ECSEL Research and Innovation actions (RIA)

AMASS

Architecture-driven, Multi-concern and Seamless Assurance and Certification of Cyber-Physical Systems

Prototype for seamless interoperability (a) D5.4

Work Package:	WP5 Seamless Interoperability
Dissemination level:	PU = Public
Status:	Final
Date:	30 March 2017
Responsible partner:	Luis M. Alonso (TRC)
Contact information:	luis.alonso@reusecompany.com
Document reference:	AMASS_D5.4_WP5_TRC_V1.0

This deliverable is part of a project that has received funding from the ECSEL JU under grant agreement No 692474. This Joint Undertaking receives support from the European Union's Horizon 2020 research and innovation programme and from Spain, Czech Republic, Germany, Sweden, Italy, United Kingdom and France.

Contributors

Names	Organisation
Luis M. Alonso, Borja López	The REUSE Company
Jose Luis de la Vara, Jose María Álvarez	Universidad Carlos III de Madrid
Ángel López	Tecnalia Research & Innovation

Reviewers

Names	Organisation
Frank Badstuebner (peer reviewer)	Infineon
Eugenio Parra (peer reviewer)	Universidad Carlos III de Madrid
Barbara Gallina (TC-member review)	Mälardalen University
Cristina Martínez (Quality Manager)	Tecnalia Research & Innovation

TABLE OF CONTENTS

List of Figures

Abbreviations and Definitions

API	Application Programming Interface
ARTA	AMASS Reference Tool Architecture
CACM	Common Assurance and Certification Metamodel
CDO	Connected Data Objects
CCL	Common Certification Language
CPS	Cyber-Physical Systems
ECSEL	Electronic Components and Systems for European Leadership
EEF	Extended Editing Framework
EMF	Eclipse Model Framework
OPENCOSS	Open Platform for EvolutioNary Certification of Safety-critical Systems
OSLC	Open Services for Lifecycle Collaboration
SACM	Structured Assurance Case Metamodel
SafeCer	Safety Certification of Software-Intensive Systems with Reusable Components
SVN	Apache Subversion
TRL	Technology Readiness Level
V&V	Verification and Validation
WP	Work Package

Executive Summary

The document is AMASS deliverable D5.4 - Prototype for seamless interoperability (a). It is the output of the task T5.3 Implementation for Seamless Interoperability and is based on the results from tasks T5.1 Consolidation of Current Approaches for Seamless Interoperability and T5.2 Conceptual Approach for Seamless Interoperability.

Task T5.3 develops a tooling framework to implement prototype support for seamless interoperability in CPS assurance and certification. T5.3 is being carried out iteratively, in close connection with the conceptual tasks (T5.2 and Tx.2 in the other technical WPs), and with validation results from the implementation being used to guide further refinement of the conceptual approach. The implementation is closely guided by the requirements of the case studies, which are used to evaluate the prototype.

The first prototype iteration releases the basic building blocks (Prototype Core) as a consolidation/integration of previous projects for:

- Access Management
- Data Management
- Evidence Management

More concretely, the developed tools in the first prototype support the following use cases:

- Characterise artefact
- Link artefact with external tool
- Specify artefact lifecycle
- Evaluation artefact
- Conduct impact analysis of artefact change
- Specify process-related information for artefacts

This document presents in detail the pieces of functionality implemented in the AMASS Tool Platform for the areas above, their software architecture, the technology used, and source code references.

D5.4 relates to other implementation-related AMASS deliverables:

- Installable AMASS Tool Platform for the first prototype
- User manuals and installation instructions
- Source code description

In addition, D5.4 related to the following AMASS deliverables:

- D2.1 (Business cases and high-level requirements) includes the requirements that have been implemented in D5.4.
- D2.2 (AMASS reference architecture (a)) presents the abstract architecture based on which D5.4 has been created.
- D2.6 (Integrated AMASS platform (a)) reports the results from validating the implementation described in D5.4.
- D5.1 (Baseline requirements for seamless interoperability) reviews the main background on seamless interoperability for AMASS and proposes a way forward. D5.4 corresponds to the initial realisation of this way forward.
- D5.5 (Prototype for seamless interoperability (b)) and D5.6 (Prototype for seamless interoperability (c)) will describe the second and third version, respectively, of the seamless interoperability support in the AMASS Tool Platform.

1. Introduction

The AMASS approach focuses on the development and consolidation of an open and holistic assurance and certification framework for CPS, which constitutes the evolution of the OPENCOSS [12] and SafeCer [16] approaches towards an architecture-driven, multi-concern assurance, reuse-oriented, and seamlessly interoperable tool platform.

The expected tangible AMASS results are:

a) The **AMASS Reference Tool Architecture**, which will extend the OPENCOSS and SafeCer conceptual, modelling and methodological frameworks for architecture-driven and multi-concern assurance, as well as for further cross-domain and intra-domain reuse capabilities and seamless interoperability mechanisms (based on OSLC specifications [14]).

b) The **AMASS Open Tool Platform**, which will correspond to a collaborative tool environment supporting CPS assurance and certification. This platform represents a concrete implementation of the AMASS Reference Tool Architecture, with a capability for evolution and adaptation, which will be released as an open technological solution by the AMASS project. AMASS openness is based on both standard OSLC APIs with external tools (e.g. engineering tools including V&V tools) and on open-source release of the AMASS building blocks.

c) The **Open AMASS Community**, which will manage the project outcomes, for maintenance, evolution and industrialization. The Open Community will be supported by a governance board, and by rules, policies, and quality models. This includes support for AMASS base tools (tool infrastructure for database and access management, among others) and extension tools (enriching AMASS functionality). As Eclipse Foundation is part of the AMASS consortium, the Polarsys/Eclipse community (www.polarsys.org) is a strong candidate to host AMASS Open Tool Platform.

To achieve the AMASS results, as depicted in Figure 1, the multiple challenges and corresponding scientific and technical project objectives are addressed by different work-packages.

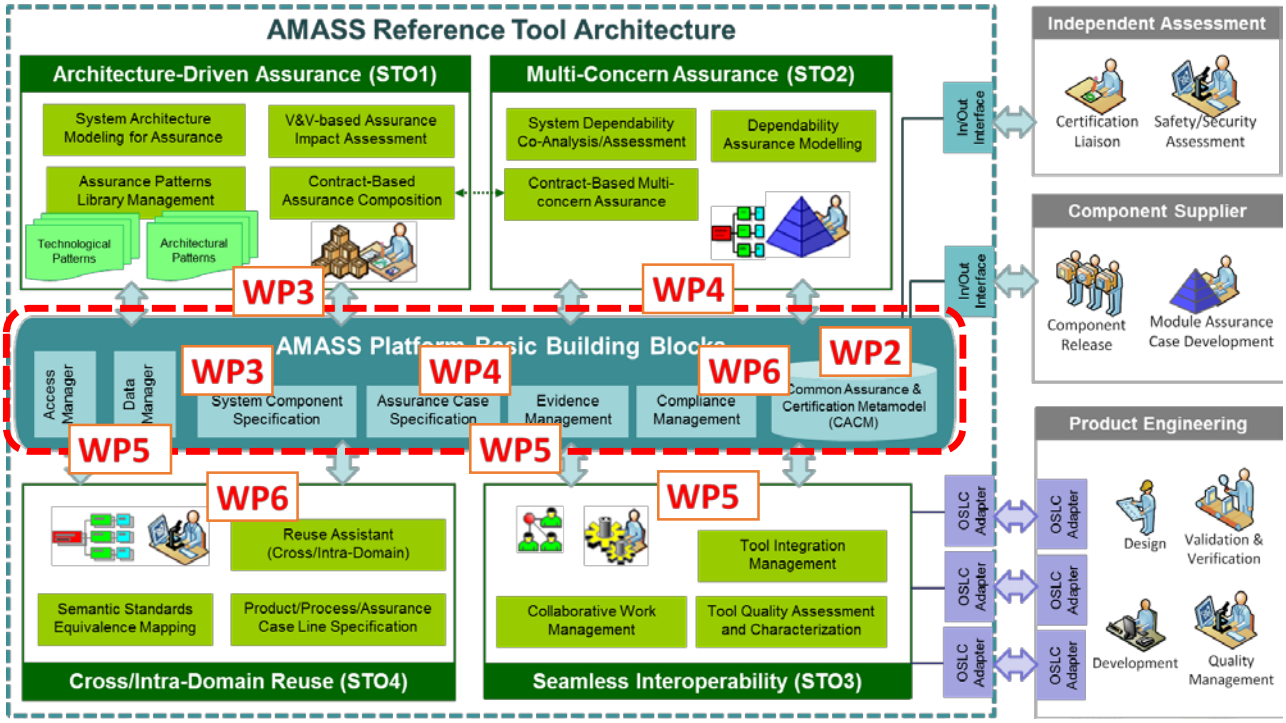

Figure 1. AMASS Building blocks

Since AMASS targets high-risk objectives, the AMASS Consortium decided to follow an incremental approach by developing rapid and early prototypes. The benefits of following a prototyping approach are:

- Better assessment of ideas by initially focusing on a few aspects of the solution.
- Ability to change critical decisions based on practical and industrial feedback (case studies).

AMASS has planned three prototype iterations:

1. During the **first prototyping** iteration (Prototype Core), the AMASS Platform Basic Building Blocks (see [2]), will be aligned, merged and consolidated at TRL4[1].
2. During the **second prototyping** iteration (Prototype P1), the AMASS-specific Building Blocks will be developed and benchmarked at TRL4; this comprises the blue basic building blocks as well as the green building blocks. Regarding seamless interoperability, in this second prototype, the specific building blocks will provide advanced functionalities regarding tool integration, collaborative work, and tool quality characterisation and assessment.
3. Finally, at the **third prototyping** iteration (Prototype P2), all AMASS building blocks will be integrated in a comprehensive toolset operating at TRL5. Functionalities specific for seamless interoperability developed for the second prototype will be enhanced and integrated with functionalities from other technical work packages.

Each of these iterations has the following three prototyping dimensions:

- ***Conceptual/research development***: development of solutions from a conceptual perspective.
- ***Tool development***: development of tools implementing conceptual solutions.
- ***Case study development***: development of industrial case studies using the tool-supported solutions. The application of the building blocks in case studies for this first prototype are described in D1.1 [1].

As part of the Prototype Core, WP5 is responsible for consolidating the previous works on specification of evidence characteristics, handling of evidence evolution, and specification of evidence-related information (e.g. process information) in order to design and implement the basic building block called "**Evidence Management**" (Figure 1). In addition, WP5 has been responsible for the implementation of the **"Access Manager"** and **"Data Manager"** blocks. Nonetheless, the functionality of these latter blocks is used not only in WP5, but in all the WPs, e.g. for data storage and access (of system components, of assurance cases, of standards' representations, etc.).

This deliverable reports the **tool development** results of the "Evidence Management", "Access Manager", and "Data Manager" basic building blocks. It presents in detail the design of the functionality implemented in the AMASS Tool Platform, the building blocks software architecture, the technology used, and source code references. The design is based on the investigated state of the art and state of practice approaches presented in D5.1 [5] and D2.2[2] [2]. Their gaps were identified and analysed to determine a way forward for seamless interoperability, enabling the formulation of requirements to achieve the interoperability vision of AMASS. This vision covers tool integration, collaborative work, and tool quality assessment and characterisation.

The rest of the deliverable presents the requirements implemented (Section 2) and describes the implementation performed (Section 3).

[1] In the context of AMASS, the EU H2020 definition of TRL is used, see http://ec.europa.eu/research/participants/data/ref/h2020/other/wp/2016_2017/annexes/h2020-wp1617-annex-g-trl_en.pdf

[2] D2.2 is an initial, non-public description of the ARTA. The deliverable that presents the final version of the ARTA will be public.

2. Implemented Functionality

This section presents the scope of the implementation work reported in this deliverable and the implemented requirements.

2.1 Scope

The scope for the prototype for seamless interoperability is the provision of edition tools for specification and management of evidence-related assurance information, mostly artefact information. The prototype also provides support for access management and data management. The main scope is highlighted with a red circle and red rectangles in Figure 2, which shows the general functional overview of the AMASS Tool Platform.

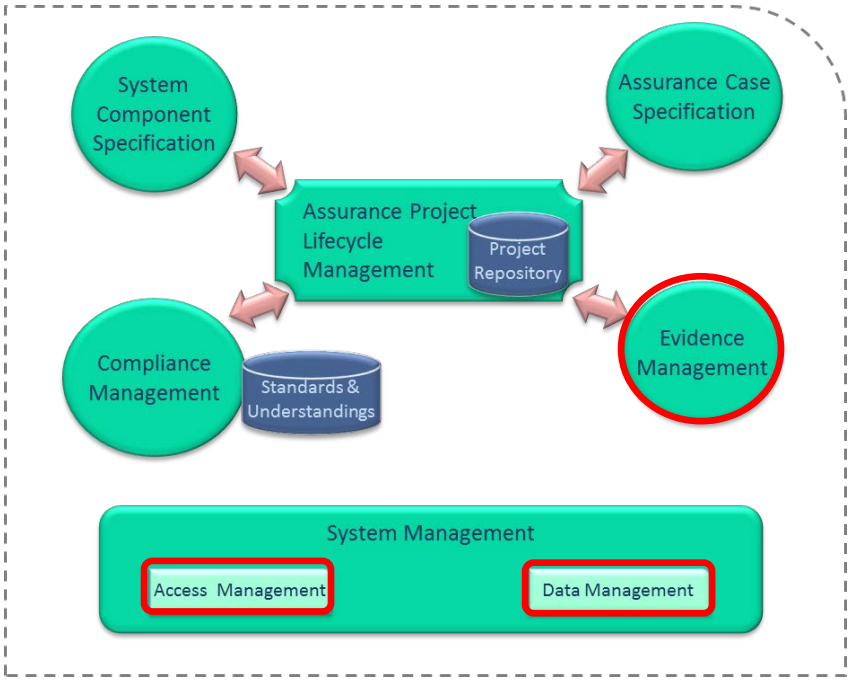

Figure 2. Functional decomposition for the AMASS platform

The Evidence Management block handles the full lifecycle of evidence artefacts and evidence chains. This includes evidence traceability management and impact analysis. The Access Management block is an infrastructure functional module that includes generic functionality for security, permissions, and profiles, whereas the Data Management block is an infrastructure functional module that includes generic functionality for data storage, visualization, and reporting.

The next section presents the use cases that the above building blocks support in the scope of WP5.

2.2 Implemented Requirements

The implemented requirements correspond to six use cases specified in D2.2 [2]. The following subsections include a short description of how the implementation performed supports each use case. The implementation is based on form-based information editors.

2.2.1 Characterise Artefact

For artefact characterization, the AMASS Tool Platform allows a user to create artefact models and add artefact definitions to the model via a tree-view based editor (Figure 3). Artefacts can later be specified for the artefact definitions (Figure 4). For each artefact, a user can specify basic data such as the name, a description, version information, and the precedent version.

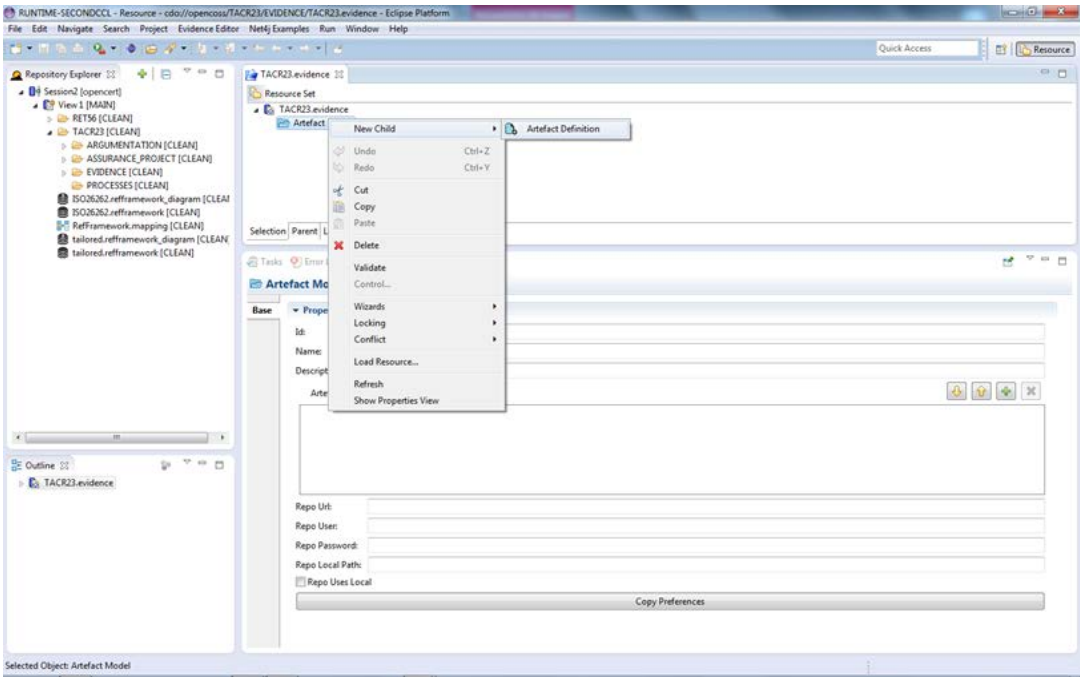

Figure 3. Artefact definition creation

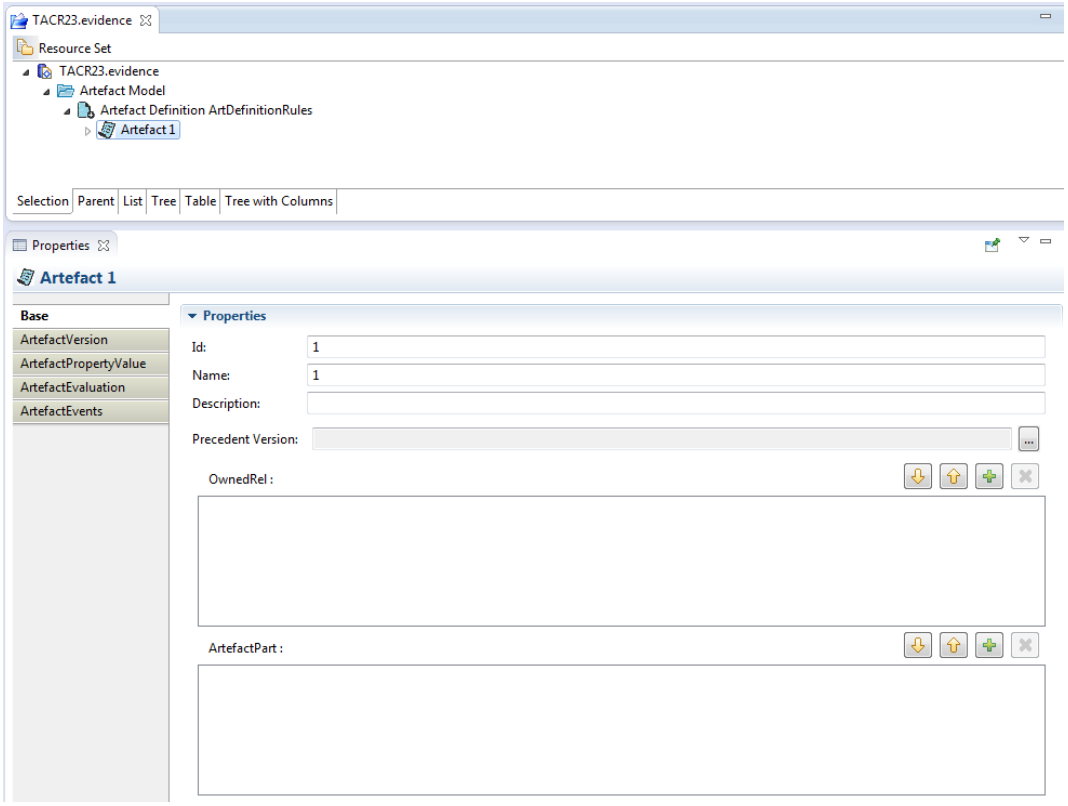

Figure 4. Artefact data specification

2.2.2 Link Artefact with External Tool

Artefacts can be linked to external tools in two main ways. First, a user can specify that the artefact repository for an assurance project corresponds to a SVN repository (Figure 5). Second, a resource can be added to an artefact (Figure 6) and, in its properties (Figure 7), a user can indicate the external location and format of the file that actually corresponds to the artefact.

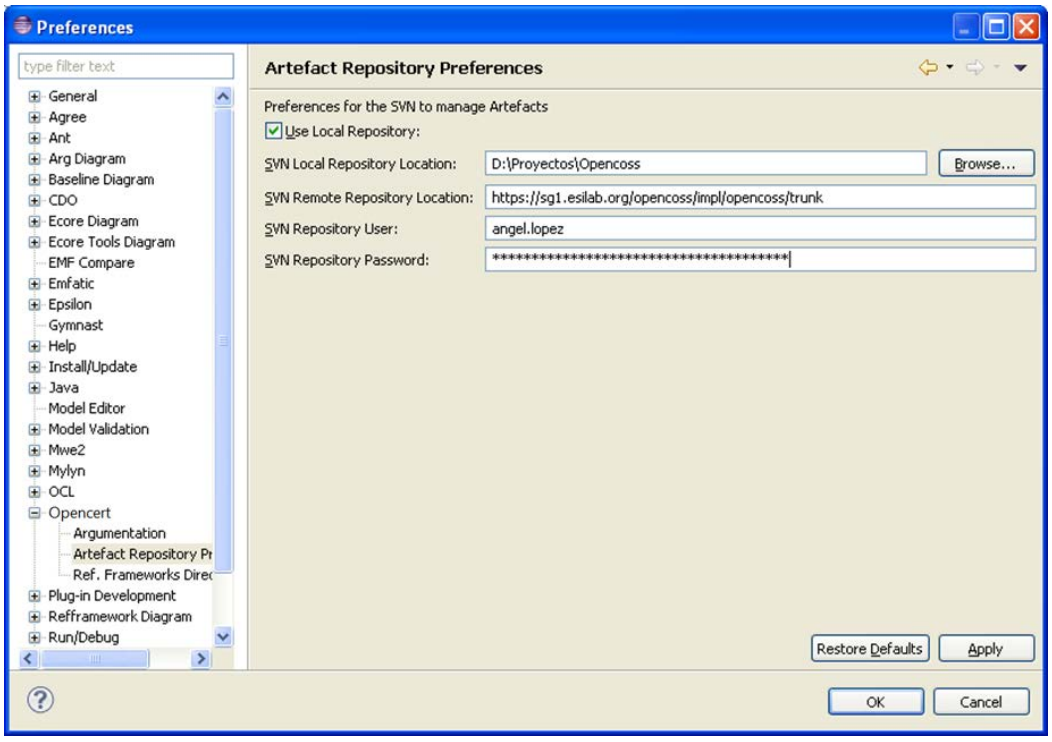

Figure 5. Use of SVN repository as artefact repository

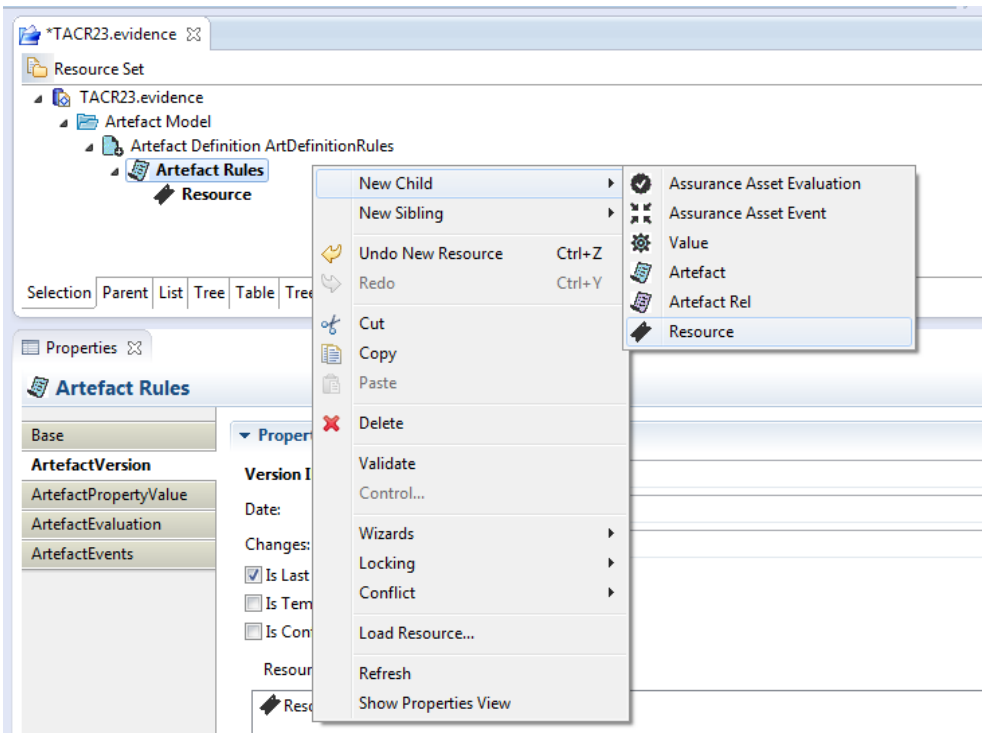

Figure 6. Resource specification for an artefact

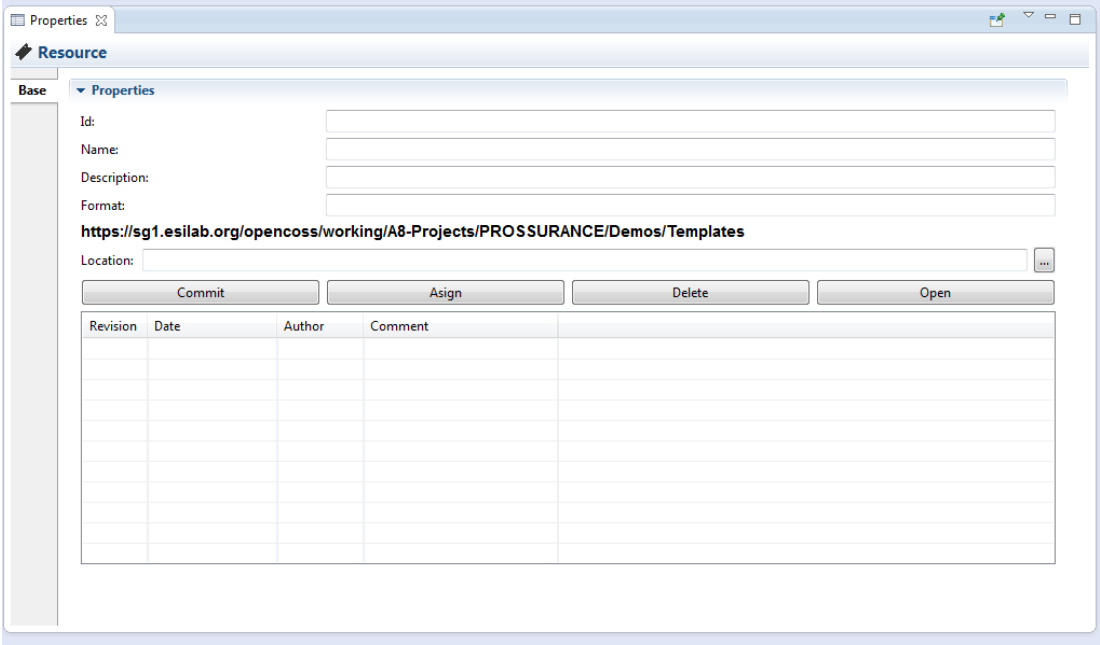

Figure 7. Resource properties

2.2.3 Specify Artefact Lifecycle

Once an artefact has been created, its lifecycle can be specified by adding events and specifying event data (Figure 8), such as the event type (creation, modification, evaluation, and revocation) and when the event happened.

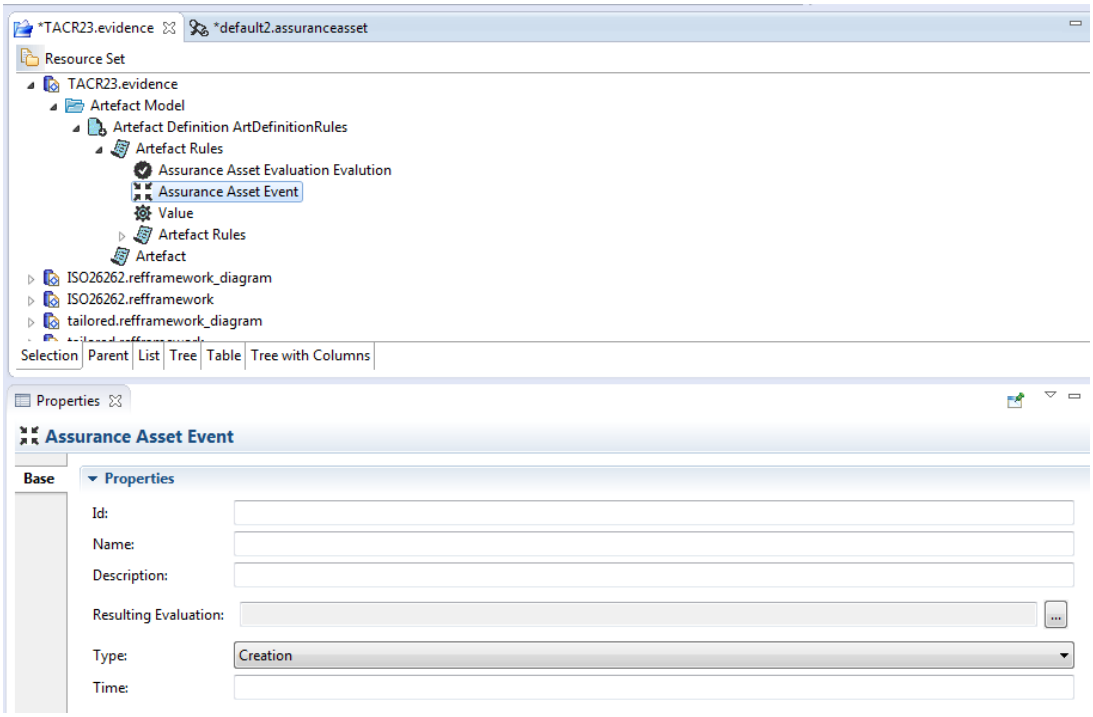

Figure 8. Artefact event properties

2.2.4 Evaluate Artefact

A user can add evaluations to artefacts. The users can also specify the evaluation criterion, the criterion description, the evaluation result, and its rationale, among other properties (Figure 9).

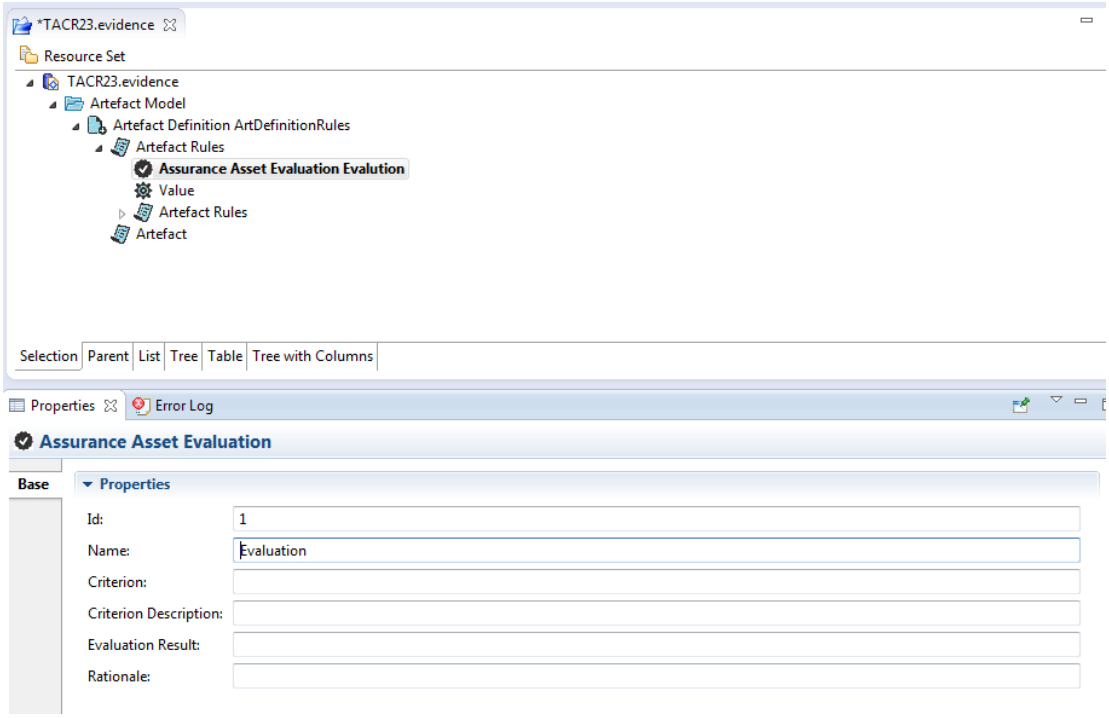

Figure 9. Artefact evaluation properties

2.2.5 Conduct Impact Analysis of Artefact Change

When changes are made to artefacts and these changes result in modification events (Figure 10), the users can determine the impact of such changes in other artefacts and accept it or refuse it (Figure 11).

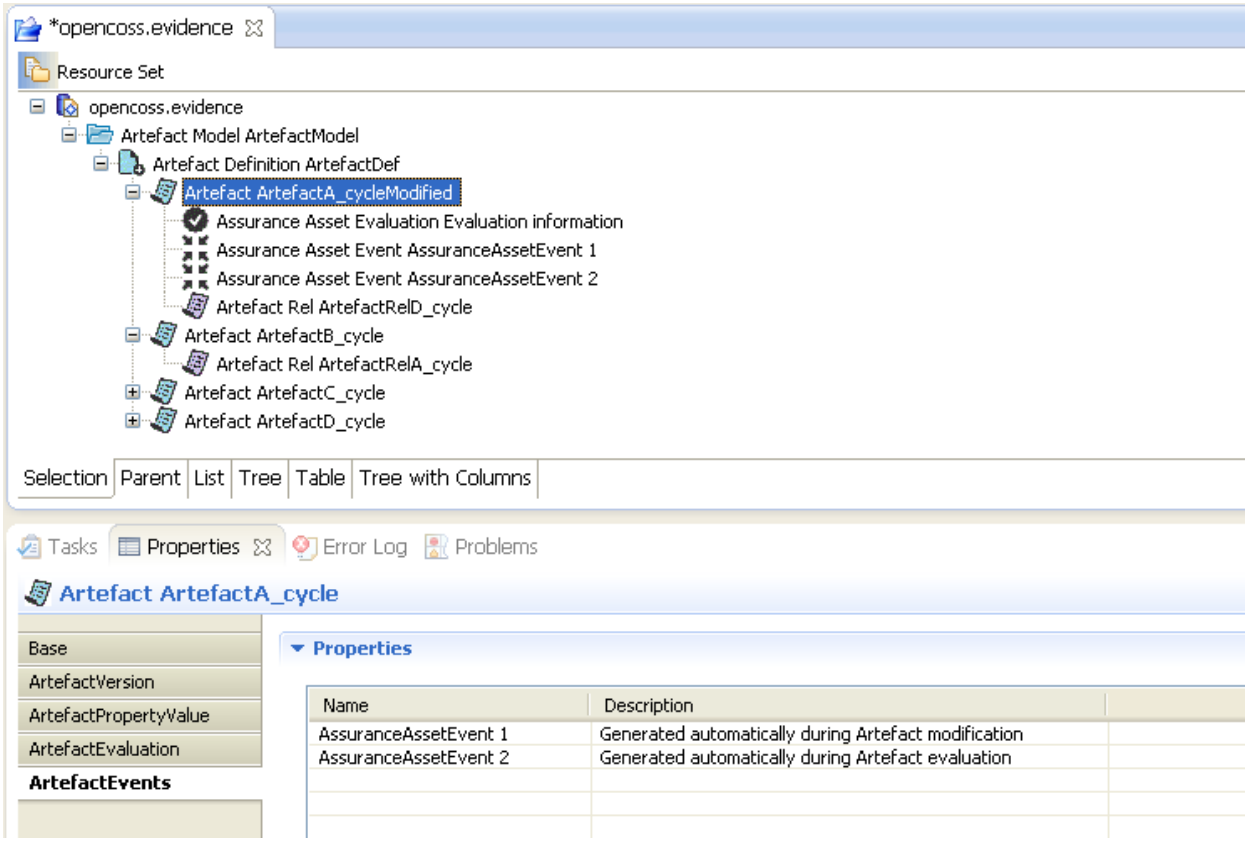

Figure 10. Modification event of an artefact

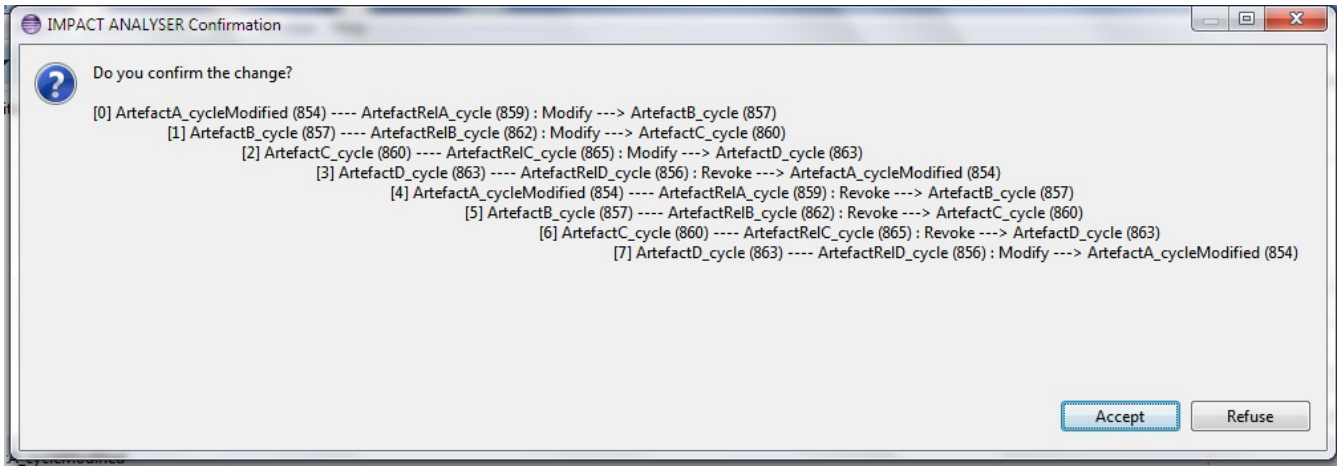

Figure 11. Impact analysis information

2.2.6 Specify Process-Related Information for Artefacts

Process-related artefact information is specified by means of process models (Figure 12). These models can contain information about activities, participants, persons, tools, organizations, and techniques involved in the processes of an assurance project. Artefacts can later be associated to these elements. For example, 'activity artefacts' is a set of activity data (Figure 13) with which the input and output artefact of an activity can be specified.

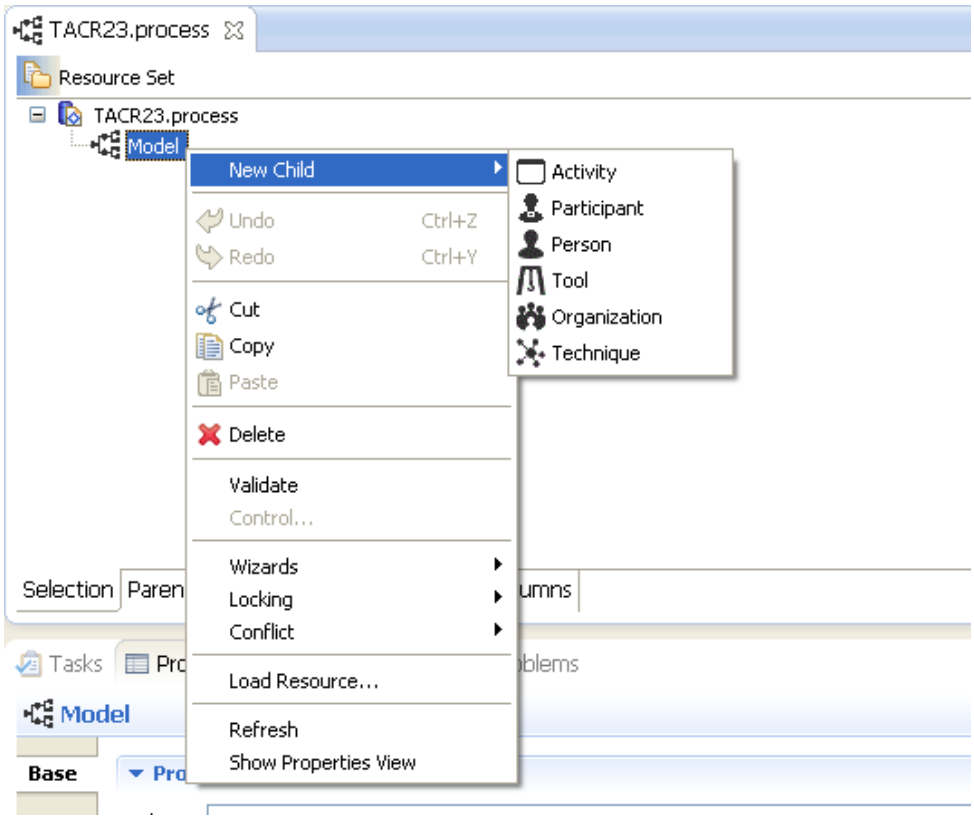

Figure 12. Process model

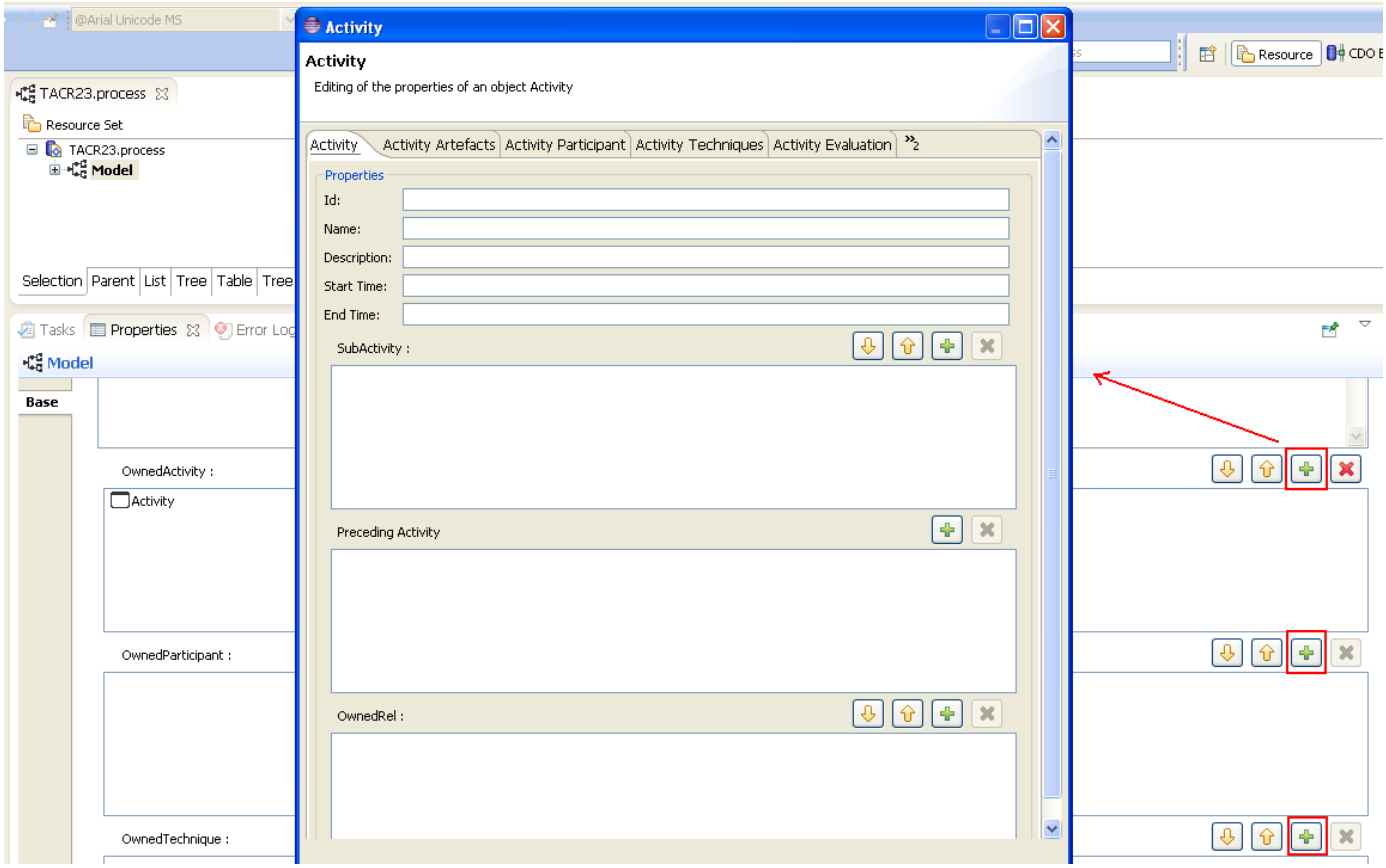

Figure 13. Activity data

2.3 Installation and User Manuals

The steps necessary to install the first prototype are exhaustively described in the AMASS User Manual [7] (currently under elaboration for all the AMASS building blocks), thus they are not repeated in this deliverable. In the user manual of the first AMASS tool prototype implementation, the users can find the installation instructions, the tool environment description, and the functionalities for the specification of evidence-related assurance project information: artefact repository preferences, artefact definitions, artefacts, artefact resources, artefact property values, artefact events, artefact evaluations, impact analysis, executed processes, and property models.

3. Implementation Description

This section presents the modules that have been implemented, the underlying metamodel, and the source code created.

3.1 Implemented Modules

The Evidence Management building block (Figure 14) is an Eclipse-Based editor for artefact and executed-process information of an assurance project. It contains plugins for edition of artefact models and of process models. The purpose of the Evidence Characterization Editor module is to provide services for evidence storage (determination, specification, and structuring of evidence), whereas the Traceability Editor module includes services for traceability-related aspects, both for specifying relationships between evidence artefacts and for impact analysis of evidence artefact changes.

Regarding the Access Manager and the Data Manager, they have been realised in a single module (System Management; Figure 15). This module provides data access and data storage services to all the application and application infrastructure modules of the AMASS Tool Platform. These modules include those developed in WP3, WP4, and WP6, thus documented in D3.4 [3], D4.4 [4], and D6.4 [6].

The main technology for implementation of the Evidence Management modules and of the System Management module has been OpenCert [15]. The Evidence Management corresponds to editors mostly generated with the EMF [11] and EEF [10] Eclipse technologies, in addition to the implementation of some tailored functionality, e.g. for integration with SVN and for impact analysis. For the System Management module, CDO [9] is the main base technology.

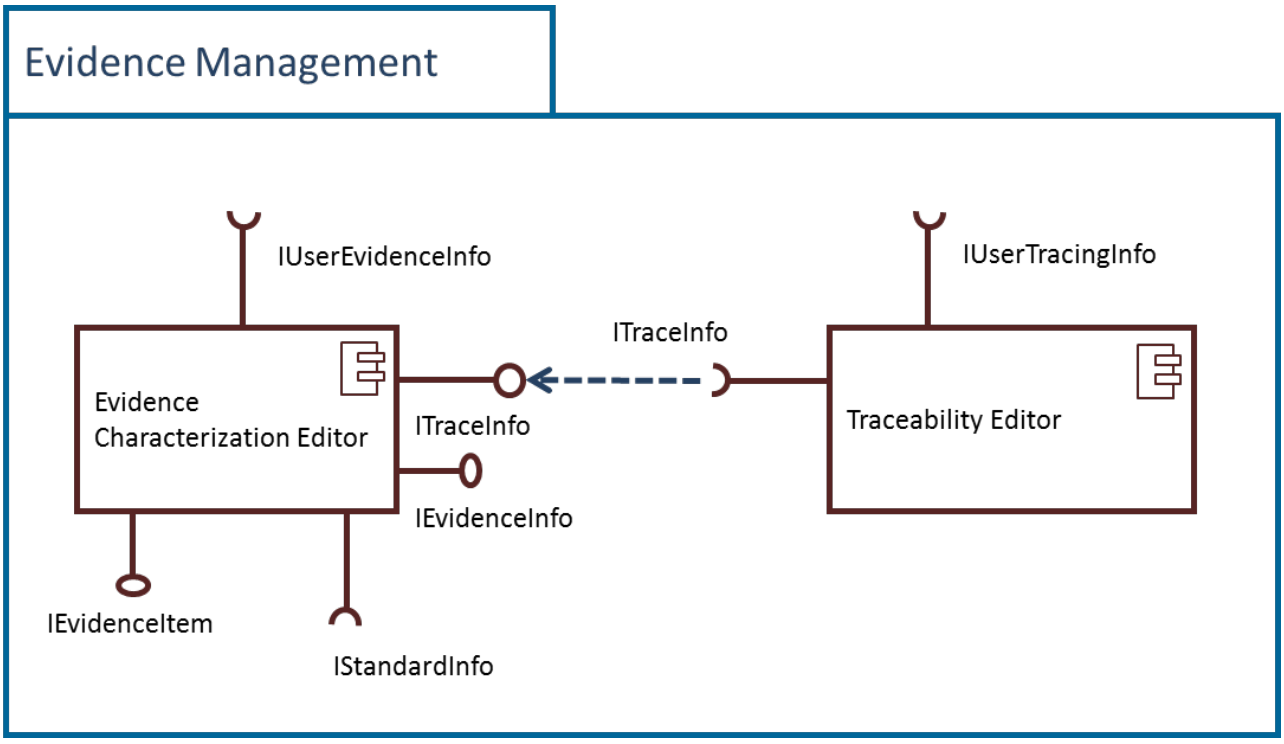

Figure 14. Evidence management modules

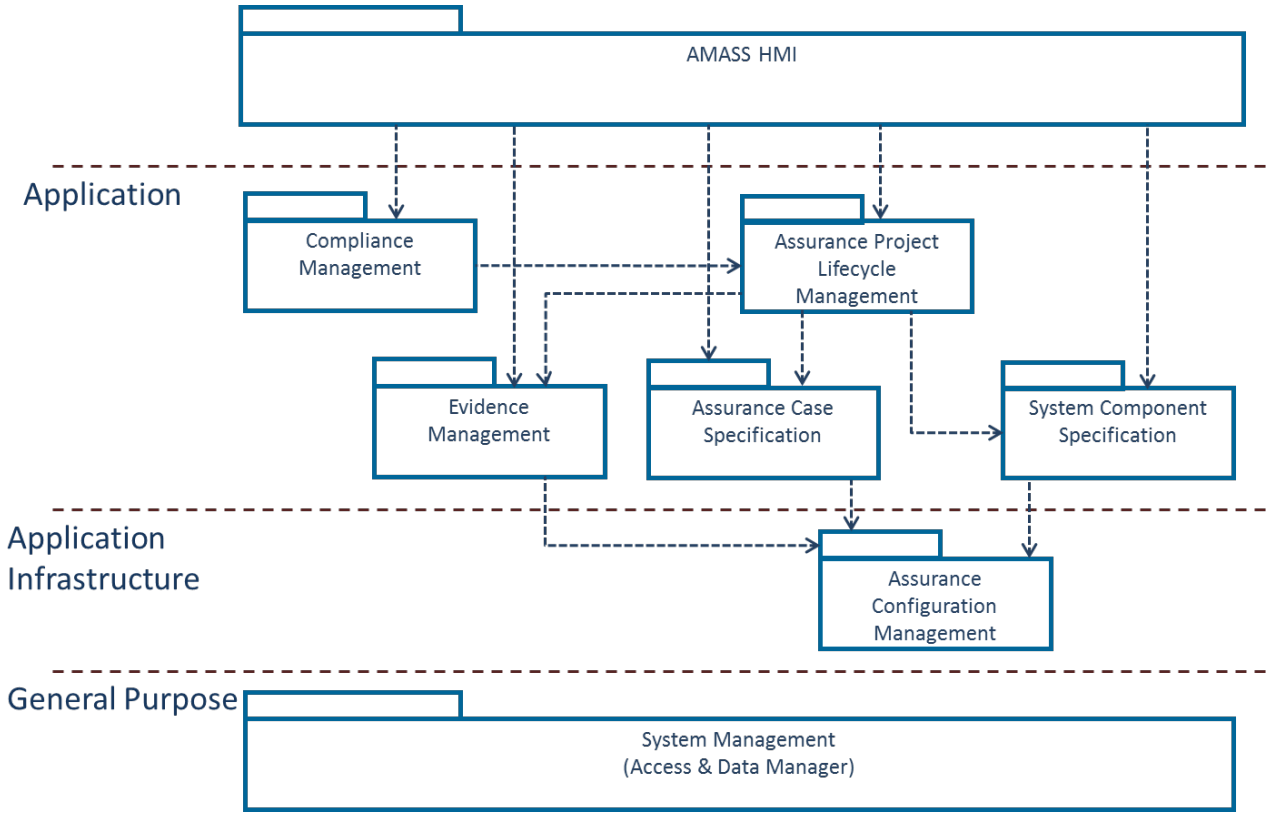

Figure 15. System management module

3.2 Implemented Metamodel

AMASS D2.2 [2] presents the CACM, including evidence management metamodels. These metamodels correspond to the envisioned, conceptual data structure necessary in AMASS purpose to provide the reuse-oriented holistic approach for architecture-driven assurance, multi-concern assurance, and seamless interoperability. However, the metamodel implemented for the Evidence Management modules does not exactly correspond to the CACM, but to the metamodel implemented in OpenCert. This situation will be re-analysed for future AMASS prototypes.

Such metamodel is the CCL created in the OPENCOSS project. The CCL can be regarded as compliant with the CACM because it supports all the evidence information specification needs represented in the CACM. However, the specification of information can be a bit different. For example, traceability information is not specified in the CCL based on a specific metamodel, but this information type is embedded in the CCL artefact metamodel.

Figure 16 shows an excerpt of the CCL to specify evidence information. Further information about the CCL can be found in [13].

3.3 Source Code Description

The source code of the first AMASS prototype can be found in the source code SVN repository [8]. The code for the first prototype's evidence management and system management modules are stored together with the other basic building blocks in the repository under "tag" to distinguish the state of the code at the time of the integrated release.

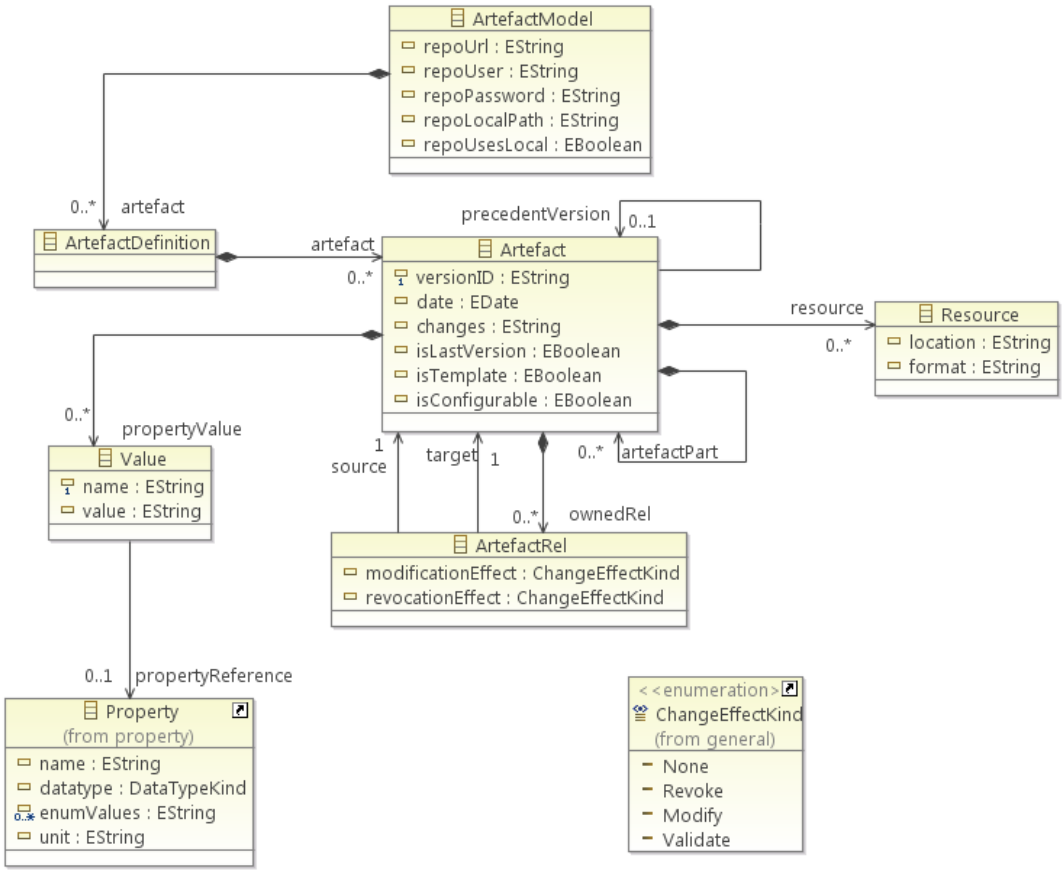

Figure 16. Excerpt of artefact information in the CCL

The necessary plugins for Evidence Management and for System Management (Figure 17) are:

- **org.eclipse.opencert.evm.evidspes**
  In this plugin, the evidence metamodel is defined and stored, and the Java implementation classes for this model are generated.
- **org.eclipse.opencert.evm.evidspes.edit**
  This plugin contains a provider to display evidence models in a user interface.
- **org.eclipse.opencert.evm.evidspes.editor**
  This plugin provides the user interface to view instances of the model using several common viewers, and to add, remove, cut, copy and paste model objects, or modify the objects in a standard property sheet.
- **org.eclipse.opencert.evm.evidspes.editor.dawn**
  This plugin is an extension of the previous one. It aims to communicate with the CDO Server to store the generated model.
- **org.eclipse.opencert.evm.evidspec.preferences**
  This plugin defines the default preferences for the communication with the SVN repository, thus it defines the type of repository (local or remote) and a user and password to connect with the remote repository.
- **org.eclipse.opencert.impactanalysis**
  This plugin contains the implementation of the change impact analysis module. This module is used by AMASS Tool Platform clients to call and execute change impact analysis.

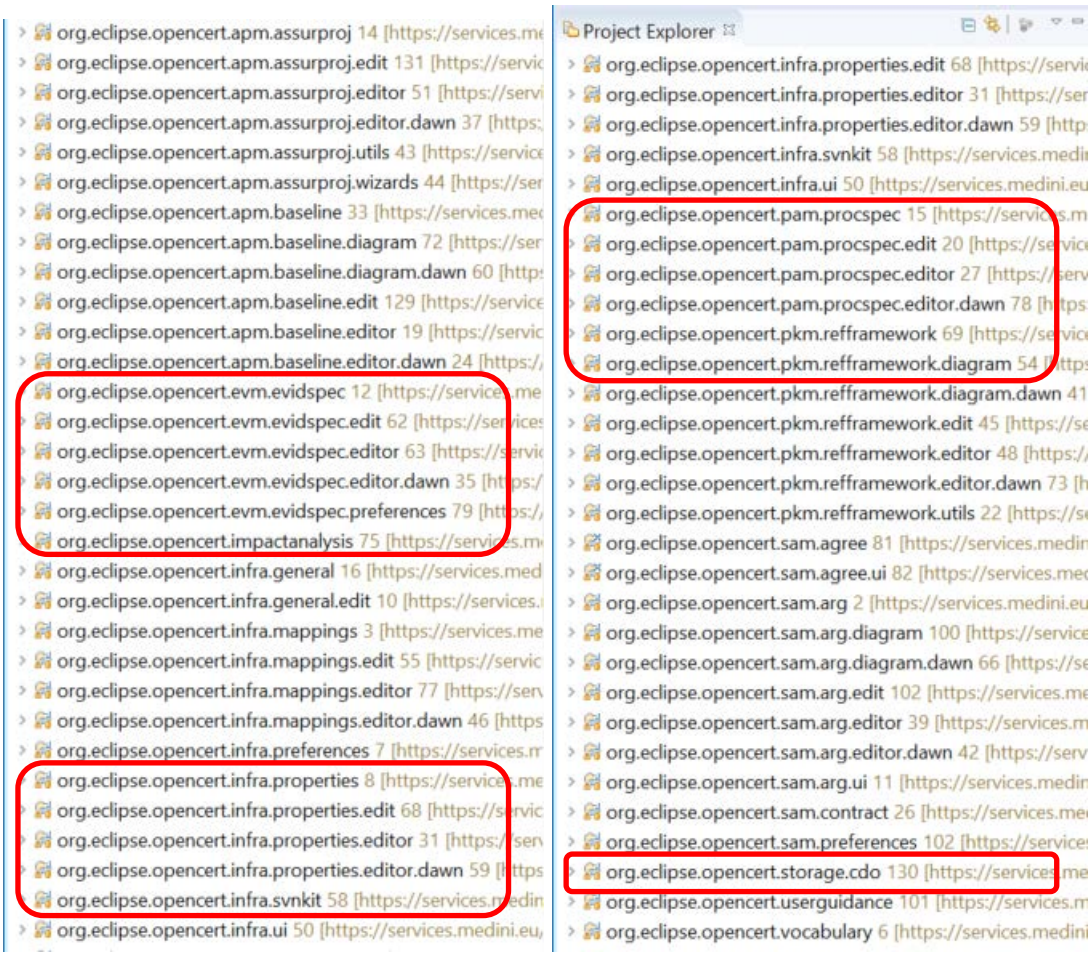

Figure 17. Evidence management and System management plug-ins

- **org.eclipse.opencert.infra.properties**
  This plugin contains the definition of the Property metamodel, and the Java implementation classes for this model.
- **org.eclipse.opencert.infra.properties.edit**
  As the edit plugin for evidence, this plugin contains a provider to display the model in a user interface.
- **org.eclipse.opencert.infra.properties.editor**
  As the edit plugin for evidence, this plugin is an editor to create and modify instances of the model.
- **org.eclipse.opencert.infra.svnkit**
  In this plugin, the functionalities necessary for the communication with the repository SVN are defined, to export and import artefacts.
- **org.eclipse.opencert.pam.procspec**
  In this plugin, the process execution metamodel is defined and stored, and the Java implementation classes for this model are generated.
- **org.eclipse.opencert.pam.procspec.edit**
  This plugin contains a provider to display process execution models in a user interface.
- **org.eclipse.opencert.pam.procspec.editor**
  This plugin provides the user interface to view instances of the model using several common viewers, and to add, remove, cut, copy and paste model objects, or modify the objects in a standard property sheet.
- **org.eclipse.opencert.pam.procspec.editor.dawn**

This plugin is an extension of the previous one. It aims to communicate with the CDO Server to store the generated model.

- **org.eclipse.opencert.storage.cdo**
  This plugin contains classes fur using the CDO server in the AMASS Tool Platform. This server provides a common storage for all AMASS Tool Platform clients and a server. It accesses PostgreSQL database as its data backend. In addition to common storage implementation, this package contains utility classes used when accessing the CDO server by its clients.

References

[1] AMASS project: D1.1 - Case studies description and business impact. 2016. http://amass-ecsel.eu/sites/amass.drupal.pulsartecnalia.com/files/documents/D1.1_Case-studies-description-and-business-impact_AMASS_final.pdf

[2] AMASS project: D2.2 - AMASS reference architecture (a). 2016.

[3] AMASS project: D3.4 - Prototype for architecture-driven assurance (a). 2016. http://amass-ecsel.eu/sites/amass.drupal.pulsartecnalia.com/files/documents/D3.4_Prototype%20for%20architecture-driven%20assurance%20%28a%29_AMASS_final.pdf

[4] AMASS project: D4.4 - Prototype for multiconcern assurance (a). 2017. http://amass-ecsel.eu/sites/amass.drupal.pulsartecnalia.com/files/documents/D4.4_Prototype-for-multiconcern-assurance-%28a%29_AMASS_final.pdf

[5] AMASS project: D5.1 - Baseline requirements for seamless interoperability. 2016. http://amass-ecsel.eu/sites/amass.drupal.pulsartecnalia.com/files/documents/D5.1_Baseline-and-Requirements-for-Seamless-Interoperability_AMASS_Final.pdf

[6] AMASS project: D6.4 - Prototype for cross/intra-domain reuse (a). 2017.

[7] AMASS project: Prototype Core User Manual, Version 0.1[3]. 2017. https://services.medini.eu/svn/AMASS_collab/WP-transversal/ImplementationTeam/PrototypeCore/AMASS_Prototype1_UserManual.docx

[8] AMASS project: Source code repository. 2017. https://services.medini.eu/svn/AMASS_source/ [4]

[9] Eclipse: CDO Model Repository. 2017. https://eclipse.org/cdo/

[10] Eclipse: EEF. 2016. https://eclipse.org/eef/#/

[11] Eclipse: EMF. 2017. https://eclipse.org/modeling/emf/

[12] OPENCOSS project. 2015. http://www.opencoss-project.eu/

[13] OPENCOSS project: D4.4 - Common Certification Language: Conceptual Model. 2015. http://www.opencoss-project.eu/sites/default/files/D4.4_v1.5_FINAL.pdf

[14] OSLC community. 2017. https://open-services.net/

[15] PolarSys: OpenCert project. 2017. https://www.polarsys.org/projects/polarsys.opencert

[16] SafeCer Project. 2015. http://safecer.eu/

[3] The current User Manual is a draft document; the final version of the manual will be integrated in D2.5 - AMASS User guidance and methodological framework (m31).

[4] The AMASS SVN code repository is open to AMASS partners with the same credentials as the SVN document repository. In case that people outside the project need access, please contact the AMASS Project Manager (huascar.espinoza@tecnalia.com)