepts. SBVR is not used directly because of subtle semantic differences between fact models in linguistic models (SBVR), conceptual models and “candidate fact models” involved in evidence collection and evaluation. Fact Model elements are used to build the conceptual model underlying the entire assurance case. **Properties** part defines provenance and timing of the evidence items and evaluations. **Evidence Evaluation** part provides means to establish exact nature of the evidentiary support that document confer on the domain assertions. The **Administration** part defines a Project element which organizes individual evidence items and evaluations into a unit of exchange. The Administrative part also provides several means for managing evidence collections projects.

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.

10.1.1 Element (abstract)

Element class is the root element of the SACM Evidence Metamodel. All other classes in the SACM Evidence Metamodel extend Element.

Semantics

Element class is an abstract class that represents any element of the SACM Evidence Metamodel. Every class of the SACM Evidence Metamodel extends Element directly or indirectly (through other classes). The main subclass of the Element is EvidenceElement, 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 Element: EvidenceElement, EvidenceProperty, EvaluationAttribute, AdministrativeElement, AdministrativeProperty, Description, and Rationale.

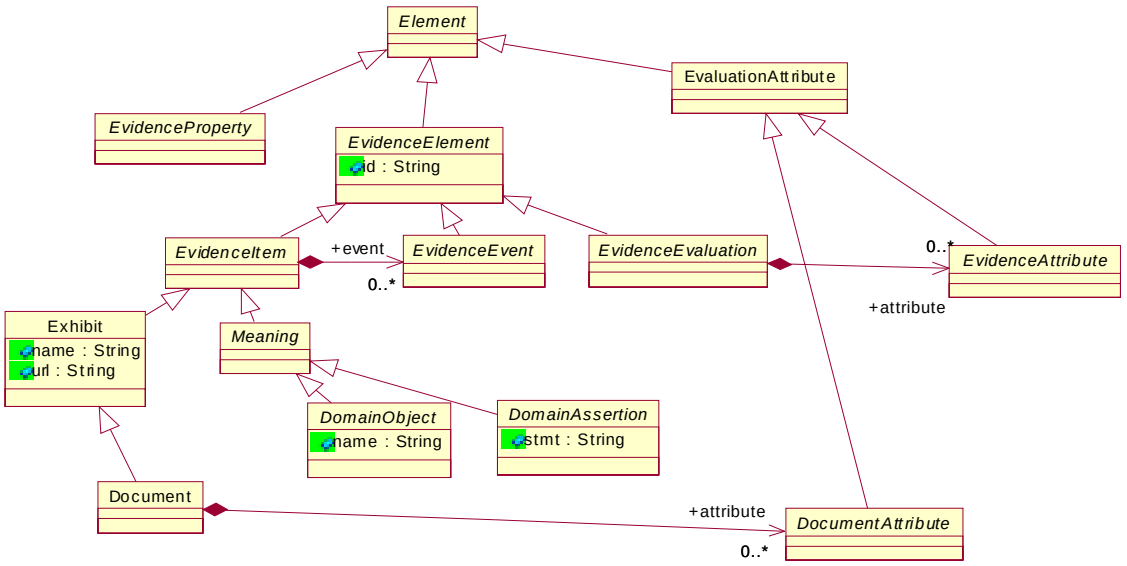


Figure 0.1 - EvidenceElements class diagram

10.1.2 EvidenceElement (abstract)

EvidenceElement is an abstract class that represents the primary elements of the Evidence Metamodel.

Superclass

Element

Attributes

Associations

- id:String
  Globally unique identifier of an SACM Evidence Metamodel evidence element.

Associations

- provenance:Provenance[0..*]
  Provenance properties of the EvidenceElement
- timing:TimingProperty[0..*]
  Timing properties of the EvidenceElement
- description:Description[0..*]
  Description of the EvidenceElement (prose)

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

Constraints

- id is a string that has the following structure:
  - url of the organization that created the evidence element
  - the name of the Evidence Metamodel class of the evidence element
  - the unique number
- id is globally unique, i.e., no two evidence elements of the same type produced by the same organization shall have the same number.

Semantics

EvidenceElement is an abstract class that represents identifiable primary element of the evidence. EvidenceElement may own certain EvidenceProperties. When an EvidenceElement owns an EvidenceProperty, the property represents a relationship between the EvidenceElement and the subject of the EvidenceProperty. Similarly, EvidenceElement may own certain EvidenceAttribute. When an EvidenceElement owns an EvidenceAttribute, the attribute represents a relationship between the EvidenceElement and the subject of the EvidenceAttribute.

10.1.3 EvidenceProperty (abstract)

EvidenceProperty is an abstract class that represents various parts of the primary evidence elements.

Superclass

Element

Semantics

EvidenceProperty is owned by various Element as appropriate. EvidenceProperty represents fundamental properties of the EvidenceElement, as opposed to EvaluationAttribute, which represent assertions that potentially can be disputed. EvidenceProperty involves one or more subjects, specified either as attributes or the associations of the EvidenceProperty element. Each EvidenceProperty represents a relationship between the Element that owns it and its subject.

10.1.4 EvaluationAttribute (abstract)

EvaluationAttribute is an abstract class that represents certain characteristics of various evidence elements that were asserted during the course of evaluation. Subclasses of EvaluationAttribute include EvidenceAttribute and DocumentAttribute.

Superclass

Element

Semantics

EvaluationAttribute is owned by EvidenceElement object as appropriate, for example EvidenceAttribute is owned by EvidenceEvaluation object and DocumentAttribute is owned by Document object. Each subclass of EvidenceElement defines specific constraints regarding what EvaluationAttribute can be owned and what is their meaning in this context. EvaluationAttribute asserts characteristics that potentially can be disputed as opposed to EvidenceProperty, which represents fundamental properties of the EvidenceElement, AdministrativeElement, and EvaluationAttribute. EvaluationAttribute involves one or more subjects, specified either as attributes or the associations of the EvaluationAttribute element. Each EvaluationAttribute represents a relationship between the EvidenceElement that owns it and its subject.

10.1.5 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, domain objects (representing concrete objects or concepts), or domain assertions (see below). EvidenceItem owns a set of events that represent the lifecycle and the chain of custody of the item.

Superclass

EvidenceElement

Associations

- event:EvidenceEvent [0..*]
  Chain of custody, represented as set of events with time stamps that determined the lifecycle of the evidence item.

Semantics

EvidenceItem represents objects that are collected as evidence. The two subclasses of EvidenceItem are Exhibit, representing physical objects presented as evidence, and Meaning, which represents associated elements of meaning, such as concepts and claims.

10.1.6 Meaning (abstract)

Meaning 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

Meaning is an element of meaning that represents a certain individual concept, a noun concept, verb phrases and claims. Two subclasses of Meaning are DomainObject, representing noun concepts, and DomainAssertion, representing verb concepts and claims.

10.1.7 DomainObject (abstract)

DomainObject 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. DomainObject 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 DomainObject 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. DomainObjects represent concepts in the domain for which the argument is being developed. Many elements of the Evidence Metamodel are concepts of the domain of evidence. In particular, Exhibit and Document is two key concepts in the domain of evidence.

Superclass

EvidenceItem

Attributes

- name:String
  Name of the domain concept

Semantics

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

10.1.8 DomainAssertion (abstract)

DomainAssertion is an abstract class that represents propositions that are involved in evidence collection. In particular, DomainAssertion involves DomainObject that represent a individual concepts corresponding to concrete physical things, collection of things, referred to as a whole, or concepts. DomainAssertions represent propositions about the domain for which the argument is being developed. Many elements of the Evidence Metamodel are propositions about the domain of evidence. In particular, EvidenceEvaluation and EvaluationAttribute are among the key propositions in the domain of evidence.

Superclass

EvidenceItem

Attributes

- stmt:String
  The statement that is the expression of the domain assertion (verbalization of the statement in a natural language).

Semantics

DomainAssertion is an element of meaning that represents a certain proposition. FormalAssertion subclass, introduced in Section 9.1, “Formal Assertions Class Diagram,” uses elements of fact model and a formal reference to an SBVR vocabulary to represent precise meaning of the assertion. DomainClaim subclass represents an informal assertion.

10.1.9 EvidenceEvent (abstract)

The lifecycle of an EvidenceItem is determined by several events, such as Creation, Acquisition or Derivation of an EvidenceItem, Transfer of an EvidenceItem, Evaluation of an EvidenceItem and Revocation of EvidenceItem. EvidenceEvent is a characteristic of an EvidenceItem. EvidenceEvent may further have some properties providing the detail of the event, such as the Provenance, Timing, and Custody properties. Analysis of EvidenceEvents establishes the entire chain of custody of an evidence item.

Superclass

EvidenceElement

Semantics

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

10.1.10EvidenceEvaluation (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 placeholders for EvidenceProperties and EvaluationAttributes.

Superclass

Element

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 Exhibits

11.1 Exhibit Class Diagram

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. This class diagram defines classes Exhibit and Document and their owned properties, represented by the subclasses of the abstract class ExhibitProperties and DocumentProperties.

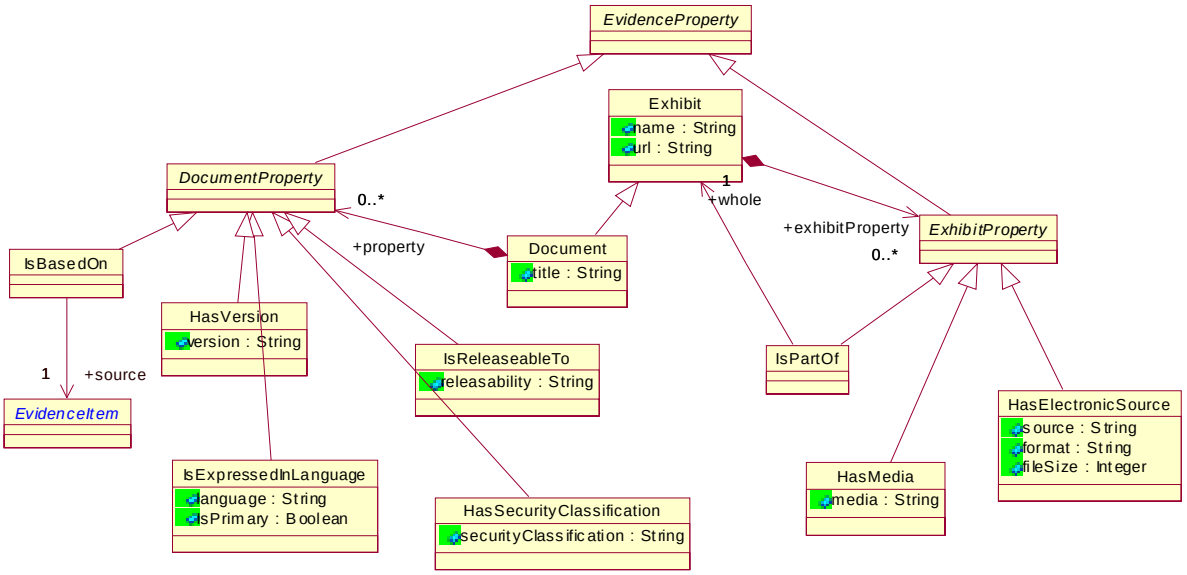


Figure 11.1 - Exhibits class diagram

11.1.1 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 the 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

- exhibitProperty:ExhibitProperty[0..*]
  The set of common physical 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 defines 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.

11.1.2 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. DomainAssertion and DomainObject on the other hand are representations of some meaning rather than of an expression of a meaning (direct or indirect). DomainObject 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 the 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

EvidenceItem

Attributes

- title:String
  The full title of the document
- shortTitle:String
  The short title of the document

Associations

- property:DocumentProperty[0..*]
  The set of common properties of the document.
- attribute:DocumentAttribute[0..*]
  Set of additional attributes of the document that characterize its quality as a source of evidentiary support.

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.

11.1.3 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.

11.1.4 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.”

11.1.5 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.

11.1.6 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.”

11.1.7 Document Property

This class defines common characteristics of documents. Additional characteristics from the evidence viewpoint are represented separately by DocumentAttribute class. Physical characteristics of a document are defined using a separate ExhibitProperty.

Superclass

EvidenceProperty

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.

11.1.8 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

DocumentProperty

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.

11.1.9 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.”

11.1.10HasVersion

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.”

11.1.11 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.”

11.1.12 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

12 Fact Model

The Facts Model defines Evidence Metamodel elements for representing the elements of meaning involved in interpretation and evaluation of evidence, and specifically, required for precisely representing assertions and claims. These fine-grained evidence elements represent individual assertions based on some pre-defined conceptual model. The two fundamental classes of the fact model are Object and Assertion. An Object is an object of significance, about which information needs to be known or held. Usually an Object corresponds to an Exhibit. Exhibit element emphasized the physical object (an SBVR thing element) while an Object emphasized the associated element of meaning. An Assertion is a relationship taken as an element of 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, hence the name fact model. 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 the assurance case, i.e., be agreed upon facts.

So, an Assertion element represents a fact involving one or more Objects. A RoleBinding element represents an association, linkage, or connection between Objects within the fact that describes their role within the fact. RoleBinding represents some semantic association between entities of evidence information.

Fact model elements correspond to some external ontology or vocabulary. Therefore in the SACM Evidence Metamodel, the superclasses of Object and Assertion are called DomainObject and DomainAssertion respectedly as these elements are part of the conceptual model of the Domain for which the assurance case is being developed.

The SACM Facts package is aligned with the OMG SBVR specification, in particular Object can be linked to SBVR IndividualConcept and Assertion can be linked to SBVR fact.

This alignment is important since the Evidence Metamodel can be viewed as a standard vocabulary related to descriptions of evidence. SBVR rules can be written using this vocabulary to formally describe further properties of evidence information. Such vocabulary is presented in Appendix 1.

The SACM Evidence Metamodel is also aligned with the RDF. Object and Assertion can be represented as RDF resources, and RoleBinding - as RDF triples.

12.1 Formal Assertions Class Diagram

The FormalAssertions class diagram focuses at the Assertion as the key element of the fact model underlying the Assurance Case.

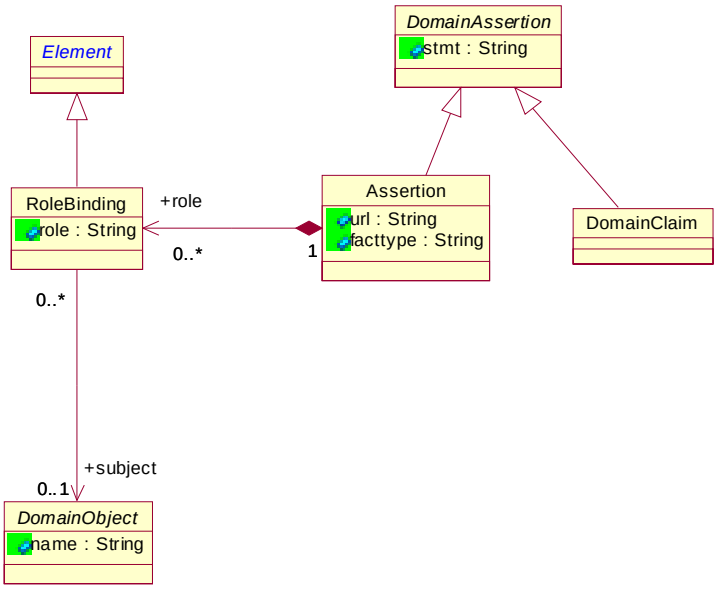


Figure 12.1 - Formal Assertions Class Diagram

12.1.1 Assertion

An Assertion is a relationship involving one or more domain objects, taken as an element of claim 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 conceptual model underlying the assurance case. Assertion represents a fact about the domain for which the assurance case is being developed.

Superclass

DomainAssertion

Attributes

- url:String
  URL of the external SBVR vocabulary or OWL vocabulary defining the formal assertion.
- facctype:String
  Designation of the fact type

Associations

- role:RoleBinding[0..*]
  Set of role bindings that further describe which DomainObject are bound to the roles that are determined by the fact type.

Semantics

Assertion is an element of meaning that states existence of a relationship between several individual domain 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 the assurance case will develop such vocabularies for the corresponding domain. In a semi-formal assurance case the nature of the relationship can be described informally through a stmt property. In this case the URL property and the facttype property shall not be used. However the references to exact DomainObjects through RoleBinding elements can be still stated. The stmt property of the DomainAssertion element provides the verbalization of the fact, which is the expression of the fact in a natural language. For informal assurance cases, a DomainClaim element can be used, which only contains the verbalization of the claim in a natural language.

12.1.2 DomainClaim

DomainClaim is an element of meaning which represents an informal proposition about the state of affairs in the domain about which the assurance case is developed.

Superclass

DomainAssertion

Semantics

DomainClaim is an element of meaning that states a generic proposition about the assurance case domain. DomainClaim is an informal element that represents claim as prose in a natural language (formal or informal), without identifying its structure. DomainClaim 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 state existence of a formally defined relationship between several individual domain objects and involve several subjects bound to specific roles. 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.

12.1.3 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

Element

Attributes

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

Associations

- subject:DomainObject[0..1]
  DomainObject 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 the 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 separates DomainObject from DomainAssertion. As a consequence, this limits the possibility to represent assertions about assertions.

12.2 Formal Objects Class Diagram

The FormalObjects class diagram focuses at objects are they are involved in assertions comprising the fact model underlying the Assurance Case.

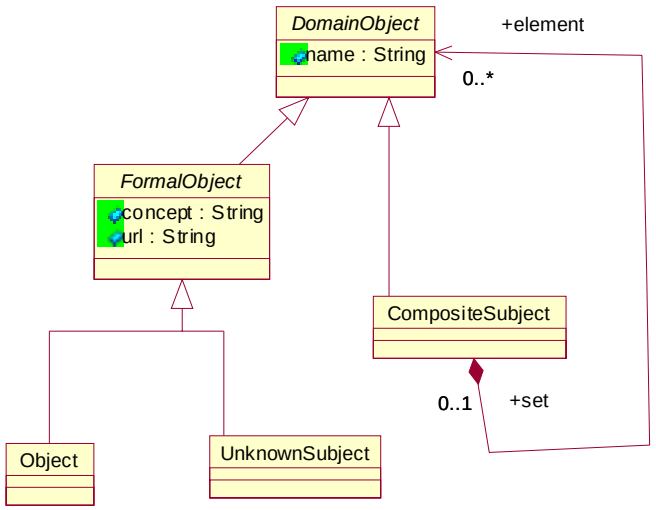


Figure 12.2 - Formal Objects Class Diagram

12.2.1 FormalObject (abstract)

FormalObject represents an individual that can be involved in assertions constituting the conceptual model underlying the assurance case.

Superclass

DomainObject

Attributes

- concept:String
  Designation of the concept.
- url:String
  URL of the external SBVR vocabulary or OWL vocabulary defining the formal concept.

Semantics

FormalObject is an element of meaning. FormalObject shall be used in fact model underlying the assurance case to represent subjects of assertions, in particular when more than one assertion refers to the same subject.

12.2.2 Object

FormalObject represents a known individual that can be involved in assertions constituting the conceptual model underlying the assurance case.

Superclass

FormalObject

Semantics

Object is an element of meaning. Object shall be used in fact model underlying the 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.

12.2.3 UnknownSubject

UnknownSubject represents an unknown individual, existence of which is however is determined by the pattern of relationships in the fact model, and that is involved in assertions constituting the conceptual model underlying the assurance case.

Superclass

FormalObject

Semantics

UnknownSubject is an element of meaning. UnknownSubject shall be used in fact model underlying the assurance case to represent unknown subjects of assertions, in particular when more than one assertion refers to the same subject.

12.2.4 CompositeSubject

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

Superclass

DomainObject

Associations

- element:DomainObject[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 fact model underlying the assurance case to represent groups of object of assertions, in particular when more than one assertion refers to the same group.

13 Evidence Properties

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

13.1 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.

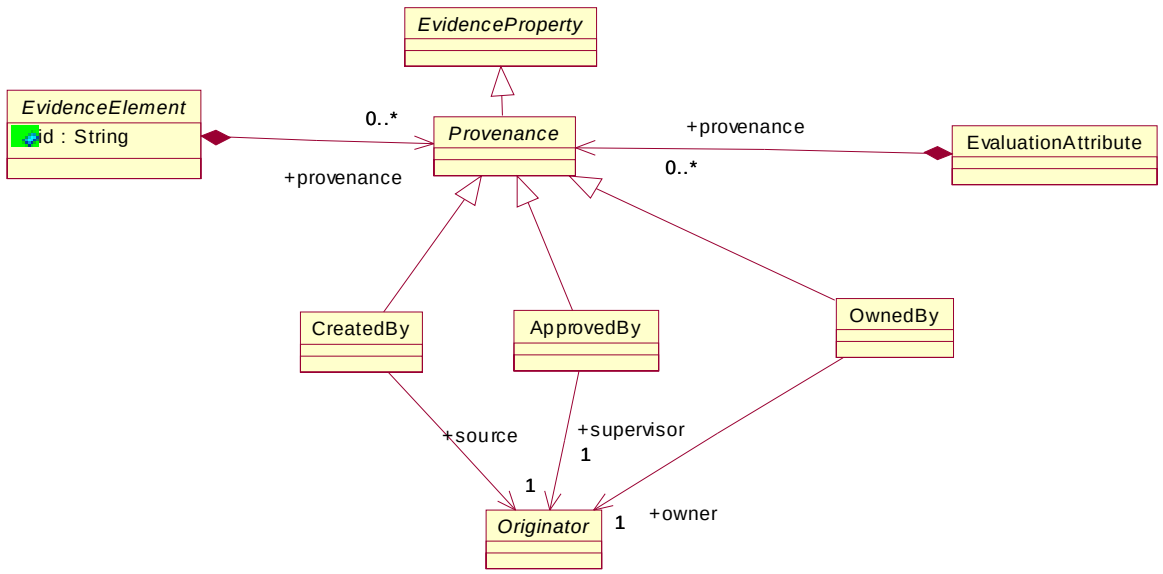


Figure 13.1 - Provenance Class Diagram

13.1.1 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.

13.1.2 CreatedBy

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

Superclass

Provenance

Associations

- source:Originator[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 originator, defined by Originator object. Originator 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 originator.”

13.1.3 ApprovedBy

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

Superclass

Provenance

Associations

- supervisor:Originator[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 originator, defined by Originator object. Originator 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 originator.”

13.1.4 OwnedBy

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

Superclass

Provenance

Associations

- owner:Originator[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 Originator object. Originator 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 originator.”

13.2 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.

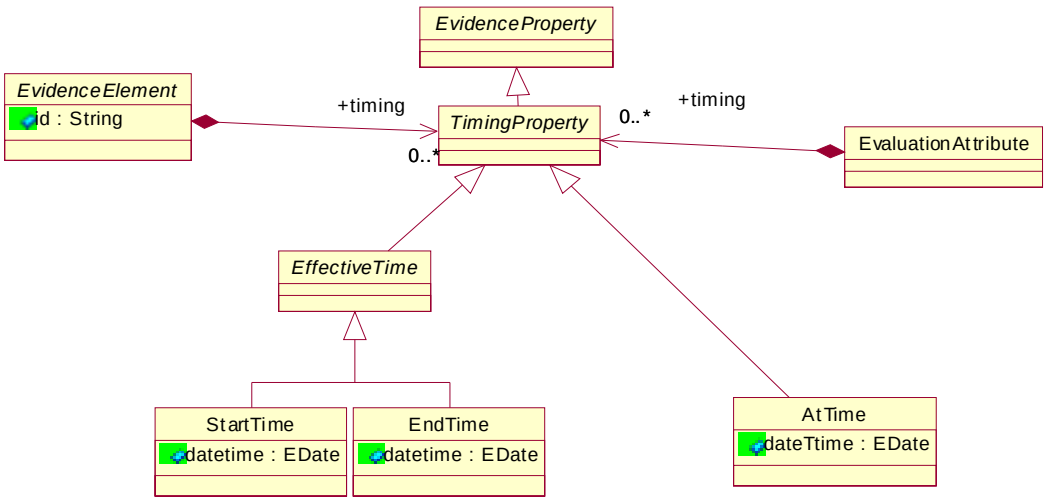


Figure 13.2 - Timing Class Diagram

13.2.1 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.

13.2.2 EffectiveTime (abstract)

EffectiveTime element is an abstract class that corresponds to effective time interval associated with the owner element. This element was added for readability purposes. Specific characteristics related to the effective time interval are defined by concrete subclasses of EffectiveTime element.

Superclass

TimingProperty

Semantics

EffectiveTime 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 detailed of the effective time interval of the owner object, defined by a particular concrete subclass of the TimingProperty element.

13.2.3 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.

13.2.4 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.

13.2.5 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.

13.3 Descriptions Class Diagram

Descriptions Class Diagram focuses at the informal annotations to the elements of the Evidence Metamodel

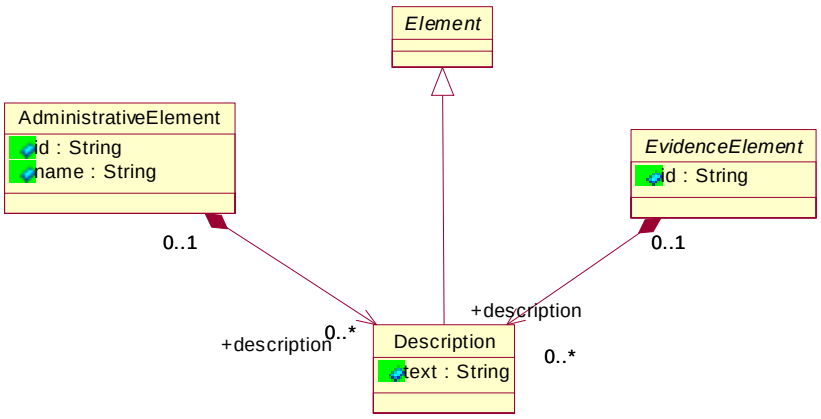


Figure 13.3 - Descriptions Class Diagram

13.3.1 Description

Description element allows informal descriptions to be associated with some elements of the Evidence Metamodel.

Superclass

Element

Attributes

- text:String
  Text of the description (prose)

Semantics

Description element represents a property of the owner EvidenceElement object or AdministrativeElement object. Description element represents association an informal textual description with the owner object.

The characteristic of Description is expressed by a sentential form “Element is described by text.”

13.4 EvidenceEvents Class Diagram

The EvidenceEvents Class Diagram focuses at the Events that determine the lifecycle of the evidence element, as well as the custody properties of the evidence element. EvidenceEvents allow storing the entire Chain of Custody of an evidence element. EvidenceEvents set the context for the time stamps and custody properties. EvidenceEvents are properties of owner object.

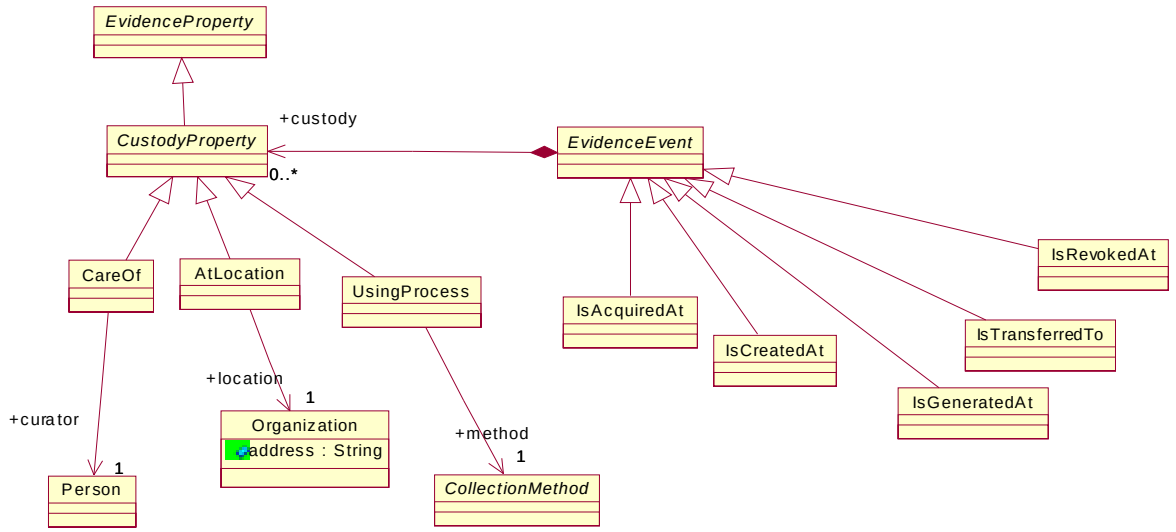


Figure 13.4 - EvidenceEvent Class Diagram

13.4.1 EvidenceEvent (abstract)

EvidenceEvent is an abstract class that represents any event related to the lifecycle of evidence element. Concrete evidence events are defined as subclasses of the EvidenceEvent element.

Superclass

EvidenceElement

Associations

- custody:CustodyProperty[0..*]
  Set of custody properties of the event

Semantics

EvidenceEvent element represents a property of the owner EvidenceElement object. This element is an abstract class that establishes a relationship between the owner object and the particular event description and associated characteristics, defined by a particular concrete subclass of the EvidenceEvent element and its owned properties, such as CustodyProperty, Provenance, and TimingProperty.

13.4.2 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	The originator of the acquisition	Element *is acquired by* originator
ApprovedBy	The person or organization who approved the acquisition.	*Acquisition of* element *is approved by* originator
OwnedBy	Organization which executed acquisition of the evidence element and has custody of the evidence element.	Element *is owned by* originator
CareOf	The curator of the evidence element within the owner organization.	Person *is curator 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

13.4.3 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	The source of the evidence element	Element *is created by* originator
ApprovedBy	The person or organization who approved the creation of the evidence element.	*Creation of* element *is approved by* originator
OwnedBy	Organization which created the evidence element.	Element *is owned by* originator
CareOf	The curator of the evidence element within the owner organization.	Person *is curator 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

13.4.4 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	The source of the transfer (the previous curator of the evidence element).	Element *is transferred from* originator
ApprovedBy	The person or organization who approved the transfer of the evidence element.	*Transfer of* element *is approved by* originator
OwnedBy	Organization which established custody over the evidence element.	Element *is owned by* originator
CareOf	The curator of the evidence element.	Person *is curator 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 roles 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

13.4.5 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 emphasizes that the evidence element is no longer admissible for supporting arguments. Revocation of an evidence element is stronger than the end of the validation period of an evidence element.

Superclass

EvidenceEvent

Semantics

IsIsRevokedAt element represents a property of the owner EvidenceElement object. IsRevokedAt element represents the state of affairs that the owner object is revoked. IsRevokedAt element may own further properties establishing additional details about the transfer event.

Property	Meaning	Verbalization
AtTime	Time of the revocation	Element *is revoked at* time
EffectiveTime	N/A	
CreatedBy	The executor of the revocation, if applicable.	Element *is revoked by* originator
ApprovedBy	The person or organization who approved the revocation of the evidence element.	*Revocation of* element *is approved by* originator
OwnedBy	Organization which established custody over the evidence element, if applicable.	Element *is owned by* originator
CareOf	The curator of the evidence element, if applicable.	Person *is curator 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

13.4.6 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	The executor of the generation of the evidence element.	Element *is generated by* originator
ApprovedBy	The person or organization who approved the generation of the evidence element.	*Generation of* element *is approved by* originator
OwnedBy	Organization which executed generation of the evidence element.	Element *is owned by* originator
CareOf	The curator of the evidence element within the owner organization.	Person *is curator 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

13.4.7 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).

13.4.8 CareOf

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

Superclass

CustodyProperty

Associations

- curator:Person[1]
  Curator of the evidence element associated with owner EvidenceEvent.

Semantics

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

13.4.9 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).

13.4.10 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 EvidenceEventobject (with the additional constraints imposed by the semantics of the owned EvidenceEvent).

14 Evidence Evaluation

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

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

- Quality Attributes of Documents, such as Primary or secondary, Document: original or derived, Consistency, Completeness, Accuracy.
- Quality Attributes of Evidentiary Support, such as Direct or indirect, Relevance, Confidence, Strength, Significance.
- Interpretation of Evidence: what an evidence item "Is", what is "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 focuses at the evidence relations between EvidenceItem, such as Exhibit and DomainAssertion, such as DomainClaim. EvidenceRelation elements allow specifying exact statement of evidentiary support between EvidenceItem and DomainAssertion.

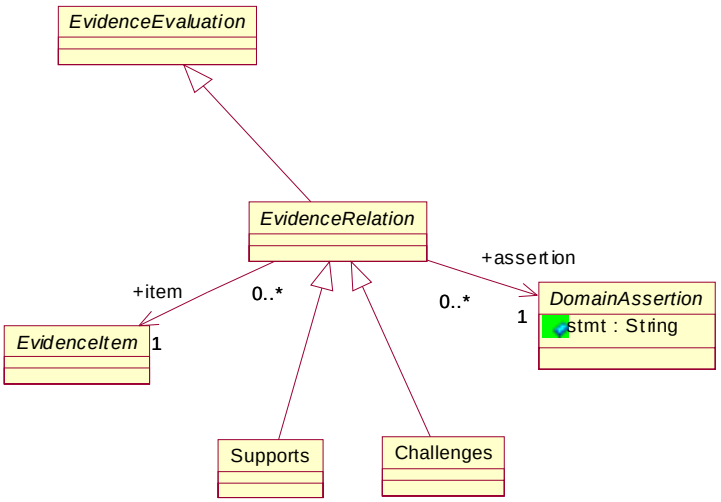


Figure 14.1 - EvidenceRelations Class Diagram

14.1.1 EvidenceRelation (abstract)

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

Superclass

EvidenceEvaluation

Associations

- item:EvidenceItem[1]
  The EvidenceItem object, such as an Exhibit or a Document that has evidentiary relations to a DomainAssertion object such as a DomainClaim.
- assertion:DomainAssertion[1]
  DomainAssertion object that receives an evidentiary relation from the EvidenceItem object.

Constraints

- DomainAssertion 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 DomainAssertion objects that is asserted during the evidence evaluation.

14.1.2 Supports

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

Superclass

EvidenceRelation

Semantics

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

14.1.3 Challenges

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

Superclass

EvidenceRelation

Semantics

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

14.2 Evidence Attributes Class Diagram

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

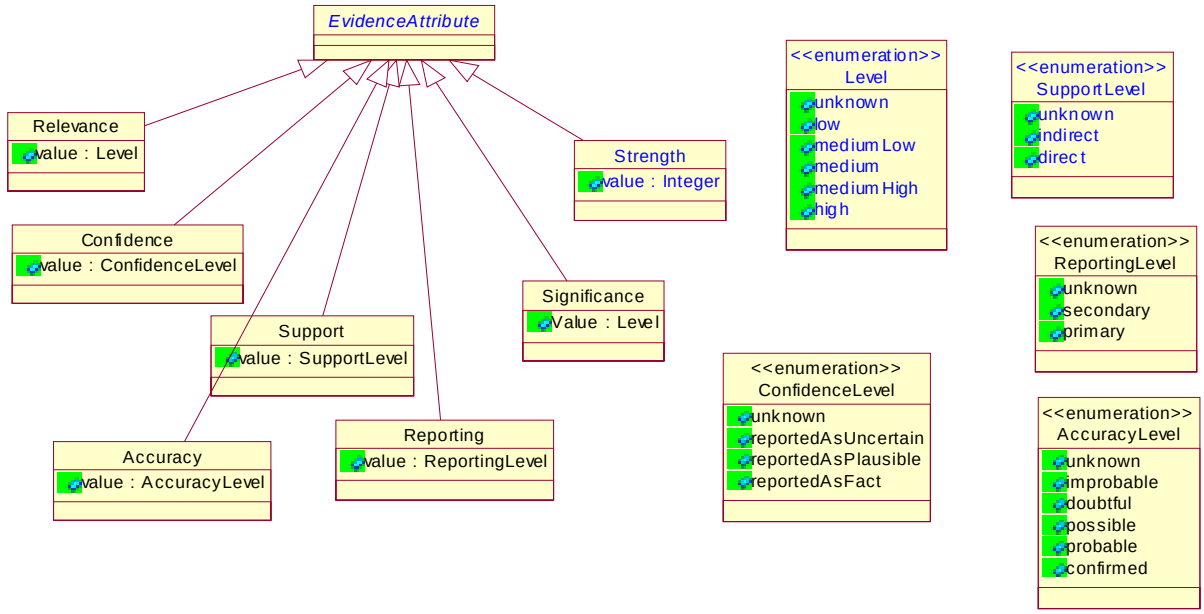


Figure 14.2 - EvidenceAttribute Class Diagram

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

14.2.2 SupportLevel (enumeration)

SupportLevel enumeration specifies the support level.

Attributes

- 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** DomainAssertion," "EvidenceItem *is a secondary record* **of** DomainAssertion."

14.2.4 ReportingLevel (enumeration)

ReportingLevel enumeration specifies the reporting levels.

Attributes

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

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.

Attributes

- 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, AdministrativeElement, and EvaluationAttribute. Confidence element includes the relevance level.

14.2.8 ConfidenceLevel (enumeration)

The ConfidenceLevel enumeration class defines confidence levels.

Attributes

- 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.

Attributes

- 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* DomainAssertion *with strength* 50," "EvidenceItem *challenges* DomainAssertion *with strength* 10."

14.3 Document Attributes Class Diagram

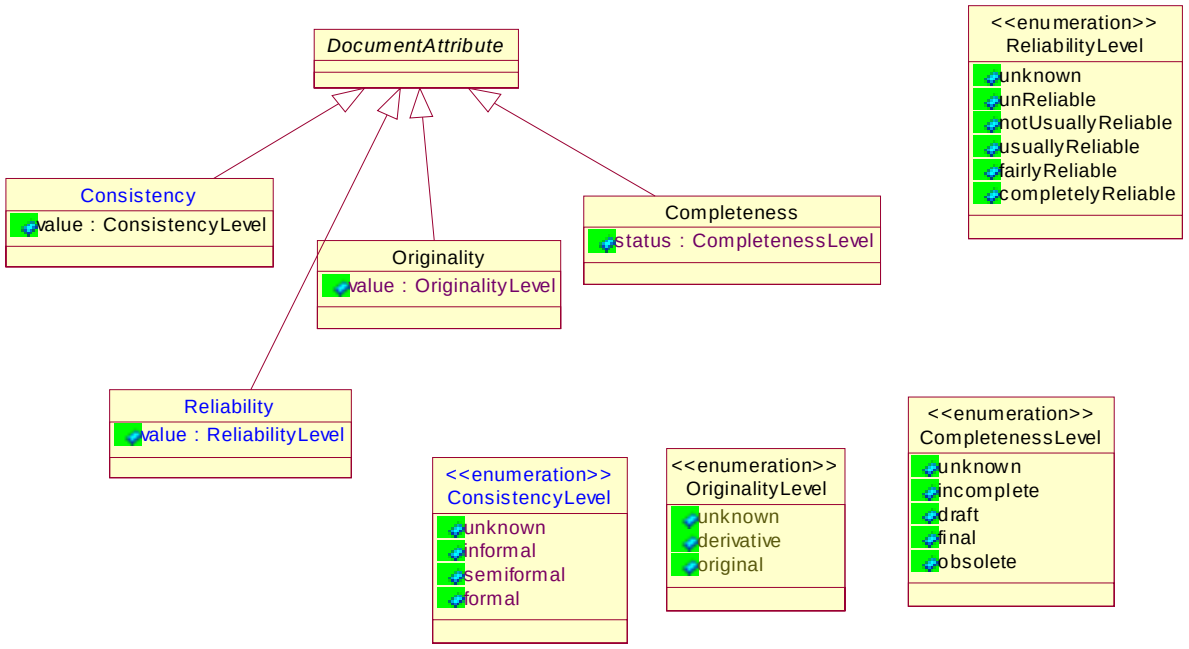


Figure 14.3 - DocumentAttributes Class Diagram

14.3.1 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.

14.3.2 OriginalityLevel (enumeration)

OriginalityLevel enumeration class defines the Originality levels.

Attributes

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

14.3.3 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.

14.3.4 ConsistencyLevel (enumeration)

The ConsistencyLevel enumeration class defines consistency levels.

Attributes

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

14.3.5 Completeness

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.

14.3.6 CompletenessLevel (enumeration)

The CompletenessLevel enumeration class defines completeness levels.

Attributes

- 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.

14.3.7 Reliability

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.

14.3.8 ReliabilityLevel (enumeration)

The ReliabilityLevel enumeration class defines reliability levels.

Attributes

- 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.

14.4 EvidenceInterpretation Class Diagram

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

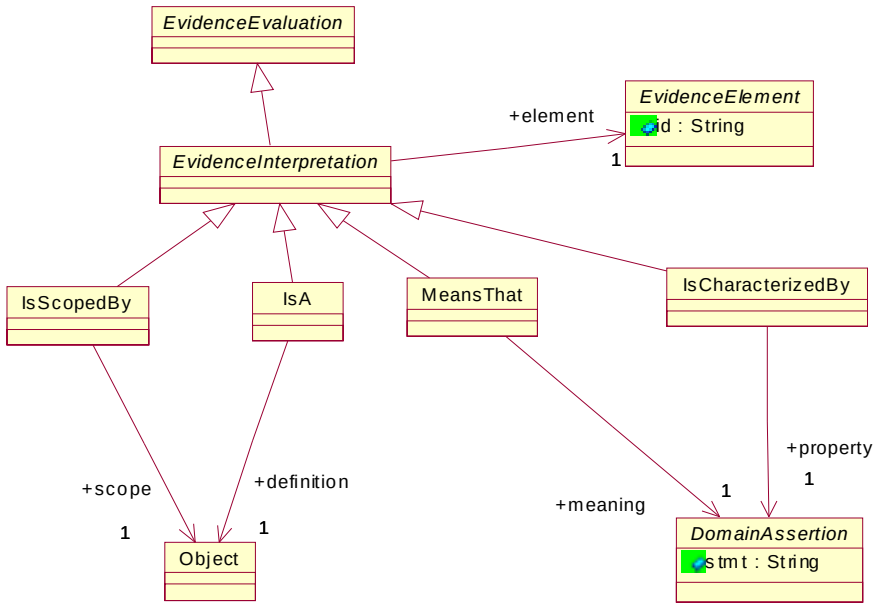


Figure 14.4 - EvidenceInterpretation Class Diagram

14.4.1 EvidenceInterpretation (abstract)

EvidenceInterpretation is an abstract class that represents a relation between one EvidenceElement and one Meaning element. 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

- element: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 Meaning object that is asserted during the evidence evaluation.

14.4.2 IsA

IsA represents a fundamental relation between one EvidenceElement and one Object element which defines the general concept of the source EvidenceElement. The actual concept is given by reference to an external formal vocabulary or ontology. The following statements are examples of IsA:

- “This metric is a McCabe’s Cyclomatic Complexity Metric.”
- “This report is a penetration testing report.”

Superclass

EvidenceInterpretation

Associations

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

Constraints

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

Semantics

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

This characteristic is verbalized as follows: “EvidenceElement is an Object.”

14.4.3 MeansThat

MeansThat represents a fundamental relation between one EvidenceElement and one DomainAssertion 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 DomainClaim 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:DomainAssertion[1]
  DomainAssertion element

Constraints

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

Semantics

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

This characteristic is verbalized as follows: “EvidenceElement means that DomainAssertion.”

14.4.4 IsCharacterizedBy

IsCharacterizedBy represents a relation between one EvidenceElement and one DomainAssertion element which defines the property 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

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

Semantics

The IsCharacterizedBy element asserts a state of affairs that the EvidenceElement, identified as the element of the MeansThat element, is characterized by a proposition, in which the subject is bound to one of the role, and which is represented by the Object that is identified as the property of the IsCharacterizedBy element.

This characteristic is verbalized as follows: “EvidenceElement is characterized by DomainAssertion.”

14.4.5 IsScopedBy

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

Superclass

EvidenceInterpretation

Associations

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

Constraints

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

Semantics

“Scope” is defined as either 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 element of the IsScopedBy element, is delimited by Object that is identified as the scope of the IsScopedBy element. The Object may represent an individual concept or an abstract concept.

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

14.5 Evidence Observations Class Diagram

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

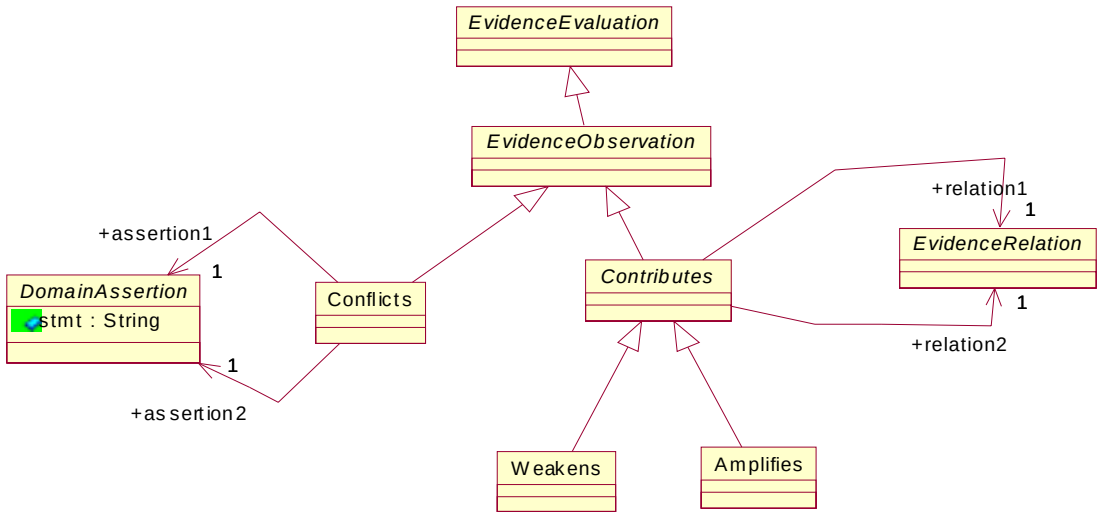


Figure 14.5 - EvidenceObservations Class Diagram

14.5.1 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.

14.5.2 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

- assertion1: DomainAssertion[1]
  The subject DomainAssertion
- assertion2: DomainAssertion[1]
  The object DomainAssertion

Semantics

The Conflicts element asserts a state of affairs that the DomainAssertion-1, identified as the assertion1 of the Conflicts element, is in conflict with DomainAssertion 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: “DomainAssertion-1 *conflicts* DomainAssertion-2”

14.5.3 Contributes (abstract)

Contributes element asserts dependency between two EvidenceRelation elements. For example, one may assert that the evidence that support the claim that “Bob is married to Alice” weakens the claim that “Bob is single” (see weakens element). These dependencies help further evaluation of evidence.

Superclass

EvidenceObservation

Associations

- relation1: EvidenceRelation[1]
  The subject EvidenceRelation
- relation2: 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.

14.5.4 Weakens

Weakens element asserts that one EvidenceRelation-1 element weakens another EvidenceRelation-2 element. This has a different meaning that the statement that any evidence supporting DomainAssertion-1 that is the assertion of EvidenceRelation-1, challenges the DomainAssertion-2 that is the assertion of the EvidenceRelation-2. Weakens relation may imply a conflict between DomainAssertion-1 and DomainAssertion-2. In that case evidence in support of DomainAssertion-1 is not relevant to DomainAssertion-2. For example, one may assert that the evidence that support the claim that “Bob is married to Alice” weakens the claim that “Bob is single.” Weakens dependencies help further evaluation of evidence.

Superclass

Contributes

Semantics

The Weakens element asserts a state of affairs that the EvidenceRelation-1, identified as the relation1 of the Weakens element, weakens EvidenceRelation-2 that is identified as the relation2 of the Weakness element. Weakens may imply a conflict between DomainAssertion-1 that is identified as assertion of EvidenceRelation-1 and DomainAssertion-2 that is identified as assertion of EvidenceRelation-2.

This characteristic is verbalized as follows: “*Evidentiary support to* DomainAssertion-1 *weakens evidentiary support to* DomainAssertion-2”

14.5.5 Amplifies

Amplifies element asserts that one EvidenceRelation-1 element amplifies another EvidenceRelation-2 element. This has a different meaning that the statement that any evidence supporting DomainAssertion-1 that is the assertion of EvidenceRelation-1, supports the DomainAssertion-2 that is the assertion of the EvidenceRelation-2. Amplifies relation may imply a coupling between DomainAssertion-1 and DomainAssertion-2. In that case evidence in support of DomainAssertion-1 may be relevant to DomainAssertion-2. For example, one may assert that the evidence that support the claim that “Bob is married to Alice” amplifies the claim that “Bob is not single.” Amplifies dependencies help further evaluation of evidence.

Superclass

Contributes

Semantics

The Amplifies element asserts a state of affairs that the EvidenceRelation-1, identified as the relation1 of the Weakens element, amplifies EvidenceRelation-2 that is identified as the relation2 of the Amplifies element. Amplifies may imply a coupling between DomainAssertion-1 that is identified as assertion of EvidenceRelation-1 and DomainAssertion-2 that is identified as assertion of EvidenceRelation-2.

This characteristic is verbalized as follows: "*Evidentiary support to* DomainAssertion-1 *amplifies evidentiary support to* DomainAssertion-2"

14.6 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 DomainAssertions. The Evidence Metamodel provides three options: Negate EvidenceRelation, Refute a DomainAssertion, 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.

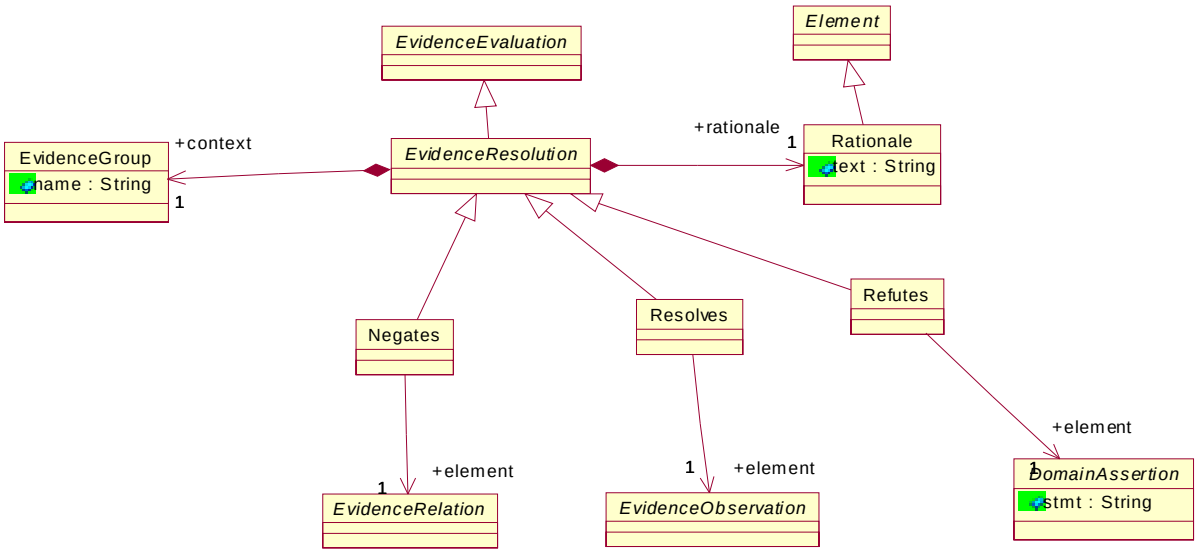


Figure 14.6 - EvidenceResolutions Class Diagram

14.6.1 EvidenceResolution (abstract)

EvidenceResolution is an abstract class that asserts resolution to conflicts between two domain assertions either directly or through refuting some domain assertion or negating some evidence relations.

Superclass

EvidenceEvaluation

Associations

- context:EvidenceGroup[0..1]
  The set of evidence element that provides the context for the resolution.
- rationale:Rationale[1]
  The rationale for the resolution (prose).

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.

14.6.2 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 DomainAssertion by claiming that the evidentiary support to the DomainAssertion 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 DomainAssertion. 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 DomainAssertion.

14.6.3 Refutes

Refutes element asserts direct refutation of a DomainAssertion. 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 DomainAssertion. Context of the refutation plays a more important role, because the conflicting claims with strong evidentiary support need to de identified.

Superclass

EvidenceEvaluation

Associations

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

Semantics

The Refutes element asserts direct refutation of a certain DomainAssertion. 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 DomainAssertion being refuted.

14.6.4 Resolves

Resolves element asserts resolution of a conflict between two DomainAssertion. 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 DomainAssertion. Context of the resolution plays a more important role, 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 DomainAssertion).

Semantics

The Resolves element asserts resolution of a conflict between two DomainAssertion. 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.

14.7 Evaluation Context Class Diagram

The EvidenceContext Class Diagram defines several utility elements that assist evaluation of evidence. This Class Diagram includes an EvidenceGroup element that allows aggregation of evidence elements and assertions to aggregations. This Class Diagram also involves two EvaluationEvaluation elements: ProvidesContext and Supercedes.

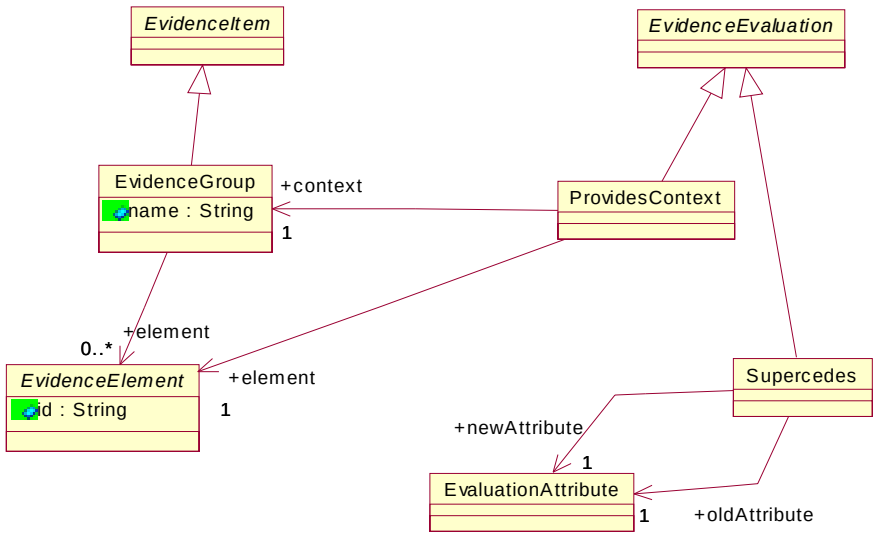


Figure 14.7 - EvaluationContext Class Diagram

14.7.1 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 and therefore can play a role of a named container for evidence items that can be used on both sided of the evidence relation. An EvidenceElement may be a member of more than one EvidenceGroup.

15 Administration

Administration package defines key framework elements that determine patterns for constructing representations of evidence information.

15.1 Project Class Diagram

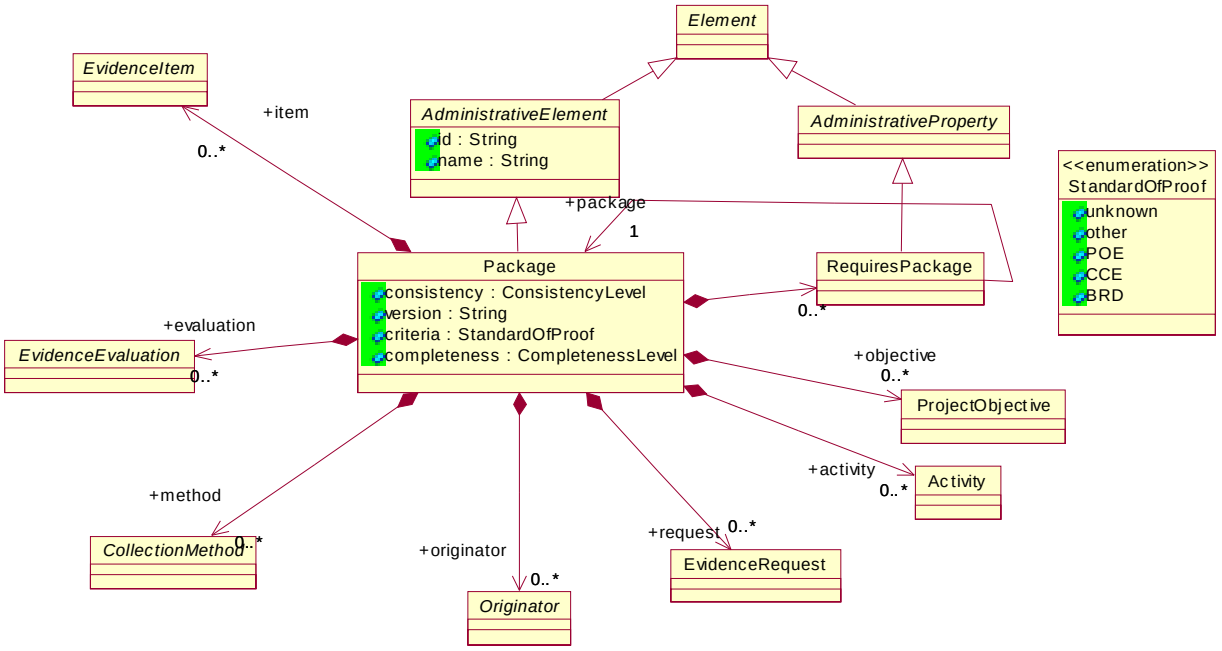


Figure 15.1 - Project Class Diagram

15.1.1 AdministrativeElement (abstract)

AdministrativeElement is an abstract class representing non-essential elements of the Evidence Metamodel that assist in managing evidence collection, interpretation, evaluation, and exchange processes.

Superclass

Element

Attributes

- id:String
  Globally unique identifier of a SACM evidence element.
- name:String
  Name of the administrative element.

Semantics

15.1.2 Package

Package element is the root element that owns EvidenceItem, EvidenceEvaluation elements, as well as other auxiliary elements related to the processes of evidence identification, collection, interpretation, evaluation, and management.

Superclass

AdministrativeElement

Attributes

- consistency:ConsistencyLevel
  Globally unique identifier of a SACM evidence element.
- version:String
  version of the package.
- criteria:StandardOfProof
  Standard of Proof used for evaluation of evidence.
- completeness:CompletenessLevel
  Level of completeness of the evidence package.

Association

- :requiresPackage[0..*]
  List of other evidence packages that are required by this package.
- item:EvidenceItem[0..*]
  List of evidence items.
- evaluation:EvidenceEvaluation[0..*]
  List of evaluations.
- method:CollectionMethod[0..*]
  List of evidence collections methods, including tools.
- originator:Originator[0..*]
  List of personnel and organizations involves in evidence collection project.
- request:Request[0..*]
  List of evidence collection requests.
- activity:Activity[0..*]
  List of project activities.
- objective:ProjectObjective[0..*]
  List of project objectives.

Constraints

- Package shall not be the object of the requiresPackage relation owned by the Package, either directly or indirectly through requiresPackage of other Packages.
- Any Package that is the object of the requiresPackage relation shall be available for exchange.

- [Completeness of the package with respect to required packages] Any Element that is referenced by any of the Element that defined in the package (i.e., that are members of the lists item, evaluation, method, originator, request, activity, and objective of the Package) shall be also defined in the Package or in one of the Package that are referred to as objects of the requiresPackage 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 packages.

Semantics

Package element is the root element of an Evidence Model. A single Package is a unit of exchange. All Element defined in Package are exchanged together as part of the Package. Elements that are referenced shall be either present in the Package or in one of the Package that is specified as required for the Package. The Evidence Metamodel does not require completeness of the closure of all required packages.

15.1.3 StandardOfProof (enumeration)

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

Attributes

- unknown
  Standard of Proof unknown
- other
  Standard of proof other than those explicitly enumerated
- POE
  Preponderance of 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.
- Genealogical Proof Standard (GPS) 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.

15.1.4 AdministrativeProperty (abstract)

AdminstrativeProperty is an abstract class that represents owned properties of AdminstrativeElement. This class is added to enhance the readability of the model.

Superclass

Element

Semantics

Defined by concrete subsclasses

15.1.5 RequiresPackage

RequiresPackage is an owned property of Package element that represents a state of affairs that the Package requires another package for the resolution of some references.

Superclass

AdministrativeProperty

Associations

- package:Package[1]
  Package that is required for the resolution of some references.

Constraints

- owner Package shall not be the package of the requiresPackage relation, either directly or indirectly.

Semantics

RequiresPackage property contributes to the completeness constraint of the Package. This is a commitment to the set of packages that need to be processed together.

15.2 ProjectActivities Class Diagram

ProjectActivities Class Diagram defines Activity AdministrativeElement and its owned properties. Activity element facilitates management of evidence collection and evaluation processes.

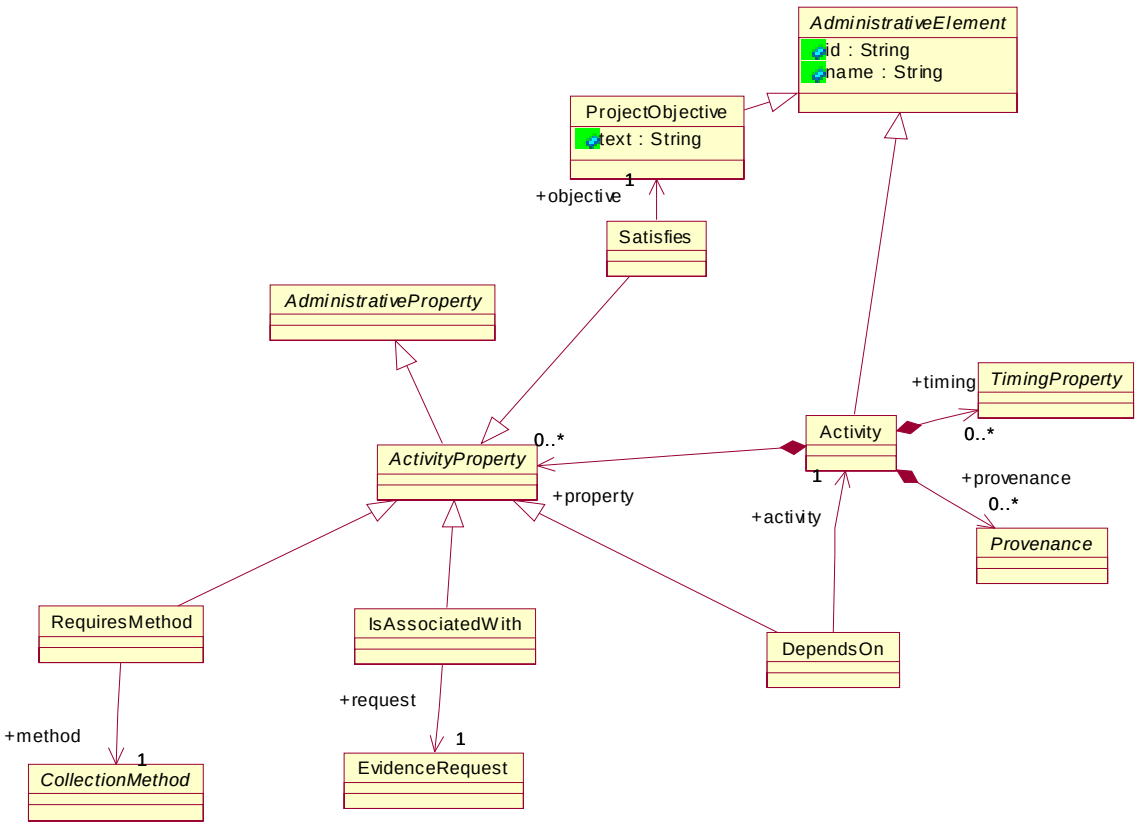


Figure 15.2 - ProjectActivities Class Diagram

15.2.1 Activity

Activity element represents an individual task that either needs to be performed during an evidence collection project (planning purposes), or has been performed during the project (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 collection project. 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 ActivityProperty (abstract)

ActivityProperty is an abstract class that represents owned properties of Activity element.

Superclass

AdministrativeProperty

Semantics

ActivityProperty represents a state of affairs involving the owner Activity object. The nature of the activity properties is defined by the concrete subclasses of ActivityProperty. ActivityProperty is an uncontestable administrative element.

15.2.3 Satisfies

Satisfies element represents a relationship between the owner Activity element and a ProjectObjective object that is identified as the objective attribute of the Satisfies element.

Superclass

ActivityProperty

Associations

- objective:ProjectObjective[1]
  Project objective that is satisfied by the activity.

Semantics

Satisfies element represents a state of affairs that the owner Activity object satisfies the ProjectObjective identified as the objective attribute of the ProjectObjective element.

15.2.4 RequiresMethod

RequiresMethod element represents a relationship between the owner Activity element and a CollectionMethod object that is identified as the method attribute of the RequiresMethod element.

Superclass

ActivityProperty

Associations

- method:CollectionMethod[1]
  Evidence collection method that is required by the activity.

Semantics

RequiresMethod element represents a state of affairs that the owner Activity object requires CollectionMethod (service, method or tool) identified as the method attribute of the RequiresMethod element.

15.2.5 IsAssociatedWith

IsAssociatedWith element represents a relationship between the owner Activity element and an EvidenceRequest object that is identified as the request attribute of the IsAssociatedWith element.

Superclass

ActivityProperty

Associations

- request:EvidenceRequest[1]
  Evidence request that is associated with the activity.

Semantics

IsAssociatedWith element represents a state of affairs that the owner Activity is associated with EvidenceRequest (EvidenceItem to be acquired, create or generated) identified as the request attribute of the IsAssociatedWith element. EvidenceRequest is a placeholder for the corresponding EvidenceItem (usually – for an Exhibit to be collected). ProjectObjective elements can be used to describe the detailed motivation for the EvidenceRequest. RequiresMethod element may be used to provide detail regarding the collection method. Description element may be used to provide additional clarifications.

15.2.6 DependsOn

DependsOn element represents a relationship between the owner Activity element and another Activity element that is identified as the activity attribute of the DependsOn element.

Superclass

ActivityProperty

Associations

- activity:Activity[1]
  Activity that the current activity depends on.

Constraints

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

Semantics

DependsOn element represents a state of affairs that the owner Activity is associated with another Activity identified as the activity attribute of the DependsOn element.

15.3 Methods Class Diagram

The Methods Class Diagram defines several elements that represent evidence collection methods. These elements are subclasses of Object element, so their formal meaning can be further defined through a reference to a formal vocabulary or ontology developed by the corresponding community of interest.

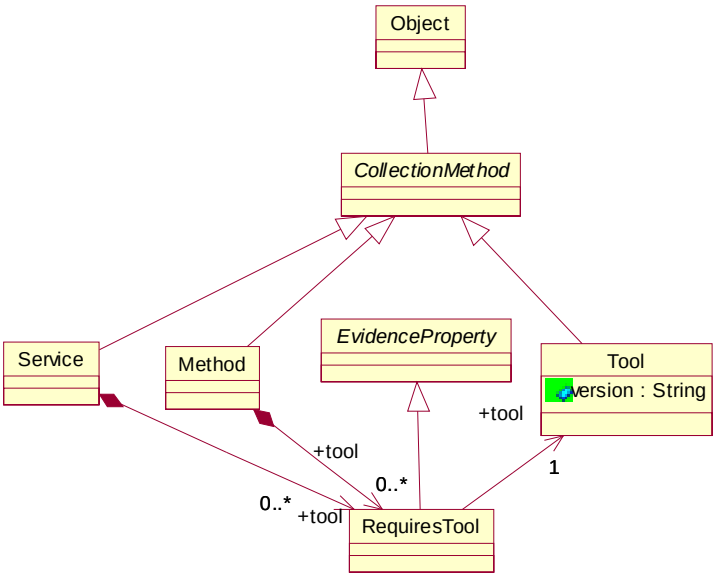


Figure 15.3 - Methods Class Diagram

15.3.1 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.

15.3.2 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.

15.3.3 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.

15.3.4 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

15.4 Originators Class Diagram

The Originators Class Diagram defines several elements that represent sources of evidence. These elements are subclasses of Object element, so their formal meaning can be further defined through a reference to a formal vocabulary or ontology developed by the corresponding community of interest.

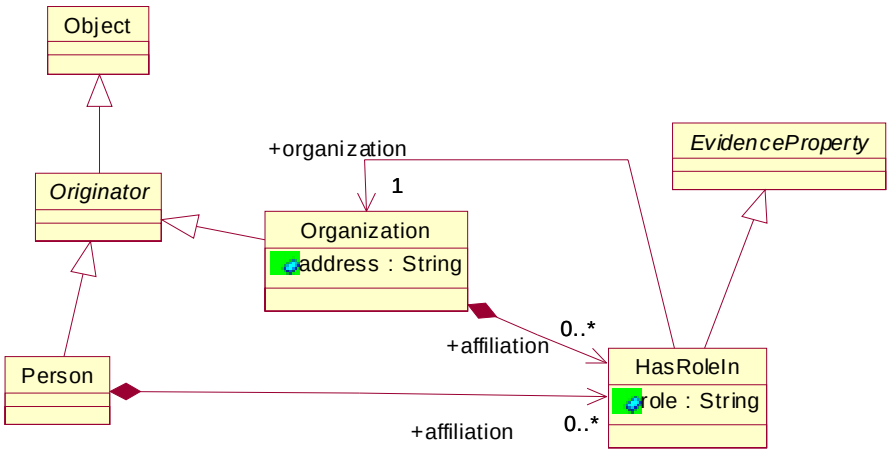


Figure 15.4 - Originators Class Diagram

15.4.1 Originator (abstract)

Originator is an abstract class that represents a source of evidence (Person or Organization) 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.

15.4.2 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

Originator

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.

15.4.3 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

Originator

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.

15.4.4 HasRoleIn

An owned property of Person and Organization

Superclass

EvidenceProperty

Attributes

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

Associations

- organization:Organization[1]
  Organization-2 with which the owner Person or Organization-1 is affiliated.

15.5 Request Class Diagram

Request Class Diagram defines EvidenceRequest element which is an AdministrativeElement, representing a placeholder for EvidenceItem to be collected during the evidence collection project. This element facilitates management of evidence collection projects and exchange of meaningful evidence information at various stages of the evidence collection process.

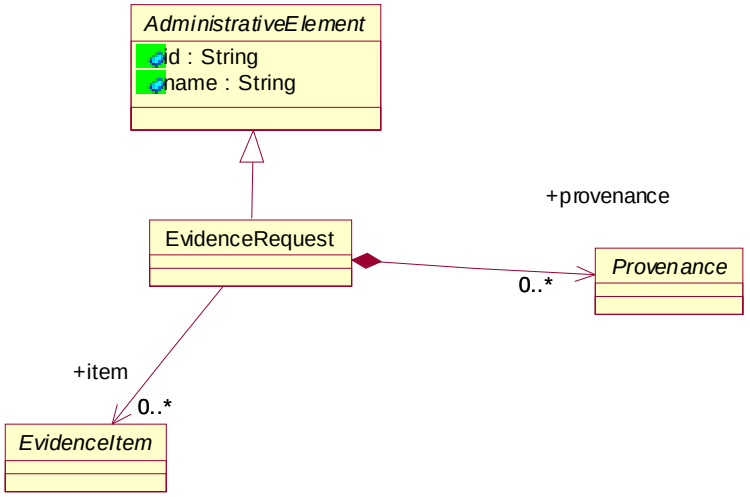


Figure 15.5 - Request Class Diagram

15.5.1 EvidenceRequest

EvidenceRequest represents a placeholder for an EvidenceItem to be collected during the evidence collection project.

Superclass

AdministrativeElement

Associations

- provenance:Provenance[0..*]
  Provenance properties of the request.
- item:EvidenceItem[0..*]
  Evidence items that satisfy the request.

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 collection project.

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’]

Domain 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: Domain Object corresponds to things in the domain of the evidence collection project

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

Domain 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 Domain Assertion involves Domain Objects and corresponds to state of affairs in the domain of the evidence collection project

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

Concept type: claim

Reference schema: stmt **of** Domain 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) [‘DomainClaim’]

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 represents segments of the fact model related to the situation for which the body of evidence is collected

Description: A formal assertion is a proposition that describes a state of affairs for which the evidence is collected. This proposition uses the vocabulary that is imported from the semantic community that is involved in the subject field within which the evidence is collected. Formal assertions for evidence collection represent (alleged) 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 collection project 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* Originator

Definition:

General concept: Provenance

Concept type: FactType

Evidence Element *is approved by* Originator

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 Originators

Originator

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

Originator *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: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">
  <containsArgumentElement xsi:type="ARM:Claim" xmi:id="1" identifier="C1" description="" content="C/S logic is fault free"/>
  <containsArgumentElement xsi:type="ARM:ArgumentReasoning" xmi:id="2" identifier="RC1.1" content="Argument by omission of all identified software hazards" describes="5 6"/>
  <containsArgumentElement xsi:type="ARM:ArgumentReasoning" xmi:id="3" identifier="RC1.2" content="Argument by satisfaction of all C/S safety requirements" describes="7 8 9"/>
  <containsArgumentElement xsi:type="ARM:InformationElement" xmi:id="4" identifier="IRC1.1" description="Identified software hazards"/>
  <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"/>
  <containsArgumentElement xsi:type="ARM:Claim" xmi:id="6" identifier="C1.2" description="" content="Unintended closing of press can only occur as a result of component failure"/>
  <containsArgumentElement xsi:type="ARM:Claim" xmi:id="7" identifier="C2.1" content="Press controls being 'jammed on' will cause press to halt"/>
  <containsArgumentElement xsi:type="ARM:Claim" xmi:id="8" identifier="C2.2" content="Release of controls prior to press passing physical PoNR will cause press operation to abort"/>
  <containsArgumentElement xsi:type="ARM:Claim" xmi:id="9" identifier="C2.3" description="" content="C/S fails safe (halts on) and annunciates (by sounding Klaxon) all component failures"/>
  <containsArgumentElement xsi:type="ARM:InformationElement" xmi:id="10" identifier="S1.1" content="Fault tree analysis cutsets for event 'Hand trapped in press due to command error'"/>
  <containsArgumentElement xsi:type="ARM:InformationElement" xmi:id="11" identifier="S1.2" content="Hazard directed test results"/>
  <containsArgumentElement xsi:type="ARM:EvidenceAssertion" xmi:id="12" identifier="C2.1.1" content="Failure 1 of PLC state machine includes BUTTON_IN remaining true"/>
  <containsArgumentElement xsi:type="ARM:EvidenceAssertion" xmi:id="13" identifier="C2.2.1" content="Abort transition of PLC state machine includes BUTTON_IN going false"/>
  <containsArgumentElement xsi:type="ARM:InformationElement" xmi:id="14" identifier="S2.1" description="" content="black box testing"/>
  <containsArgumentElement xsi:type="ARM:InformationElement" xmi:id="15" identifier="S2.2.1" content="C/S state machine"/>
  <containsAssertedRelationship xsi:type="ARM:AssertedInference" xmi:id="16" identifier="C1.1.1" description="" target="5" source="1"/>
  <containsAssertedRelationship xsi:type="ARM:AssertedInference" xmi:id="17" identifier="C1.1.2" target="6" source="1"/>
  <containsAssertedRelationship xsi:type="ARM:AssertedInference" xmi:id="18" identifier="C1.2.1" target="7" source="1"/>
  <containsAssertedRelationship xsi:type="ARM:AssertedInference" xmi:id="19" identifier="C1.2.2" target="8" source="1"/>
  <containsAssertedRelationship xsi:type="ARM:AssertedInference" xmi:id="20" identifier="C1.2.3" target="9" source="1"/>
  <containsAssertedRelationship xsi:type="ARM:AssertedContext" xmi:id="21" identifier="CIRC1.1" target="4" source="2"/>
  <containsAssertedRelationship xsi:type="ARM:AssertedEvidence" xmi:id="22" identifier="S1.1" target="10" source="5 6"/>
  <containsAssertedRelationship xsi:type="ARM:AssertedEvidence" xmi:id="23" identifier="S1.2" target="11" source="5 6"/>
  <containsAssertedRelationship xsi:type="ARM:AssertedEvidence" xmi:id="24" identifier="SC2.1" target="14" source="7"/>
  <containsAssertedRelationship xsi:type="ARM:AssertedEvidence" xmi:id="25" identifier="SC2.1.1" target="15" source="12"/>
  <containsAssertedRelationship xsi:type="ARM:AssertedEvidence" xmi:id="26" identifier="SC2.2.1" target="15" source="13"/>
  <containsAssertedRelationship xsi:type="ARM:AssertedInference" xmi:id="27" identifier="DI C2.1" target="12" source="7"/>
  <containsAssertedRelationship xsi:type="ARM:AssertedInference" xmi:id="28" identifier="DI C2.2" target="13" source="8"/>
</ARM:Argument>
```

B.2 Bluetooth Security Case

```
<?xml version="1.0" encoding="ASCII"?>
<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" identifier="BSC11">
  <containsArgumentElement xsi:type="ARM:Claim" identifier="Bluetooth secure" content="A bluetooth enabled network provides adequate
security"/>
  <containsArgumentElement xsi:type="ARM:Claim" identifier="Availability" content="A bluetooth enabled network is adequately available [1]
Section 1 para 3"/>
  <containsArgumentElement xsi:type="ARM:Claim" identifier="Access" description="" content="A bluetooth enabled network provides
adequate control for access to services and data [1] Section 1 para 3"/>
  <containsArgumentElement xsi:type="ARM:Claim" identifier="Confidentiality" content="A bluetooth enabled network provides adequate levels
of confidentiality [1] Setion 1 para 3"/>
  <containsArgumentElement xsi:type="ARM:Claim" identifier="Integrity" content="A bluetooth enabled network provides adequate levels of
integrity [1] Section 1 para 3"/>
  <containsArgumentElement xsi:type="ARM:InformationElement" identifier="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'"/>
  <containsArgumentElement xsi:type="ARM:InformationElement" identifier="References" content="[1] Bluetooth security white paper 19/4/02"/
>
  <containsArgumentElement xsi:type="ARM:InformationElement" identifier="Definition: Availability" content="The system is capable of
providing requested services to authorised users, in an acceptable/defined time"/>
  <containsArgumentElement xsi:type="ARM:InformationElement" identifier="Definition: Access" content="Only users permitted by the defined
security policy have access to services and data"/>
  <containsArgumentElement xsi:type="ARM:InformationElement" identifier="Define: Confidentiality" content="Unauthorised persons cannot
intercept and understand information to which they are not entitled"/>
  <containsArgumentElement xsi:type="ARM:InformationElement" identifier="Define: Integrity" description="" content="Services and data are
provided to authorised users as intended and without corruption"/>
  <containsArgumentElement xsi:type="ARM:ArgumentReasoning" identifier="Argue over vulnerabilities" description="" content="Argue for
each security requirement identified in the security white paper" describes="AI1"/>
  <containsAssertedRelationship xsi:type="ARM:AssertedContext" identifier="AC1" target="References" source="Bluetooth secure"/>
  <containsAssertedRelationship xsi:type="ARM:AssertedContext" identifier="AC2" target="Context: security policy and scenario for use"
source="Bluetooth secure"/>
  <containsAssertedRelationship xsi:type="ARM:AssertedInference" identifier="AI1" target="Integrity Confidentiality Access Availability"
source="Bluetooth secure"/>
</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.

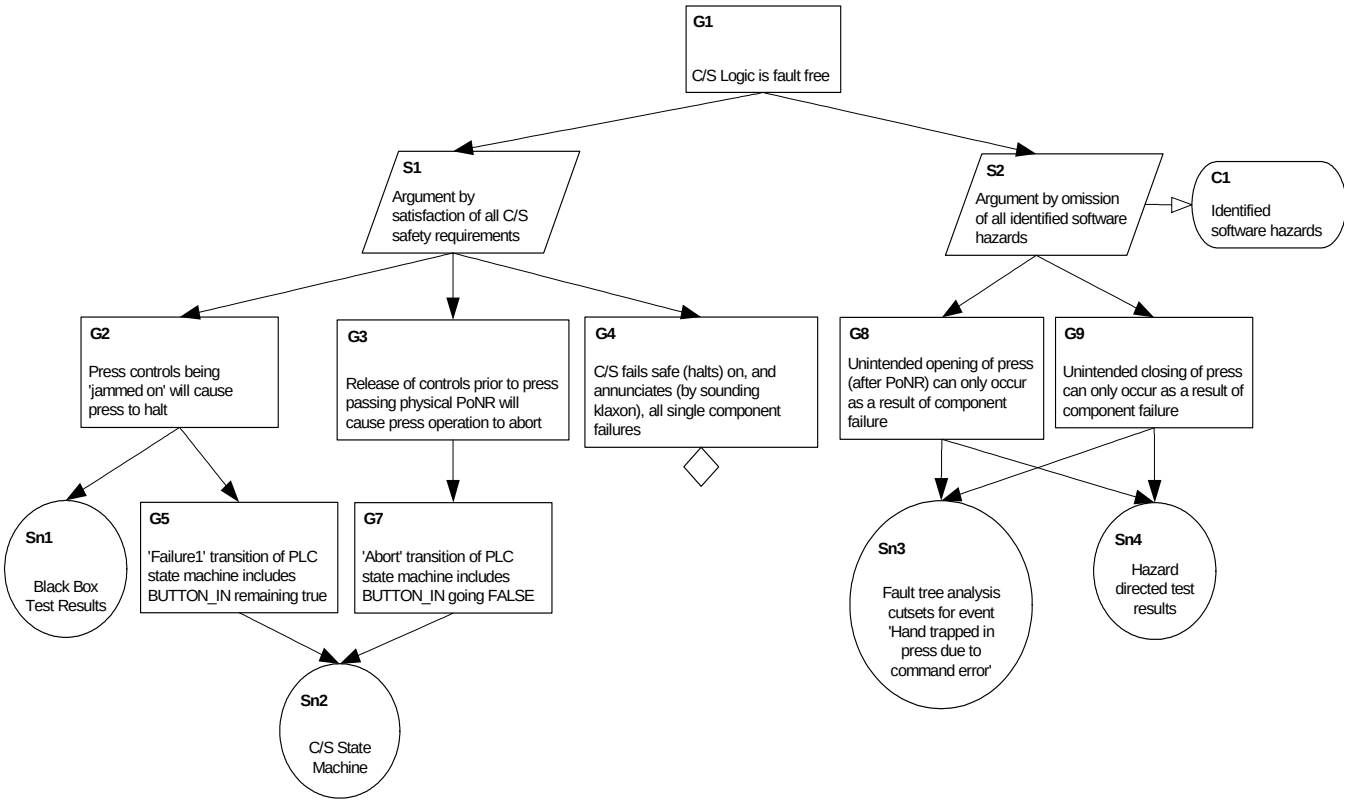


Figure B.1 - Industrial Press Safety argument (§8.3.1)

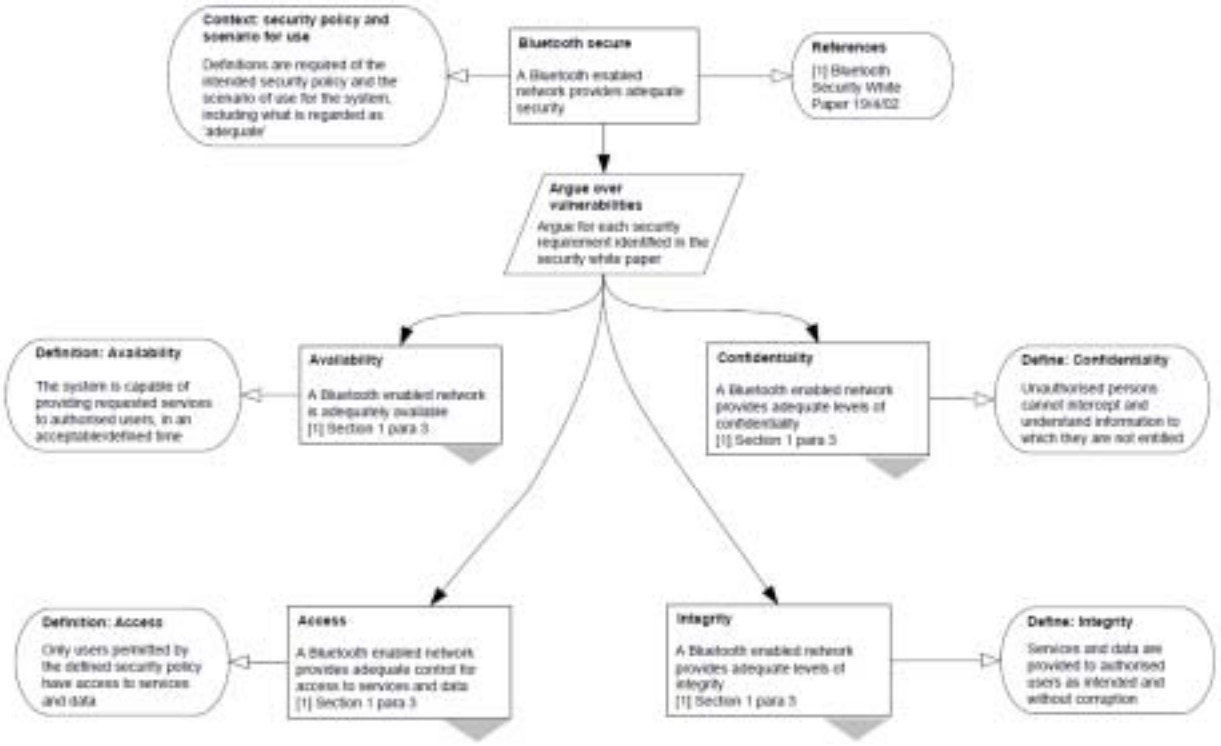


Figure B.2 - GSN Bluetooth Security Case (§8.3.2)

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.

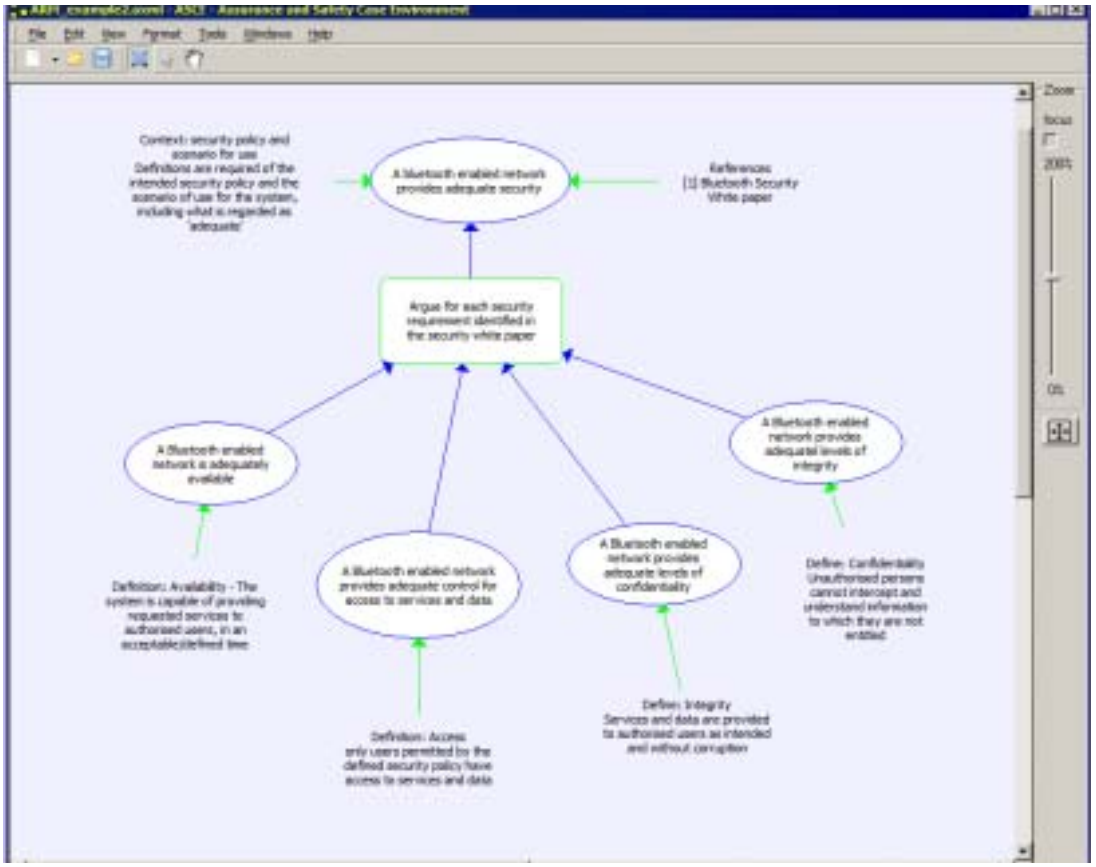


Figure B.3 - CAE of Bluetooth example - showing contextual information as visual nodes

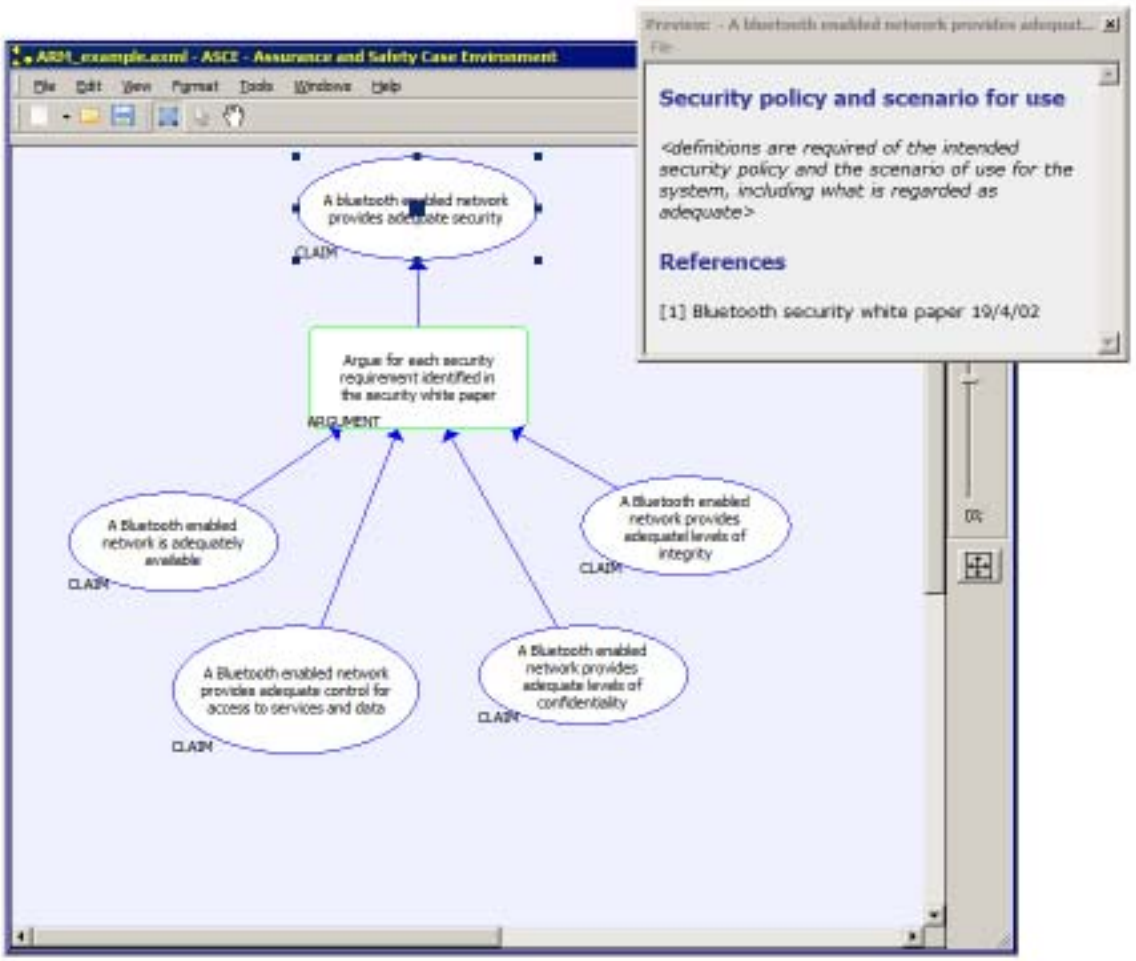


Figure B.4 - CAE representation of the Bluetooth example where contextual information held as rich text (top claim is selected)