SEVENTH FRAMEWORK PROGRAMME

Collaborative Large-scale Integrating Project

OPENCOSS

Open Platform for EvolutioNary Certification Of Safety-critical Systems

Evidence management service infrastructure: Methodological Guide D6.7

Work Package:	WP6 Evolutionary Evidential Chain
Dissemination level:	Public
Status:	Final
Date:	27 March 2015
Responsible partner:	TU/e, Mark van den Brand
Contact information:	m.g.j.v.d.Brand@tue.nl

Contributors

Names	Organisation
Mark van den Brand, Luna Yaping Luo, Martijn Klabbers	Eindhoven University of Technology
Giorgio Tagliaferri, Andrea Critelli, Vincenzo Manni	RINA Services S.p.A.
Jose Luis de la Vara, Sunil Nair, Carlo Ieva	Simula Research Laboratory
Rodolphe Arthaud	Inspearit

Document History

Version	Date	Remarks
V0.1	2014-07-10	ToC
V0.2	2014-08-05	Introduction to MG added, first version of text on concepts added, initial details on implementation added.
V0.3	2014-09-04	Contributions by RIN and SIM added. Feedback from INS.
V0.4	2014-09-15	Reviewed by TU/e
V0.5	2014-09-29	Reviewers' comments addressed and preparation for PB review
V1.0	2014-10-09	Deliverable finalisation after PB review
V2.0	2015-03-19	Adapted towards prototype 3
V2.0	2015-03-26	Processed remarks by Carlo Ieva

TABLE OF CONTENTS

List of Figures

List of Tables

Executive Summary

This methodological guide aims at describing the Evidence Management Service Infrastructure (EMSI). Via this infrastructure evidence management tools can be integrated and made available for safety certification by users from industry and assessors and auditors from certification bodies. The EMSI provides an infrastructure to couple and disclose tools for evidence management. The notion of evidence management is recaptured and the relation of the Common Certification Language (CCL) to evidence management is described. Understanding the CCL is relevant in order to integrate tools in an efficient way. The workflow for integrating (third party) tools is described. The architecture of the EMSI is described. Furthermore the basic functionality of the EMSI is presented. The interfaces are also described. Finally, the governance of the EMSI is described. The governance model focuses on the infrastructure and workflow related to error reporting.

1 Introduction to the Methodological Guide

Safety assurance and certification are amongst the most expensive and time-consuming tasks in the development of safety-critical embedded systems. European innovation and productivity in this market is curtailed by the lack of affordable (re)certification approaches. Major problems arise when evolutions to a system entail reconstruction of the entire body of certification arguments and evidence. Further, market trends strongly suggest that many future embedded systems will be comprised of heterogeneous, dynamic coalitions of components. As such, they will have to be built and assessed according to numerous standards and regulations. Current certification practices (e.g., traditional, monolithic and standards-based approaches) are likely to be prohibitively costly to apply to this kind of embedded systems (Rushby, 2007).

The OPENCOSS project aims to devise a common certification framework that spans different vertical markets for railway, avionics/aviation and automotive industries, and to establish an open-source safety certification infrastructure. The ultimate goal of the project is to bring about substantial reductions in recurring safety certification costs, and at the same time increase product safety through the introduction of more systematic certification practices. Both will boost innovation and system upgrades considerably.

This methodological guide describes the Evidence Management Service Infrastructure (EMSI). Via this infrastructure evidence management tools can be integrated and made available for safety certification by users from industry and assessors and auditors from certification bodies. The EMSI provides an infrastructure to couple and disclose tools for evidence management.

1.1 Context

This OPENCOSS Methodological Guide describes the methodology that is used in the OPENCOSS platform for certification. A central asset of OPENCOSS is the evidence management service infrastructure (EMSI). The EMSI has been developed by a number of partners of an EC-sponsored consortium consisting of certification bodies, transport industry, tool vendors and consultancy firms (see the list of OPENCOSS partners in http://www.opencoss-project.eu/node/6). Within OPENCOSS the goal of the EMSI is to specify and implement a management infrastructure for the safety certification evidential chain. By using this infrastructure it is possible to integrate evidence-related tools in a flexible way and provide the functionality in a web-based manner to developers and assessors. This Methodological Guide states how the EMSI can be extended with evidence-related tools; both tooling developed in OPENCOSS and third-party tools.

OPENCOSS and EMSI provide a software framework to support the manufacturers of safety components, the companies that use these components within their products and services, and the assessors of the same products and services within the Automotive, Railway, and Avionics industry domains in their evidence management. The support is focused on the management of evidence for (parts of) safety critical systems within or across the industry domains. Figure 1 below shows the architecture of EMSI for evidence management.

The following functionalities have been implemented as the result of software development done for the OPENCOSS platform prototype:

- Client-server infrastructure
- Central data storage used by both server and clients
- Evidence Gap Analysis web report
- Evidence items Change Impact Analysis
- Events mechanism framework in the server
- REST API serving evidence information
- Forms editors for evidence management
- Integration of the evidence editor with the Impact Analysis functionality

The implementation of the above functionality is described in detail in D6.6. The following diagram depicts the deployment and communication between the main implementation modules of the OPENCOSS Platform. A repository for EMF models and meta-models provides data persistency for the OPENCOSS tools. OPENCOSS platform client tools have been implemented as Eclipse plugins and are supposed to be installed on user machines. They have been implemented using EMF (Eclipse Modelling Framework) technology and auxiliary technologies like EEF (Extended Editing Framework) and GMF (Graphical Modelling Framework). The OPENCOSS clients provide editors for the CCL model data, and the evidence change impact analysis engine.

The OPENCOSS platform server has been implemented as a set of web applications using an Apache Tomcat server.

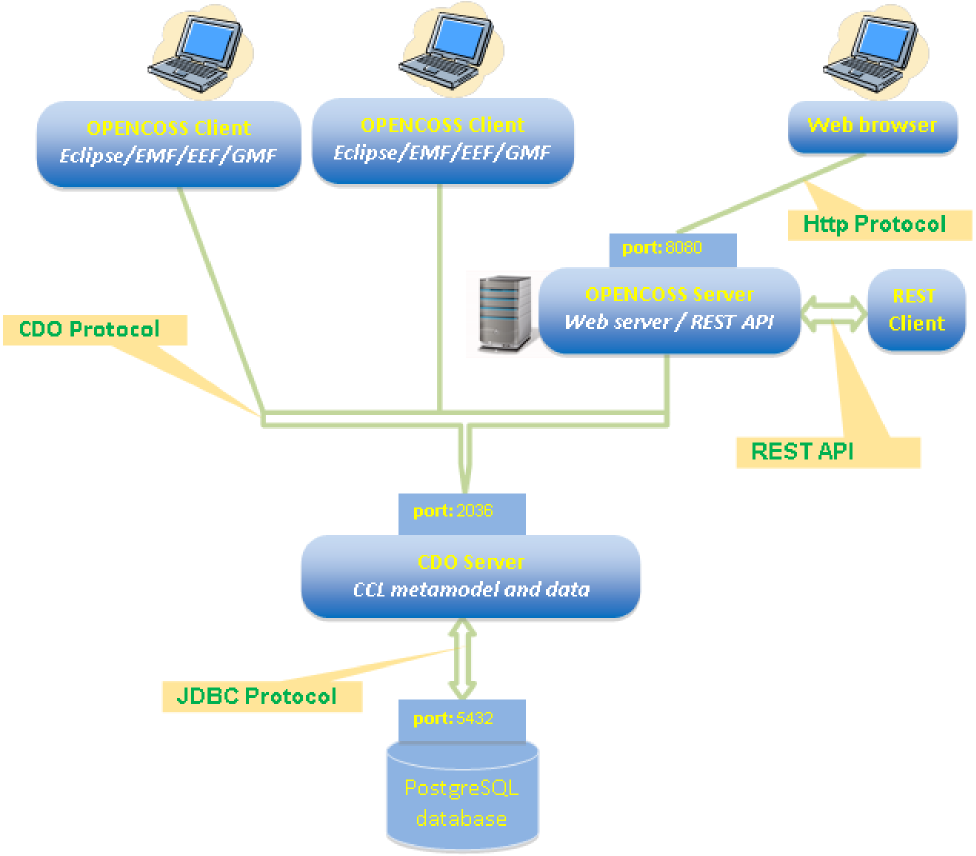


Figure 1. Architecture of OPENCOSS EMSI

1.2 Purpose of the Methodological Guide

The purpose of this Methodological Guide is to:

- Summarize the methodological concepts that are used in developing the EMSI.
- Give a set of rules to integrate tools with EMSI.
- Scope the EMSI to specific developers.
- Describe the technological choices made for the architecture.
- Introduce how the EMSI architecture is governed and maintained.

1.3 Scope of the Methodological Guide

This methodological guide aims at describing the Evidence Management Service Infrastructure (EMSI). Via this infrastructure evidence management tools can be integrated and data made available for safety certification by users form industry and assessors and auditors from certification bodies. This is an important distinction. In some safety-critical projects, tools (i.e. software) used in the development (and analysis) process themselves need to be certified/qualified (indeed, this would be an issue for the OPENCOSS tooling if it were ever to be used for real in the automotive or avionics domain (and possibly others)). Here, we are not talking about tool certification (that has not been addressed in the OPENCOSS project at all), but about the presentation of data. The methodological guide describes the architecture, the provided interfaces to connect (third-party) tools, and specifies the requirements for connecting (third-party) tools.

The methodological guide focuses on scenarios involving tool developers who need to integrate their evidence management tools. The users of the integrated evidence management tools should read the user manuals for the individual tools.

This methodological guide focuses on the concepts provided by and the design decisions for the service infrastructure. It is not a guide on how to use the integrated evidence management tools. In Figure 2 these user groups are positioned towards each other and the framework:

- Platform users within industry (Platform User), using the evidence management toolset to manage their evidence information;
- OPENCOSS governance (Platform developer), managing the EMSI and its toolset.
- Existing tools integraton with EMSI (Industrial developer), using the APIs to connect to the OPENCOSS server.

An important note is that EMSI enables to integrate itself with other external tools.

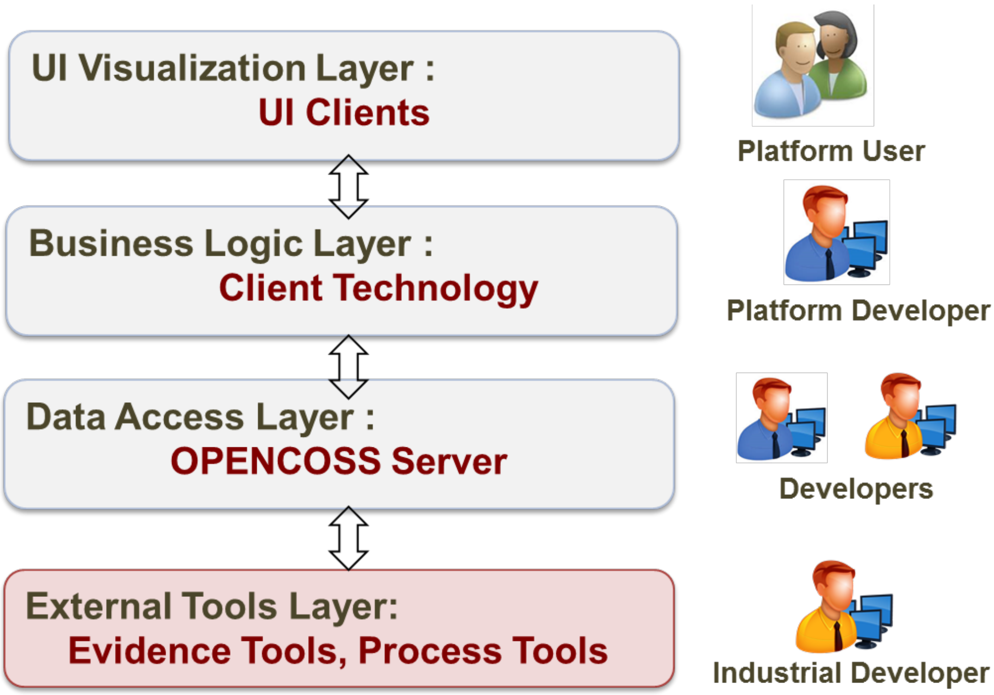


Figure 2 . User groups of EMSI (Grey layers: OPENCOSS platform; Red layers: Third party tools)

1.4 Structure of the Methodological Guide

Following the contextual discussion in Chapter 1, Chapter 2 describes the concepts of the evidence management approach developed by OPENCOSS. Chapter 3 describes the implementation of the evidence management service infrastructure and describes the usage of the evidence management service infrastructure from a developer’s point of view. Chapter 4 describes the governance of the evidence management service infrastructure. Chapter 5 presents the guidance for using the infrastructure. Finally, Chapter 6 presents our conclusions. A case study will be presented in the Appendix A.

2 Concepts of Evidence Management

The “evidence” and “artefact” concepts are described in deliverable D2.2 (and also in the CCL defintions in D4.4) as follows:

- *Evidence* consists of a collection of artefacts that provide evidentiary support to a set of claims in an argument. In other words, evidence is information, based on established facts or expert judgments, which are presented to show that the claim to which it relates is valid (i.e., true) in the context of the argument. Anything that supports the claim can be presented as evidence. Often, this information is demonstrating that a certain event or process took place. Various artefacts may be used as evidence, such as documents, expert testimony, test results, measurement results, records related to process, product, and people, etc.
- An *artefact* is a versioned document or data item, or a collection of these; a 'certification artefact' or 'assurance artefact' that indicates a document, data item, or collection required as part of the demonstration of assurance or compliance, either an evidence item, argument fragment or requirements document. Note that the term may be used at different levels of granularity - for example to refer to a single requirement, or an entire document.

In essence, information contained in an artefact can be used as evidence in an assurance project, and a relation exists between the two concepts. Nonetheless, an artefact or its content are not evidence per se, but can be used as and can become evidence if they are used to support some claim.

2.1 Background information on Evidence Management

EMSI should not be targeted at:

- Creating artefacts such as models, source code and V&V products (e.g., modelling of fault trees)
- Supporting practices specific only to a particular domain or standard (i.e., the platform must aim to be generic, so that it can be used in different domains and for different safety standards, allowing customization for a given standard to comply with)
- Requiring proprietary technologies for its use (e.g., DOORS)

A survey, see D6.1, was performed which revealed that much manual work is still performed in industry to check (1) completeness of the body of evidence and (2) to analyses evidence change impact. Therefore, the EMSI could significantly contribute to the improvement of the state of the practice in these aspects. Information about change impact should be recorded in the platform, and traceability between pieces of evidence should be shown by means of matrices. The benefits from using other ways to show traceability (e.g., by means of models) were studied in OPENCOSS. In addition, it was explicitly acknowledged the difficulty in dealing with and maintaining evidence traceability, in order to guarantee that traces are adequate and complete.

The role of the EMSI is to manage the evidence required in an assurance project, focusing on chains of evidence produced during the project, and also to increase the effectiveness and efficiency of evidence management. As shown below, evidence management involves different activities, such as evidence collection, combination, and evaluation.

2.2 Role of CCL in Evidence Management

The CCL is a common conceptual and notational framework for specifying certification assets. The CCL is used as a means to secure mutual agreement of meanings across a complex organisational or supply-chain structure in a given assurance project within a domain, and can also be employed in reuse scenarios to discuss abstract notions from different domains. This common conceptual framework for different safety standards aims to enable management of claims, evidences and arguments in a common format, sharing patterns of certification assessment and allowing cost-effective re-certification between different standards.

The CCL consists of two main elements (Figure 3): a set of metamodels and a propositional language. The first one provides the concepts from which models of safety standards and of the certification assets managed in an assurance project can be created. For evidence management, the concepts are provided by means of an evidence characterization metamodel (Figure 4). From which evidence characterization models are created. Therefore, the CCL defines what information can be and might have to be collected for the evidence items of an assurance project, as well as how to specify evidence traceability or evidence evaluations. The evidence characterization model of an assurance project might refer to elements of models of safety standards. With regard to the propositional language, it provides and groups the terminology used in safety standards and assurance projects.

In summary, evidence management in the OPENCOSS platform is performed according to the CCL, and the CCL should also fulfil evidence management-related requirements.

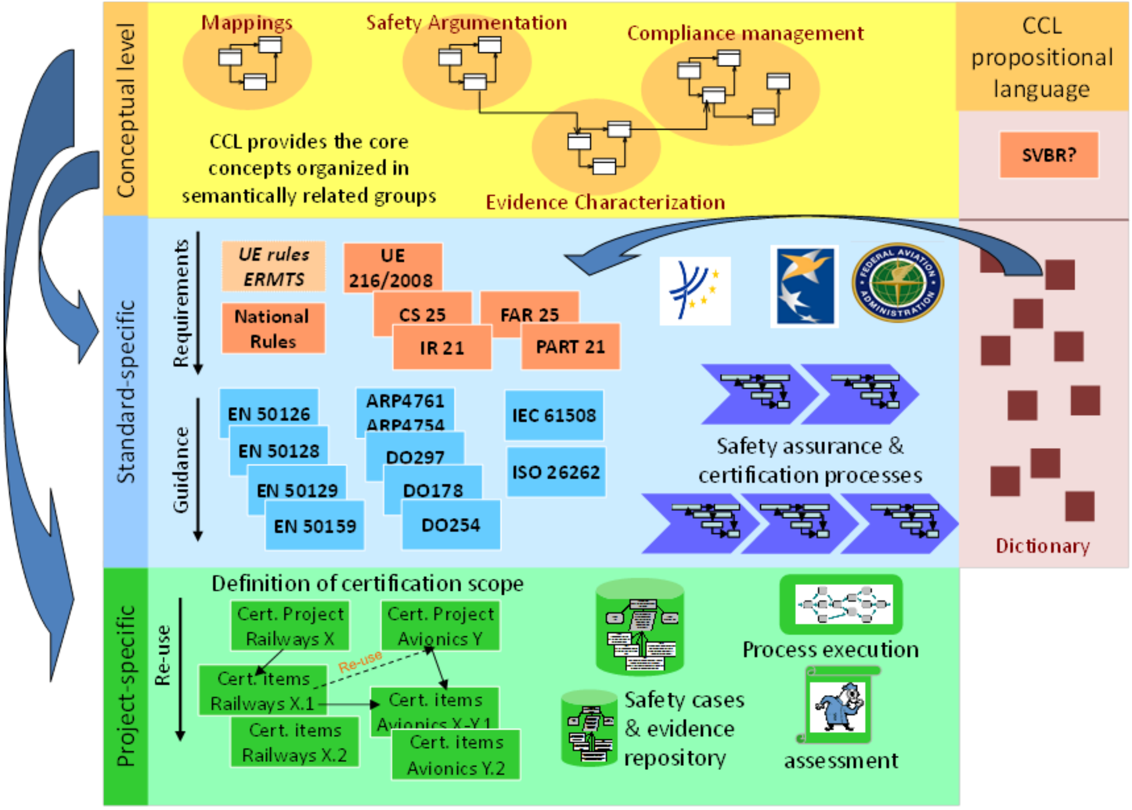


Figure 3. Overview of the CCL and its use

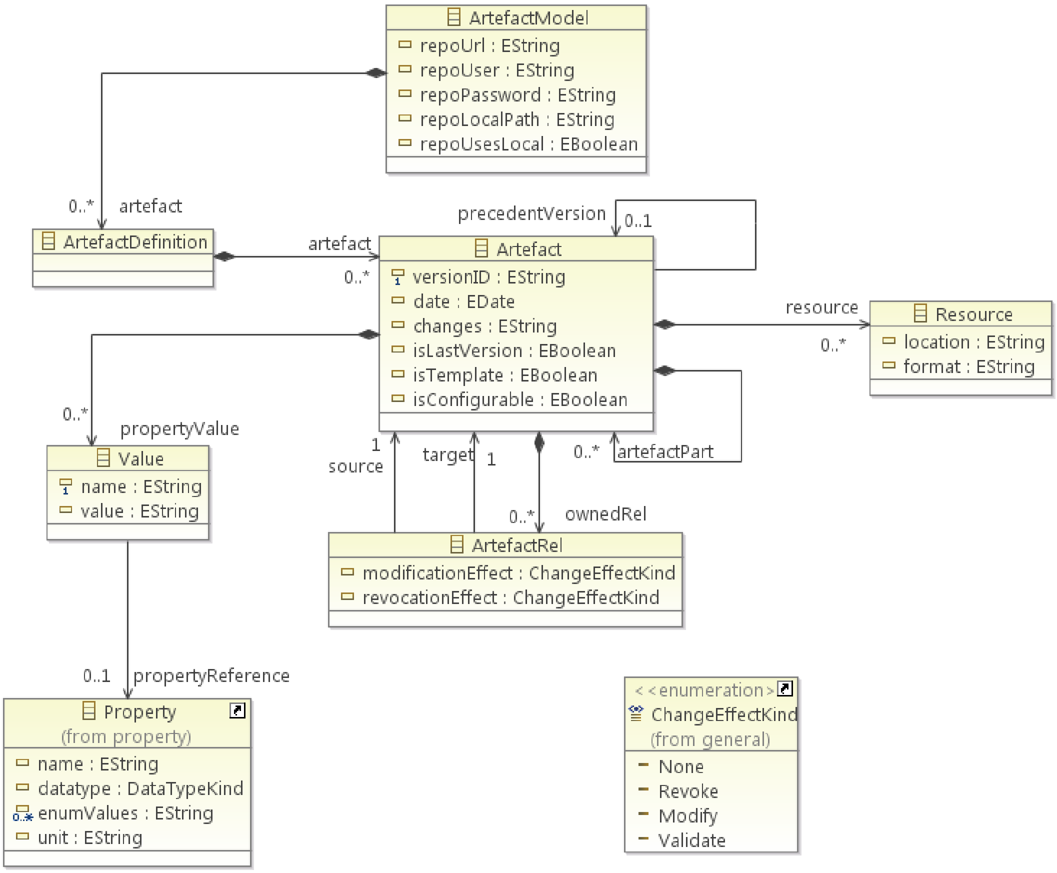


Figure 4. Fragment of CCL metamodel for evidence characterization in assurance projects

2.3 Specification of the Evidence Management Service Infrastructure

In D6.2, five main functional areas were defined for specifying component level requirements related to evidence management in the OPENCOSS tool platform:

- **Evidence storage**, concerned with the determination, specification, and structuring of the evidence items of an assurance project.
- **Evidence traceability**, concerned with the specification and adequate maintenance of traceability information about relationships between evidence items of an assurance project.
- **Evidence evaluation**, concerned with the assessment of the completeness and adequacy of the body of evidence of an assurance project, and of specific criteria defined for evaluation of individual evidence items.
- **Evidence change impact analysis**, concerned with the identification and analysis of possible effects resulting from changes in the body of evidence of an assurance project.
- **Integration with external tools**, concerned with the possibility of importing and exporting information from and to external tools, and information synchronization with them.

Examples of functionality that the OPENCOSS platform provides are:

- Integration with product engineering tools
- Evidence import/export
- Traceability between the information used as evidence
- Evidence dependency and change impact analysis

- Evidence and chains of evidence consistency checking
- Dashboards about the status of an assurance project and about its body of evidence

3 Implementation of the Evidence Management Service Infrastructure

3.1 Workflow for integration of tools

In order to integrate (third party) tools it is first necessary to understand the CCL evidence meta-model. Information on this can be found in the methodological guide on the CCL, see D4.7, and in the CLL definition documents, D4.4. Section 4 of D6.6 contains detailed instructions on the installation of the EMSI for software developers. The installation instructions for a (third party) tool have to be followed, this under the assumption that the EMSI was successfully installed. Finally, the tool has to be integrated, see Section 4 of D6.6. The main steps are:

1. Installation of OPENCOSS platform database
2. Installation and setup of IDE
   a. Installation of Eclipse
   b. Check out OPENCOSS platform server source code, and compilation of source code
3. Running OPENCOSS server in Eclipse IDE debugger
4. Building OPENCOSS server web application war files.

The obtained OPENCOSS platform now allows the integration of external evidence management tools.

3.2 Evidence Management Service Infrastructure Integration Prerequisites

The Evidence Management Service Infrastructure (EMSI) aims to guarantee the interoperability of various kinds of tools (see Figure 5), whereby evidence can be manually generated or automatically generated by tools (code generators, testing tools, safety analysis tools, etc.). The evidence from different kind of tools is gathered by means of standardized and well-defined evidential adapters.

During the software development and safety assurance, various kinds of tools are employed. For example, different forms of software testing can be used to validate and/or verify various parts of a system under development. External theorem provers can be used to automatically prove the correctness of (parts of) the code. A program that is proven correct can be considered more trustworthy than a section that has only been tested for correctness. EMSI provide a set of interfaces and protocols to communicate with a selected set of safety-critical development tools. The goal is not to cover all the possible tools, but to provide some adapters. Furthermore, a template scheme with an emphasis on tools commercialized by project partners is provided. This will be open and extensible for further integrations.

The ingredients for a proper integration are:

- A common registry of evidence types provided by external tools, this is strongly connected to the CCL and to the evidence taxonomy.
- A common format for evidence import/export from external tools. This is in the form of XML, every evidence artefact has a unique global identifier (GUID), and the data integrity has to be assured.
- A push and pull import and export mechanism to allow efficient exchange of data between the tools and infrastructure.
- Traceability of imported and remote evidence by means of the GUIDs.

3.3 Interface Required for Evidence Management Tools

This section contains how the evidence management service infrastructure is to be used from a developer's point of view. The user manual describing how to use the functionality implemented in the OPENCOSS platform can be found at:
https://svn.win.tue.nl/repos/opencoss/WP-transversal/Implementation/ThirdPrototype/OPENCOSS_Prototype3_UserManual.doc

Sections 2.6 and 2.7 of D6.6 describe two examples of how external tools can be integrated in the OPENCOSS platform.

OSLC is an option for integrating tools, but the EMSI will not provide functionality to support OSLC directly. However, the solution is close to the OSLC and shares some of the technical concepts.

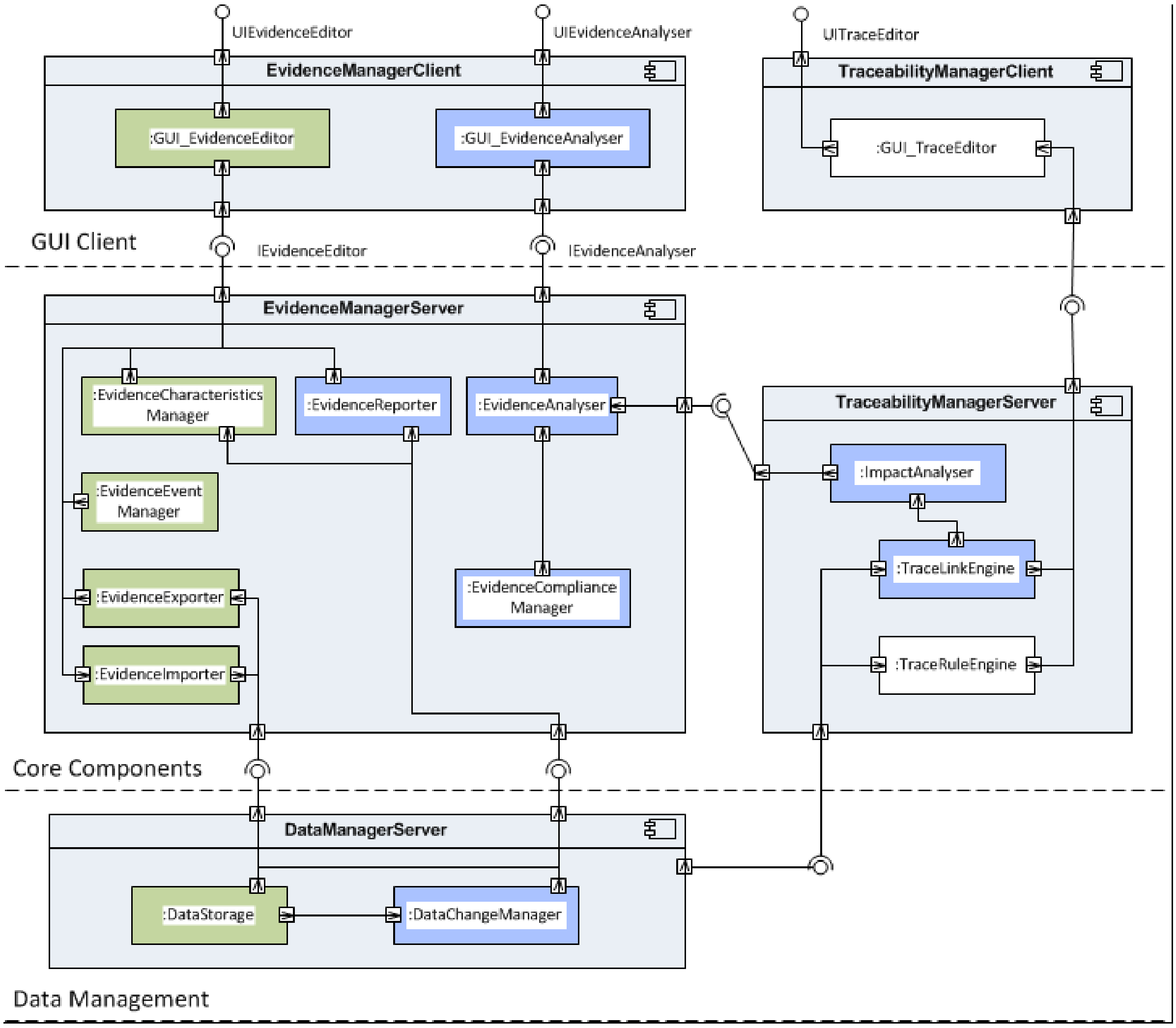


Figure 5. OPENCOSS Tool Components - components implemented in the 1st prototype are presented in green, while the components of 2nd and 3rd prototype are in blue

4 Governance of the Evidence Management Infrastructure

This chapter describes how the EMSI is governed and managed. It states who is responsible for what aspects of EMSI and how contacts can be made in case of questions, remarks or requests for changes with respect to the methodology.

4.1 EMSI within the OPENCOSS platform

The OPENCOSS Governance group, consisting of the participants of the OPENCOSS project, is responsible for managing and maintaining the OPENCOSS platform (Access & Security, reporting and data management), including the CCL. The information for managing and maintaining the platform is part of WP3.

The OPENCOSS tools will be delivered by means of baseline releases. A baseline release is defined by a specific version of each of the tools that are listed in the OPENCOSS platform. The definition of a new baseline necessarily implies that significant changes (enhancements) are brought to the above mentioned list of tools of which the OPENCOSS platform is made up: an enhancement may consist in adding a new function, keeping the functionality of the previous baseline unchanged, or may consist in changing some functionality, performance or non-functional characteristics of the previous baseline, as well as changes in the published interfaces and/or public data formats.

During the whole lifetime of the system, several releases of the same baseline can be issued:

- the first prototype release,
- optionally, one or more **consolidation** releases, consisting of intermediate releases in order to progressively build the full and coherent set of tools attached to the baseline, and finally
- one or more **maintenance** releases. These consist only of errors fixed after the publication of the first official release of a baseline.

The Change Control Management consists of the management of activities, which allow moving from one baseline release to another one. The Change Requests (CR's) offer a transparent, formal and ordered processing of the changes leading to new releases.

For an efficient management of the submitted CR's, the OPENCOSS Governance group is internally organized in order to receive, review and classify the CR's received from the submitters via an ad hoc specific tool publicly available, for instance Bugzilla. There will be a difference between tool specific error reports and infrastructure specific error reports.

To be accepted into the CCM process, the CR's must be formally correct. A Change Request shall only be submitted using an ad hoc tool publicly available, so as to manage them in an efficient way. The necessary infrastructure will be provided via a mature bug tracking systems, e.g., Bugzilla.

The information relevant for the submission of the CR is listed here below (where an asterisk is put, it shall be mandatory to fill the related field):

- Headline:* which gives a textual unequivocal identification and indicates the general topic of the CR, not exceeding a few words,
- Reference tool:* to which the CR refers (including version numbers),
- Error/Enhancement:* the rationale of the CR shall be given, so does the CR relate to either the need for debugging the specified baseline or to the need for functional or performances improvement,

- Problem/need description:* which gives a detailed overview about the problem/need. The reason for the CR shall be clearly indicated,
- Severity indication,
- Supporting documents for problem/need description: lists all files which are attached to the CR, in relation with the CR problem/need,
- Solution proposal by submitter: which indicates the solution preferred by the submitter,
- Supporting documents for solution proposal: lists all files which are attached to the CR, in relation with the proposed CR solution,
- Preliminary assessment of the benefits: which provides, in case of an enhancement, as a first step the order of magnitude of the benefits resulting from the expected improvement of performances, safety, reliability and maintainability,
- Supporting documents for preliminary assessment of the benefits: lists all files which are attached to the CR, in relation with the preliminary assessment of the benefits,
- Submitting Organisation:* which defines the proposer of the CR.

Contact person Name and Email address:* of the expert representing the previously-mentioned organisation, who will be the contact person in case of further exchange between originator and OPENCOSS governance group is needed. To provide the information relevant for the submission, the submitter shall log in to the ad hoc tool provided by the OPENCOSS project. As a general rule, within five working days after its submission, the Core Team within the OPENCOSS Governance group performs the pre-analysis of the CR. This pre-analysis consists in checking:

- that the mandatory fields are duly filled, and
- that the information provided in free text fields and attached documents, if any, is usable for further analysis.

When a CR cannot be accepted due to missing or unusable information, the CR state is changed to ‘rejected’, providing the reason(s) of such rejection. During a period of two months, the submitter will have the possibility to provide the required information in order to make the CR valid. If the required information has not been provided after this two months period, the CR shall be considered as definitively rejected. This event will be notified to the submitter, together with a short motivation.

At any stage of the CR workflow, it may appear that a CR may be linked to one or more other CR, either in terms of problem/need or in terms of the solution found. If it is made sure that a particular CR can be fully covered by another CR dealing with the same subject, the CR state changes to ‘superseded’; the reference to the superseding CR shall be indicated. Conversely, the CR, which supersedes one or more other CR’s, shall refer to the list of superseded CR’s. Each state transition shall be notified to the submitter, through the email address provided in the CR submission form.

The Core Team within the OPENCOSS Governance group is also in charge of assigning the incoming CR’s to the competent technical teams (Platform Developers) devoted to cope with the following areas of interest:

- Access & Security,
- Reporting
- Data management

In order to organize the work of the Core Team and the dedicated WG's in the most efficient way, and especially to manage logically a situation when there will be so many logged CR's that it will not be possible to treat all of them in the same time, each of the three abovementioned technical working groups will set priorities for the CR’s. Since it may depend on many non-technical factors, it is not possible to predefine an exhaustive list of criteria for the prioritization of CR's. However the CR's are stamped with a severity qualifier in order to help, together with e.g. the classification error/enhancement and the reference baseline release. If the CR is considered as relevant for the next expected baseline or is related to the maintenance of an existing baseline the CR state changes to 'Assigned'. If the OPENCOSS Governance Group estimates, on the basis of the information provided by the submitter, that this CR is relevant but not for the next expected baseline, the CR is postponed. The CR state changes to 'Postponed'. Internal meetings within the OPENCOSS Governance group will be held at regular intervals to discuss the current state of the CR's, their progress, the workload of the different technical WG's and any potential need of synchronization between them. The solution to any CR shall consist in a list of unambiguously identified changes to one or more tool(s) of the OPENCOSS platform; together with these proposed amendments, it is possible to add a separate justification for this solution. All these information will be stored in a specific database publicly available.

In addition to this, the user will be given access to a Frequently Asked Questions (FAQ) section recording the list of questions and answers related to the collection of evidences, the specification of evidence traceability and their evaluation. This FAQ section contains also QA related to the tool integration and tool usage. In order to submit a question, the user has to submit a specific form by filling out all the mandatory fields. The Governance group shall assign the question to the responsible technical team.

5 Guidance for Usage of the Evidence Management Service Infrastructure

This section presents guidance about how OPENCOSS evidence-related concepts should be used for evidence management with the EMSI. The section includes information about how the CCL is supposed to be used for evidence management in an assurance project, and aims to provide useful, relevant information on EMSI usage that has not been included in other OPENCOSS deliverables.

The section has been divided according to the five main functional areas for evidence management in the OPENCOSS tool platform.

5.1 Evidence Storage

Guidance concerning evidence storage focuses on evidence specification. More specifically, the purpose is to provide insights into and advice about how the artefacts of an assurance project should be specified with the EMSI. The guidance focuses on three main aspects.

Artefact Definition
For each instance of a given artefact type (e.g., requirement), an artefact definition must be specified (e.g., Req1, Req2, and Req3). Otherwise, it would not be possible to track their lifecycles. It must be noted that the lifecycle of each artefact (e.g., versions of an artefact definition) must be maintained independently.
The notion of artefact type in this description mainly corresponds to the CCL Reference Artefact, although this can vary depending on the nature or purpose of evidence management. In this sense, several artefact definitions, and their corresponding artefacts, can represent the materialisation (compliance map) of a given Reference Artefact.

Artefact granularity
The granularity of the artefacts of an assurance project can vary: set of documents (e.g., system specifications), document (e.g., requirements specification), parts of a document (e.g., a given single requirement), etc. The granularity will depend on the purpose of an artefact and of traceability-related purposes.
As a rule of thumb, an artefact (and thus an artefact definition) must be specified if: (1) the artefact must be linked to others; (2) the lifecycle of the artefact must be tracked, or; (3) the artefact is used in some other general OPENCOSS functional area (Prescriptive Knowledge, Assurance Project, Process, or Argumentation Management; see D2.3).

Company- or domain-specific practices
The concepts used for evidence management are generic and aim to support practices across different application domains and companies. However, there exist specific practices that are not directly and explicitly represented in the concepts, and a company must be aware of this.
As an example, a company might have its own criteria for evaluating the artefacts of its assurance projects. In this case, evaluation is a broad concept that supports company-specific evaluation practices.
The possibility of formally supporting custom extensions in a way that allows EMSI to keep track of those specific practices is currently out of the scope of WP6 work. However, such support could potentially be provided by means of the CCL vocabulary.

5.2 Evidence Traceability

Safety evidence traceability was defined in D6.3 as the degree to which a relationship can be established to and from artefacts that are used as safety evidence. D6.3 further presents the main reasons for tracing evidence and the main challenges. In this deliverable we focus on traceability between the artefacts of an assurance project. This mainly relates to the specification of artefact relationships. Other types of evidence-related traceability include, among others, the relationships between artefacts and activities, claims, or participants.

Possible relationships between artefacts [1][3][5] of which an EMSI user needs to be aware are:

- *Constrained_By*: a relationship of this type from an artefact A to an artefact B documents that artefact B defines some constraint on artefact A; for example, source code can be constrained by coding standards.
- *Satisfies*: a relationship of this type from an artefact A to an artefact B documents that artefact A realisation implies artefact B realisation too; for example, a design specification can satisfy a system requirement.
- *Formalises*: a relationship of this type from an artefact A to an artefact B documents that artefact A is a formal representation of artefact B; for example, a Z specification can formalise a requirements specification in UML or natural language.
- *Refines*: a relationship of this type from an artefact A to an artefact B documents that artefact A defines artefact B in more detail; for example, a low-level requirement can refine a high-level requirement.
- *Derived_From*: a relationship of this type from an artefact A to an artefact B documents that artefact A is created from artefact B; for example, source code can be derived from a system model when a source code generator is used.
- *Verifies*: a relationship of this type from an artefact A to an artefact B documents that artefact A shows that artefact B properties are true; for example, model checking results can verify a requirement.
- *Validates*: a relationship of this type from an artefact A to an artefact B documents that artefact A shows that that artefact B properties can be regarded as valid; for example, a test case can validate a requirement.
- *Implements*: a relationship of this type from an artefact A to an artefact B documents that artefact A corresponds to the materialisation of artefact B; for example, source code can implement an architecture specification.

Two relationships are already explicitly supported in the CCL: *Evolution_Of* (precedentVersion) and *Composed_Of* (artefactPart). Traceability is always bilateral, and you have implements and implemented-by relationships.

5.3 Evidence Evaluation

By evidence evaluation we mainly refer to the activity targeted at judging the adequacy of an artefact and the results associated with this activity. This activity is specified in the EMSI by:

- Specifying evaluation events for an artefact
- Associating the event with a specific evaluation (i.e., its information)

In this section we focus on the criteria that can be used for evaluating evidence.

SACM criteria for evidence evaluation and the result of the corresponding evaluation [2] are:

- *Completeness*: unknown, incomplete, draft, final, obsolete.
- *Consistency*: unknown, informal, semiformal, formal.
- *Originality*: unknown, derivative, original.

- *Relevance*: unkown, low, mediumLow, medium, mediumHigh, high.
- *Reliability*: unknown, unReliable, nonUsuallyReliable, usuallyReliable, fairlyReliable, completelyReliable
- *Significance*: unkown, low, mediumLow, medium, mediumHigh, high.
- *Strength:* a numerical value between 0 and 100.

Regarding other possible criteria, *trustworthiness* and *appropriateness* are proposed in [4]. As mentioned above, a company can have its own evidence evaluation criteria. The most common approach in industry for evidence evaluation is the use of checklists (see D6.2), thus conformance to a checklist or to some of its sections or fields can be used as evaluation criterion.

Further insights about evidence evaluation, especially in the scope of an argumentation structure, can be found in WP5 deliverables. D6.1 also presents the results from reviewing the literature on safety evidence provision, which includes information about other specific approaches for evidence evaluation (e.g., Bayesian Belief Networks).

5.4 Evidence Change Impact Analysis

In D6.3 we defined safety evidence change impact analysis as the activity that attempts to identify, in the body of safety evidence, the potential consequences of a change. Possible consequences can be the need for adding, modifying, or revoking some artefact. This section aims to provide insights into this activity.

As indicated in previous WP6 deliverables, a survey of the state of current practice concerning safety evidence change impact analysis has been conducted. The insights below are based on the results of the survey. The results are based on 97 valid responses representing 16 application domains, 28 countries, and 47 safety standards. It must also be noted that the conceptual guidance for evidence traceability is also important for impact analysis, as recording and maintaining traceability is necessary for impact analysis.

The respondents of the survey acknowledged having dealt with evidence change impact analysis in the following general situations:

- Modification of a new system during its development
- Modification of a new system as a result of its V&V
- Reuse of existing components in a new system
- Re-certification of an existing system after some modification
- Modification of a system during its maintenance
- New safety-related request from an assessor or a certification authority
- Re-certification of an existing system for a different operational context
- Re-certification of an existing system for a different standard
- Re-certification of an existing system for a different application domain
- Changes in system criticality level
- Independent assessment of the risk management process
- Hazards identified after the fact
- Re-certification for temporary works
- Accident analysis
- System of system reuse

Regarding the artefact types involved in evidence change impact analysis, Table 1 shows the median frequency (5-point Likert scale: never, few projects, some projects, most projects, and every project) with which different artefact types trigger impact analysis and are affected by changes in the body of the safety evidence. In the survey, the artefact types were presented as:

- System Lifecycle Plans (e.g., development plans, validation and verification plans, quality plan, safety plan, modification procedures, and operation procedures)
- Reused Components Information (e.g., historical service data and reliability specifications)
- Personnel Competence Specifications (e.g., personnel training and experience assessment)
- Safety Analysis Results (e.g., the results from Fault Tree Analysis and Failure Mode and Effects Analysis)
- Assumptions and Operation Conditions Specifications (e.g., the constraints on the working environment of a system)
- Requirements Specifications (e.g., safety requirements or performance requirements)
- Architecture Specifications (e.g., system components and AADL diagrams)
- Design Specifications (e.g., the internal characteristics of system components and SysML diagrams)
- Traceability Specifications (e.g., the relationships between requirements and test cases and between requirements and design)
- Test Case Specifications (e.g., the inputs, execution conditions, and predicted results using a system)
- Tool-Supported V&V Results (e.g., testing results, simulation results, and formal verification results)
- Manual V&V Results (e.g., inspection results and review results)
- Source Code (e.g., Ada code or C code)
- User Manuals, Installation Guides, Operations Guides, etc.
- Safety Cases (documented argument aimed at providing a compelling, comprehensive, and valid case that a system is safe for a given application in a given operating environment)

In summary, it is important that the EMSI users are aware of the fact that evidence change impact analysis can be necessary in the general situations listed, that impact analysis can be necessary from changes in the artefact types mentioned, and that they can be affected by changes.

Table 1. Role of different artefact types in evidence change impact analysis

	Impact Analysis Trigger	**Affected by Changes**
System Lifecycle Plans	Some projects	Few projects
Reused Components Information	Few projects-Some projects	Few projects
Personnel Competence Specifications	Few projects	Few projects
Safety Analysis Results	Most projects	Some projects
Assumptions and Operation Conditions Specifications	Some projects	Some projects
Requirements Specifications	Most projects	Some projects
Architecture Specifications	Some projects	Some projects
Design Specifications	Most projects	Some projects
Traceability Specifications	Most projects	Some projects
Test Case Specifications	Most projects	Some projects
Tool-Supported V&V Results	Some projects	Few projects
Manual V&V Results	Some projects	Most projects
Source Code	Most projects	Some projects
Safety Cases	Some projects	Some projects

5.5 Integration with External Tools

Current EMSI integration with external tools mainly corresponds to the possibility of using artefacts from a SVN as evidence. The possibility of providing specific conceptual guidance will be analysed in the future, once further integration mechanisms have been implemented. At this moment the integration of QM and Medini Analyze is one of the few examples, see Appendix A.

6 Conclusions

This methodological guide gives an overview of how external evidence management tools can be integrated with the EMSI, based on the OPENCOSS platform. It gives guidelines for integration, guidance on how to deal with change requests and describes the overall structure of the underlying data structures.
This methodological guide has been a living document through the project, and the current version reflects this fact. Every time a new external tool is added to the EMSI it led to adaptation of this guide. The third prototype gave new insights on the use of the API, an example of integration is shown in Appendix A.

Abbreviations and Definitions

CCL	Common Certification Language
DX.Y	OPENCOSS deliverable X.Y
EMSI	Evidence Management Service Infrastructure
SACM	Structured Assurance Case Metamodel
FMEA	Failure Mode and Effects Analysis
FTA	Fault Tree Analysis
V&V	Verification and Validation
WP	OPENCOSS work package

References

[1] Espinoza, A., et al.: Analyzing and Systematizing Current Traceability Schemas. In: SEW'06

[2] OMG: Structured Assurance Case Metamodel (SACM), version 1.0. 2013

[3] Pohl, K.: Requirements Engineering: Fundamentals, Principles, and Techniques. Springer, 2010

[4] Sun, L., Kelly, T.: Elaborating the Concept of Evidence in Safety Cases. In: SCSC 2013

[5] Wiegers, K.E.: Software Requirements, 2nd ed. Microsoft Press, 2003

Appendix A

A case study on EMSI usage is based on the Section 4 of D3.3.

The OPENCOSS platform has been designed to be open towards external tools, either to integrate with the existing tool landscape or to provide facilities for further potential analysis or automation tools for the future. For both types of tools implementation technologies were selected aiming at guaranteeing a future proof approach as well as cross-platform integration capabilities as web-technologies and REST services.

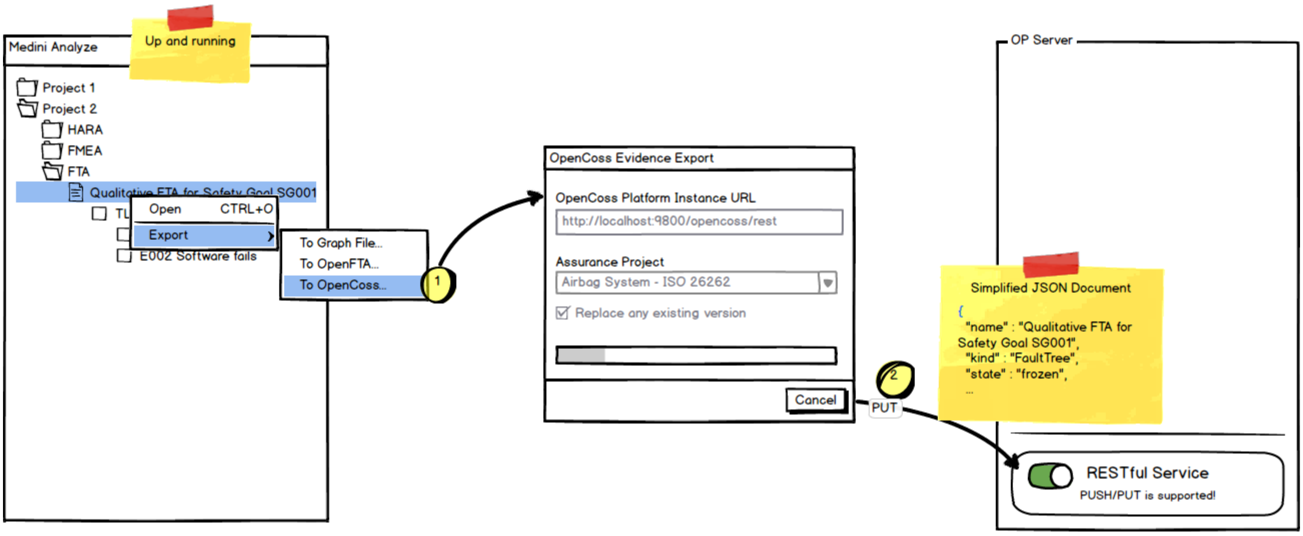
A challenge was the integration of tools that are not per-se web and server based, as most analysis tools for safety analyses current are rich clients. They may act as clients in an integration scenario but they cannot be used as reliable sources of safety information. Some details about the integration of Medini Analyze as external evidence tool as well as Atego Process Director as external process tool is outlined in the sections below.

Integration with external evidence tools

Traditionally many analysis techniques as FMEA, FTA or PHA do not require any specific tool and thus have been relying on Off-The-Shelf tools such as Microsoft Office (Excel and Word) for quite a while. This situation is rapidly changing and model-based tools that better integrate with the design tool landscape become more attractive. These tools are often rich-client tools that store their data in proprietary file formats or databases and are not designed out of the box for integration with a centralized assurance management platform such as OPENCOSS. Two integration scenarios were identified and implemented that can bring such tools and the OPENCOSS platform together:

1. A **push** scenario was the tool is used as is but the results are finally or periodically "pushed" (aka published) to the platform. For that scenario dedicated evidence management service and API of the OPENCOSS platform was developed and used.
2. Secondly a **pull** scenario was approached were the platform itself is pulling data from external tools that have to offer appropriate interfaces to do so. For that purpose a proprietary service is offered by the tool to query and pull evidence information from the tool. This interface is tool proprietary.

These scenarios are depicted in Figure 6.



OpenCoss Platform Instance URL: http://localhost:9800/opencoss/rest

Assurance Project: Airbag System - ISO 26262

☑ Replace any existing version

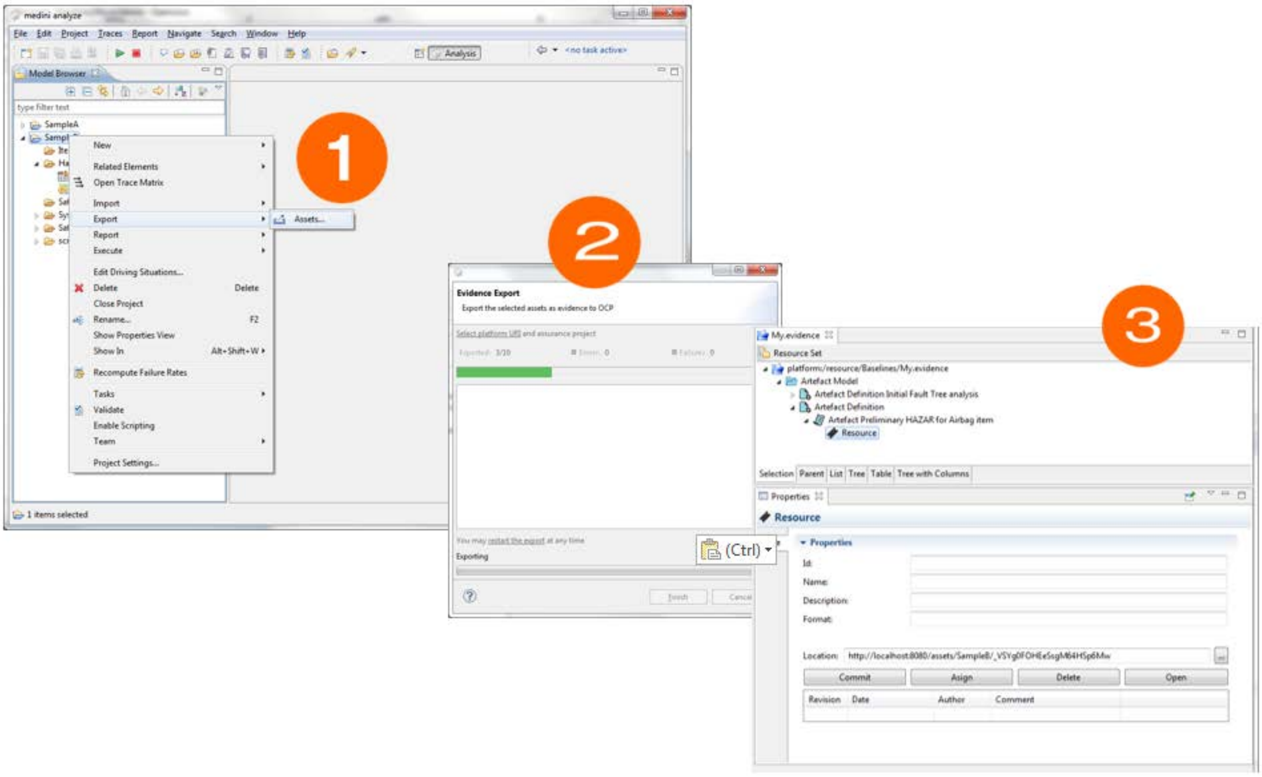
```
{
 "name" : "Qualitative FTA for
Safety Goal SG001",
 "kind" : "FaultTree",
 "state" : "frozen",
 ...
```

Figure 6. Medini analyze and OPENCOSS data exchange (1).

Medini analyze as an example

Both parts were implemented using the Medini Analyze safety tool. This tool uses a local workspace to store safety analysis and source data. To offer an API for the pull scenario a lightweight web server was integrated into the tool.

All meta-data of analysis results can be pushed as Artefacts to the OPENCOSS platform using the evidence interface. This interface provides a CRUD (Create/Read/Update/Delete) style of interface to interact with the OPENCOSS platform. A critical part was to find a suitable mapping between the tools metamodel and the OPENCOSS CCL metamodel. There are quite a number of differences in concept that required a harmonization.



Evidence Export
Export the selected assets as evidence to OCP
Select platform URI and assurance project
Exported: 3/10 Errors: 0 Failures: 0
You may restart the export at any time
Exporting

Id:
Name:
Description:
Format:
Location: http://localhost:8080/assets/SampleB/_VSYg0FOHEeSsgM64HSp6Mw

Revision	Date	Author	Comment

Figure 7. Medini analyze and OPENCOSS data exchange (2).

Further details on the scenarios and the implementation can be found in D6.5.