Collaborative Large-scale Integrating Project

Open Platform for EvolutioNary Certification Of Safety-critical Systems

Intermediate implementation of the evidence management service infrastructure
D6.5

Work Package:	WP6: Evolutionary Evidential Chain
Dissemination level:	Public
Status:	Final
Date:	10th March 2015
Responsible partner:	Janusz Studzizba (Parasoft S.A.)
Contact information:	januszst@parasoft.com

Contributors

Names	Organisation
Idoya del Rio, Huascar Espinoza	TECNALIA Research & Innovation
Janusz Studzizba, Dariusz Oszczedlowski	Parasoft SA
Jose Luis de la Vara, Sunil Nair	Simula Research Laboratory
Jan Mauersberger, Joachim Hößler	ikv++ technologies ag
Jérôme Lambourg	AdaCore

Document History

Version	Date	Remarks
V0.1	2013-11-04	Document creation, initial ToC, and initial content
V0.2	2013-11-07	Description of the first prototype
V0.3	2013-11-11	Update on the description of the first prototype
V0.4	2013-11-13	Document review, update, and description of the plans for future implementation
V0.5	2013-11-15	Document review and update
V0.6	2013-11-18	Ready for WP review
V0.7	2013-12-04	Ready for PB review
V1.0	2013-12-10	Ready for EC review
V1.1	2015-03-10	Updated to align with the final status of work

TABLE OF CONTENTS

List of Figures

List of Tables

Abbreviations

API	Application programming interface
CCL	Common Certification Language
DAO	Data Access Object
DX.Y	OPENCOSS deliverable X.Y
DoW	Description of Work
EMF	Eclipse Modeling Framework
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
OSLC	Open Services for Lifecycle Collaboration
QM	The Qualifying Machine
REST	Representational State Transfer
SVN	Subversion
TX.Y	OPENCOSS task X.Y
V&V	Verification and Validation
WP	OPENCOSS Work Package

Executive Summary

This document (D6.5) is the fifth deliverable of WP6. This WP aims to define a safety certification management infrastructure for an evolutionary evidential chain. The overall goal of D6.5 is to present the intermediate implementation of the evidence management functions of the OPENCOSS tool platform.

D6.5 summarises the implementation and methodological work performed T6.4 task in the scope of implementation of the first prototype. Since T6.4 is an implementation task, the core and main result of this deliverable is the implementation of a prototype tool for the evidence infrastructure in OPENCOSS platform. In particular the following core items constitute D6.5 achievements, referenced in this umbrella document:

- The installable first prototype of the evidence management infrastructure of OPENCOSS tool platform
- The user manual of the prototype
- The installation procedures
- A description of the source code

In parallel to the core implementation work, a deep technology research has been performed by the partners participating in T6.4. The results from this research are briefly presented in this document and they were used in the implementation of the second prototype of the OPENCOSS tool platform, which are presented in D6.6.

We also indicate the detailed requirements for evidence management specified in D6.2 that have been deployed for D6.5.

1 Implementation of the First Prototype of the OPENCOSS Tool Platform

As a general strategy, the OPENCOSS Consortium decided to follow an incremental approach for research and development, unlike the DoW approach, which did not take into account the evolution of the different solutions. To do so, the OPENCOSS project released in three iterations. This report describes the progress of the first iteration. The further development and results are described in D6.6 document.

In implementation-related documents, the naming convention follows these concepts:

- **Environment** refers to (a large part of) software tools used to manage a safety project process. The OPENCOSS tool platform is the main environment in this document.
- **Workbenches** refer to only one or a few activities (e.g., "*Evidence Management*" workbench).
- **Tools** refer to only specific tasks in the software tool (e.g., "Evidence Analysis").

A Tool is an element of a Workbench that resides in the OPENCOSS Environment.

This deliverable is concerned with one of the Workbenches: ***Evidence Management***. This workbench manages the full life-cycle of evidence and evidence chains. This includes evidence traceability management and impact analysis. In addition, this module is in charge of communicating with external engineering tools (requirements management, implementation, V&V, etc.). The functionality implemented in the first prototype is described in Section 1.1

The goal of the first prototype tools is threefold:

(a) ***Evaluate CCL Metamodels***. Some of the key aspects are:

   a. Completeness of the Metamodels regarding the critical information to be used in the tool-assisted processes supported by the OPENCOSS tool platform (this has been specified in D2.2 and D2.3).
   b. Ability to create models of Standards, Company-specific Processes/Practices, and Project-specific Assets using different kinds of styles and granularity levels - according to the Case Study data from the automotive, avionics, and railway domains.
   c. Ability to specify modularity and reuse of systems assurance and certification information (e.g. arguments modularity).
   d. Ability to capture information that might be externally managed by different tools (e.g. artefacts stored in SVN repositories, or process definition and execution managed by workflow tools).
   e. Mappings between different model elements to support reuse (cross-project, cross-domain, etc.).

   Excluded from this first prototype evaluation are aspects such as (i) *Vocabulary,* although there is some initial work being done in this area, (ii) Cross-domain reuse (iii) Compositionality of systems assurance/certification based on a contract-based approach, and (iii) Compliance assessment, among others.

(b) ***Evaluate Tools Functionality and Usability***. The current version of the tools is Eclipse-based and was intentionally based on EMF technology to speed-up its development in the first stage. These prototype tools are intended to evolve to different technology in the future, following the studies currently being

performed in WP5, WP6 and WP7 (Web technologies, OSLC, etc.). Some key aspects to be evaluated in this first prototype tool are:

a. Functionality centered on Storage, Edition, and Navigation (basic Edition functionality).
b. Graphical editors to represent information such as Standards and Company-specific Processes (Both also referred to as Reference Frameworks) and Argumentation.
c. Editor's usability in terms of effort to perform tool actions.

(c) ***Evaluate the Data Completeness and Suitability from Industrial Case Studies***. Currently we got data from industrial partners about three case studies. These data cover different aspects of the assurance and certification process and in some cases have different levels of granularity, detail and completeness. Since every case study may evaluate particular areas of the OPENCOSS goals, the evaluation of data completeness might be necessary. The goal of using tools to model the Case Studies is to evaluate the following aspects:

a. Availability of information to perform a meaningful excerpt of the case studies.
b. Ability to extract the required information from case studies data so as to model them to achieve OPENCOSS goals (guidance, reuse, process automation, etc.).

D6.5 describes the functionality implemented and the references to the implementation assets.

1.1 Implemented Functionalities

As stated above, the "Evidence Management" workbench's scope includes the full life cycle of evidence and evidence chains, evidence traceability management, evidence impact analysis, and communication with external engineering tools. This section details both the requirements satisfied and the components deployed to show the implementation scope of the first prototype.

Regarding the requirements, Appendix A includes tables with the detailed requirements specified in D6.2, indicating if the requirements have been implemented in the first protytope. Regarding the components, Figure 1 (based on a similar one in D6.3) shows the deployed components, which are in green.

Inside the GUI Client layer, the GUI Evidence Editor component has been deployed. This component provides different kinds of user interfaces for the visualization and edition of evidence information.

As an example, Figure 2 shows the user interface for viewing versions of artefacts and their files.

Inside the Core Components layer, the following components have been implemented:

- Evidence Characteristic Manager
  This component allows a user to:
  - Read and edit evidence characteristics from the Data Manager Server.
  - Create and delete evidence element items.
  - Create and use libraries of evidence properties
- Evidence Event Manager
  At this moment, the prototype allows a user to define events but not to manage the lifecycle of evidential artefacts.
- Evidence Exporter and Evidence Importer.
  The prototype has a functionality that allows a user to associate data files to the evidence. These data files can be stored in local or external repositories. In the case of the external repository, a SVN server has been used.

Inside the Data Management Layer, the Data Storage component has been deployed to manage the CCL entities of the evidence metamodel.

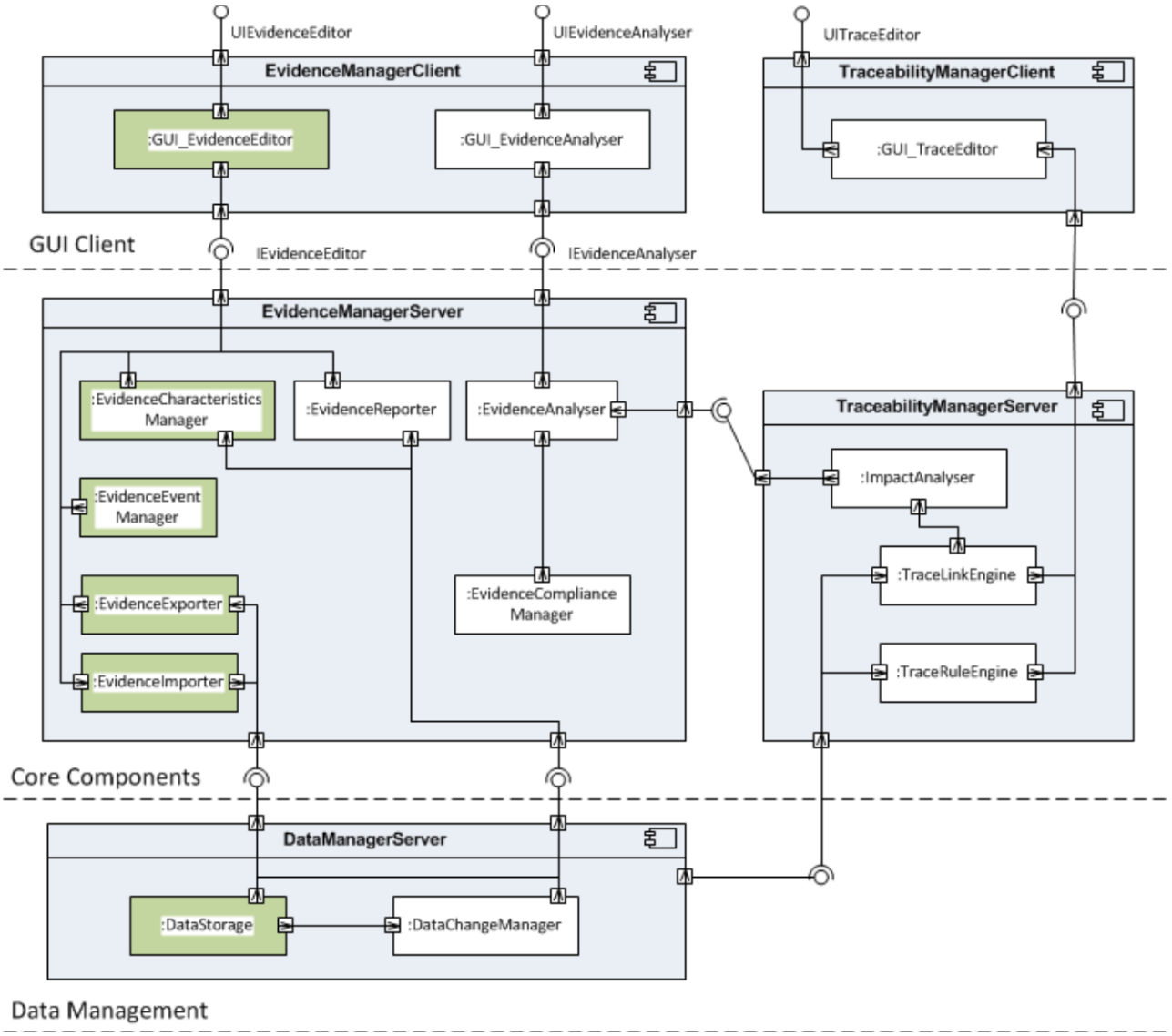

Figure 1. OPENCOSS Tool Components - the components implemented in the first prototype are presented in green

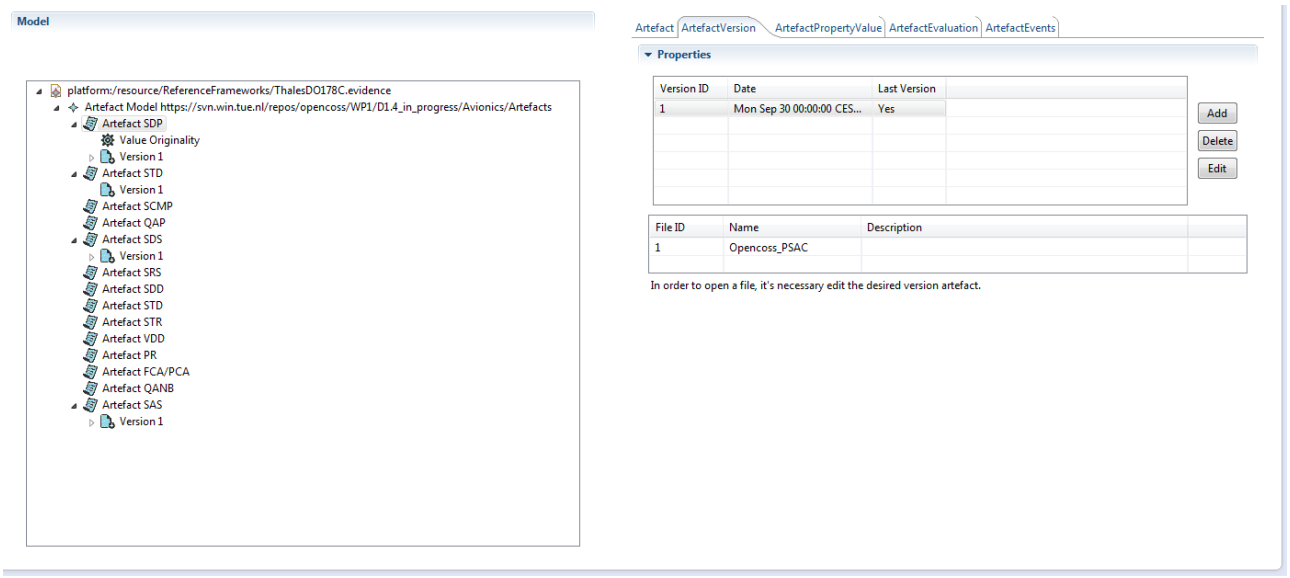

Figure 2. GUI Evidence Editor

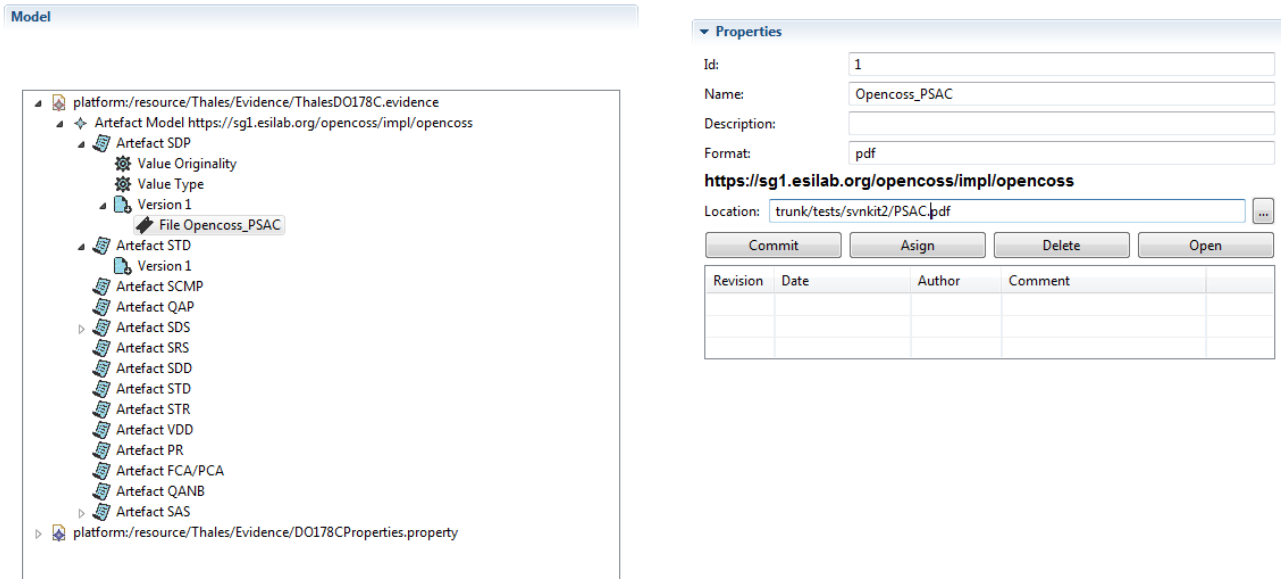

Figure 3. Export and import of evidence

1.2 Installation

The steps necessary to install the first prototype are described in the document “OPENCOSS first developer guide” (Date 22/10/2013 - V0.5). This document is hosted, with the source code of the prototype, in the remote location https://svn.win.tue.nl/repos/opencoss-code/branches/prototype/0.5, under the doc branch.

The “OPENCOSS first developer guide” is a developer guide of the first OPENCOSS tool prototype implementation. The developers can find the source code installing instructions, step by step, in order to set up their workspaces to implement new functionalities to the OPENCOSS Prototype.

1.3 User manual

The user manual for Evidence Management is detailed in the document “OPENCOSS first prototype user manual” (Date 23/10/2013 - V0.6; Section 7). This document is hosted, with the source code of the first

prototype, in the remote location https://svn.win.tue.nl/repos/opencoss-code/branches/prototype/0.5, under the doc branch.

In summary, the "OPENCOSS first prototype user manual" is a user manual of the first OPENCOSS tool prototype implementation. The users can find the installation instructions, the tool environment description, and the functionalities for the creation of Reference Frameworks (models representing Standards, Regulations, or Company-specific Processes), Assurance Projects and the associated Baseline (subset of Reference Framework to be applied in a specific assurance project), Evidence models (Artefacts), Process models (Activities), Compliance Maps (so far, compliance maps from Reference Artefacts to Artefacts), and Argumentation models.

1.4 Source Code Description

After installing the first prototype and following the steps described in the document "OPENCOSS first developer guide" V0.5, all the source code can be found under the plugins branch.

Once all the plugins are installed, these are the necessary ones for Evidence Management:

- org.opencoss.evm.evidspes
  In this plugin, the evidence metamodel is defined and stored, and the Java implementation classes for this model are generated.
- org.opencoss.evm.evidspes.edit
  The edit plugin includes adapters that provide a structured view and perform command-based edition of the model objects.
- org.opencoss.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 to modify the objects in a standard property sheet.
- org.opencoss.evm.evidspec.preferences
  This plugin defines the default preferences for the communication with the SVN repository, thus it defines the type of repository (local o remote) and a user and password to connect with the remote repository.
- org.opencoss.infra.svnkit
  In this plugin, the functionalities necessary for the communication with the repository SVN (SVNKIT V1.3.8) are defined, to export and import artefacts.

In addition, theses plugins are necessary to handle the evidence properties:

- org.opencoss.infra.properties
  This plugins contains the definition of the Property metamodel, and the Java implementation classes for this model.
- org.opencoss.infra.properties.edit
  As the edit plugin for evidence, this plugin contains a provider to display the model in a user interface.
- org.opencoss.infra.properties.editor
  As the edit plugin for evidence, this plugin is an editor to create and modify instances of the model.

Figure 4 shows all the plugins above.

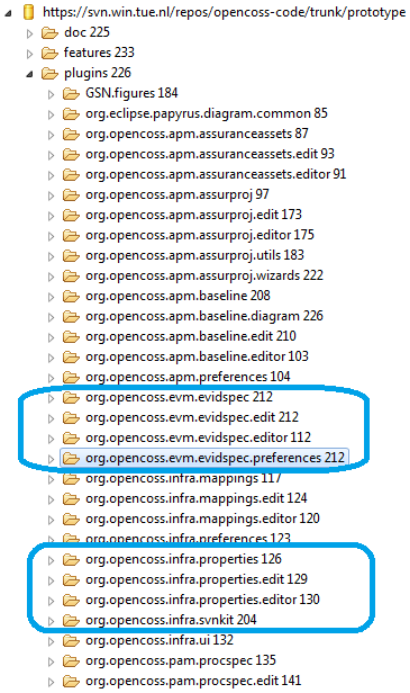

Figure 4. Evidence management plugins

2 Research Work

Additionally to the core implementation work described in Section 1, architecture and technology research that is related with implementation has been performed. The main results and conclusions are presented in this section.

2.1 Common Storage Provider

An evaluation of EMF-enabled storage technologies was performed. Results of this evaluation are gathered in the WP6/D6.5_in_progress/TechnologyResearch/CDOvsTeneovsTexovsEMFStore/ folder. As a result of this evaluation, it was decided that Teneo (http://wiki.eclipse.org/Teneo) should be used in the OPENCOSS tool platform. Teneo is a database persistency solution for EMF.

After several initial meetings with OPENCOSS partners, the initial approach for manual implementation of a properly granulated DAO (Data Access Object) service layer that would be accessed by all OPENCOSS modules (e.g. editor tools) was dropped. This happened due to the concern of other partners who work with the EMF technology and did not want to manually implement extensions to EMF-generated editors. As a result of this, as a compromise, another approach was suggested: to provide the partners with a way to work with the EMF's Resource API, driven internally by Teneo. All participants agreed upon it. The effort to implement this approach resulted in the creation of an initial prototype named StorageProvider.

As a result of the first prototype, StorageProvider skeleton code concept has been implemented. The main idea behind it is to create an additional abstraction layer for HibernateResource (provided by Teneo). With this abstraction layer, StorageProvider should prevent some erroneous interactions with the HibernateResource.

As StorageProvider is available for every OPENCOSS tool as a monolithic module and the one and only way to interact with the centralized OPENCOSS database, it provides functionality such as:

- Checking tool compatibility with current database/schema versions to prevent data corruption
- Performing some sanity checks and validation of the data pushed to the database

2.2 Approach to using OSLC Technology

OSLC as a tool integration approach (loose coupling of tools instead of point-to-point integration) and OSLC as a technology foundation (Web based, REST API, using Web standard) were identified as a candidate technology for prototype development in OPENCOSS. OSLC is supported by major organizations and among others by IBM, General Motor, Oracle, Siemens, etc. Domination by IBM is a possible risk.

Even though OSLC is a promising technology addressing some of the fundamental problems of past integration approaches, it does not suit the first and most likely the second prototype development phase. The major concerns are the impact on the complexity of the tooling infrastructure – especially OPENCOSS-internal – as well as the initial effort required to get a minimal OSLC environment running among all the tools and platform services.

Some experiments run in the project with the integration of known "OSLC Ready" software (e.g. DOORS) have shown a significant level of complexity that has to be mastered before gaining access to OSLC resources. Furthermore, some of the tested products themselves still seem to be "under development". Workflows and interactions are changing constantly over minor versions and are not stable. Experience from other EC projects such as iFEST has also shown that there are not mature enough tools and libraries

for OSLC yet. As a consequence, it has been decided that OSLC is not going to be used as an implementation technology in the second prototype phase.

However, it has been planned to further evaluate OSLC in the following months. Furthermore, OPENCOSS structures and conceptual models might be adapted to the possible future use of OSLC if appropriate and required. Existing OSLC concepts and specifications such as "Asset Management" would be compared to the current CCL metamodels. The changes that could be required to the CCL metamodels in order to consolidate or be compliant with an OSLC approach or specification would be mentioned in WP4.

2.3 Integration with the Qualifying Machine

Integration with a Qualifying Machine (QM) has been studied as a part of work in the first prototype. However after further work in the second prototype, the idea of QM integration has been dropped.

The Qualifying Machine is an open-source tool under development by AdaCore. Its aim is to ease the certification of safety-critical software by providing many features dedicated to help users with their compliance with safety standards. Such features are:

- Automatic reference of artefacts from a model-based description
- Traceability handling from various sources: naming conventions, explicit reference in the artefacts, and explicit from a traceability matrix.
- Impact analysis with monitoring of artefacts data.
- Documentation generation from the referenced artefacts, with scripting support

In the context of OPENCOSS, the most interesting features are the traceability handling and impact analysis, which are important functional blocks of WP6. The monitoring of artefact data could also be an interesting feature to help the WP6 components determine if the stored evidence is properly synchronized with their potentially existing data stored outside of the OPENCOSS tool platform.

There are however several things to consider to make this integration possible. First of all, communication between the QM and the rest of the OPENCOSS tool platform has to be implemented. One of the main issues at hand is that most of the platform is developed in Java, with base services (such as the storage manager) providing services using Java APIs only. The QM is however written in Ada 2012, and embeds a Python 2.7 interpreter as well as RESTful HTTP APIs. Therefore, assessing whether the integration of the QM is cost-efficient in the context of OPENCOSS is important and crucial. The main solution to this issue would be via the use of a tool named AJIS (Ada-Java Interfacing Suite), which automatically creates Java bindings from Ada, or Ada bindings from Java.

A second issue that has already been partially resolved is to make sure that the QM's internal model for handling artefacts is compatible with the CCL, the language used by the OPENCOSS tool platform to store its data and work on it. Although the work is still in progress, most of the issues are either easy to solve or not significant. A proper report still needs to be written and reviewed.

Also, if the QM is handling traceability and impact analysis, it may be partially used in other areas of WP6, such as a support for some of the functionalities of the Evidence Manager. Such use and integration would also require more design.

3 Plans for Implementations in the second and third prototypes

As mentioned above in this document, the OPENCOSS Consortium decided to follow an iterative approach for research and development. D6.5 presents a summary of the work done in the first development iteration (i.e., implementation of the first prototype).

The work planned for the implementation of the second prototype can be divided into 2 phases:

1. Before having feedback from the users of the first prototype, the following detailed architecture, technology research and implementation work has been planned:
    - Implementation of a prototype of the common storage provider, as described in Section 2.1.
    - Incorporating the Common Storage Provider as a single common gateway to data storage, used by all OPENCOSS platform tools and workbenches
    - Technology research regarding server side implementation
    - Design of the detailed Process API (mainly WP7), including mapping to CCL entities. Atego Process Director would be taken as a reference external tool.
    - Design of the integration and communication of the QM modules with the OPENCOSS tool platform or pieces of functionality.
    - Design of GUI mock-ups for specific pieces of functionality not covered by the first prototype (e.g., the evidence evaluation functionality).

2. After receiving and processing the entire feedback from the users of the first prototype, OPENCOSS partners had means to prioritize the work and decide which pieces of functionality should be implemented in the second prototype. The feedback from the first prototype has been received from:
    - The current evaluation of the first prototypes with industrial case studies data
    - The EC review in January 2014
    - The initial work on the architecture, storage approach, technology, and GUI/web mock-ups for the second prototype

4 Conclusion

D6.5 has presented the status of the initial phase of implementation of the evidence management infrastructure of the OPENCOSS tool platform. This status corresponds to the development of a first prototype in the Eclipse platform, from which the infrastructure has been developed further.

In addition to this prototype, the work towards creating D6.5 has focused on the research, analysis, and test of technologies. The final evidence management infrastructure would be developed according to the results and conclusions from this piece of work. The main areas addressed are related to the implementation of a common storage provider, the use of the OSLC technology, and the integration with the QM.

Finally, we have presented our plans to progress in the implementation of the infrastructure in the second prototype.

All this work have served as basis for a second prototype, which has been presented in D6.6.

Appendix A. Detailed Requirements for Evidence Management Deployed in D6.5

This appendix presents the detailed requirements specified in D6.2 that have been deployed in the first prototype of the evidence management infrastructure. These requirements are listed in Tables 1, 2, 3, 4 and 5, and grouped by functional area groups. The column labelled as "Deploy" indicates the state of the deployment, and the column labelled as “Comment” is a small explanation of the state of deployment.

As presented in the below table, only limited number of D6.2 requirements have been implemented in the first prototype. The reason is that the target of the first prototype was the implementation of entire CCL metamodel and UI editors for the data stored in OPENCOSS platform. This enabled industry partners to use the OPENCOSS platform first prototype by entering data (argumentation, evidence pieces, process items, etc.) of their own safety projects and give the informative feedback for the second prototype work. Apart from model and data editors, no automated functionality like impact analysis, API to external process tools, traceability visualization have been implemented in the first prototype.

Table 1. Component level requirements for evidence storage

ID	Name	Deploy	Comment
01_01	Evidence item to provide	*	Part of D4.5 CCL Editors (Reference Framework Editor)
01_02	Evidence types to provide		
01_03	Apply existing evidence characterization model		
01_04	CCL-based evidence item characteristics	X	
01_05	Association of evidence types to evidence items		
01_06	Apply existing evidence types		
01_07	Association of artefacts to evidence items	X	
01_08	CCL-based artefact information	X	
01_09	Modification of artefacts associated to evidence items	X	
01_10	Association of already-used artefacts to evidence items		
01_11	Evidence item drop	X	
01_12	Confirmation of evidence item drop	X	
01_13	Drop of evidence item characteristics after evidence item drop	X	
01_14	Drop of artefacts after evidence drop	X	
01_15	Evidence item information modification	X	
01_16	Evidence item reuse		
01_17	Record of evidence item history		
01_18	Automatic evidence repository creation		
01_19	Evidence item characteristics status specification		
01_20	Use of colours to report on evidence item characteristics completeness		
01_21	Use of colour to report on evidence item characteristics incompleteness		
01_22	Tree-view for display of evidence items		
01_23	Intermediate nodes of tree-view		
01_24	Modification of intermediate nodes of tree-view		
01_25	Drop of intermediate nodes of tree-view for display of		

	evidence items		
01_26	Association of evidence types to intermediate nodes of tree-view		
01_27	Proposal of intermediate nodes for tree-view		
01_28	Inclusion of evidence items in tree-view		
01_29	Automatic inclusion of evidence items in tree-view		
01_30	Generation of HTML-based evidence reports		
01_31	Generation of document-based evidence reports		
01_32	Customization of evidence report generation		

Table 2. Component level requirements for evidence traceability

ID	Name	Deploy	Comment
02_01	Evidence traceability link specification		
02_02	CCL-based specification of evidence traceability links		
02_03	Definition of evidence traceability links types		
02_04	Reuse of evidence traceability links types		
02_05	Display of source traced evidence items		
02_06	Display of target traced evidence items		
02_07	Notification of evidence traceability links correctness		
02_08	Use of colours for evidence traceability links correctness		
02_09	Evidence traceability link proposal		
02_10	Evidence traceability link drop		
02_11	Confirmation of Evidence traceability link drop		
02_12	Evidence traceability link drop after evidence item drop		
02_13	Evidence traceability links completeness assessment		
02_14	Matrices for traceability visualization		
02_15	Source and target evidence types in matrices for traceability visualization		
02_16	Display of required actions related to evidence evaluation		
02_17	Creation of evidence traceability links in matrices		
02_18	Models for visualization of chains of evidence		
02_19	Tables for traceability visualization		
02_20	Evidence types of tables for traceability visualization		
02_21	Columns of tables for evidence visualization		
02_22	Creation of evidence traceability links in tables		
02_23	Reuse of tables for traceability visualization		

Table 3. Component level requirements for evidence evaluation

ID	Name	Deploy	Comment
03_01	CCL-based evidence item evaluation	X	
03_02	Evidence evaluation rationale		
03_03	Possible relationships with the evaluation of related evidence items		
03_04	Definition of evidence item evaluation criteria		
03_05	Modification of evidence item evaluation criteria		
03_06	Reuse of evidence item evaluation criteria		
03_07	Definition of categories for evidence item evaluation criteria		
03_08	Modification of categories for evidence item evaluation criteria		
03_09	Reuse of categories for evidence item evaluation criteria		
03_10	Modification of evidence item evaluation		
03_11	Specification of required actions after evidence evaluation		
03_12	Email for required actions after evidence evaluation		
03_13	Evidence item associated to a required actions after evidence evaluation		
03_14	Use of colours in evidence items associated to a required actions after evidence evaluation		
03_15	Status of actions related to evidence evaluation		
03_16	Display of required actions related to evidence evaluation		
03_17	Completion of actions required as a result of evidence evaluation		
03_18	Use of colours to report on the status of actions required after evidence evaluation		
03_19	Report on evidence set completeness		
03_20	Report on evidence set adequacy		

Table 4. Component level requirements for evidence change impact analysis

ID	Name	Deploy	Comment
04_01	Determination of evidence items affected by a change		
04_02	Determination of evidence traceability links affected by a change		
04_03	Confirmation of changes		
04_04	Impact of a possible evidence change		
04_05	Specification of actions required after evidence item changes		
04_06	Evidence items associated to actions required after evidence item changes		
04_07	Evidence traceability links associated to actions required after evidence item changes		
04_08	Suggestion of actions after evidence item change		
04_09	Email with required actions after evidence item changes		
04_10	Status of actions related to evidence change		
04_11	Display of required actions after evidence item changes		
04_12	Completion of actions required after evidence item changes		
04_13	Use of colours to report on the status of actions required after evidence item changes		
04_14	Use of colours to report on the status of evidence items after evidence item changes		
04_15	Use of colours to report on the status of evidence traceability links after evidence item changes		

Table 5. Component level requirements for integration with external tools

ID	Name	Deploy	Comment
05_01	Evidence item information import		
05_02	Automatic artefact association to imported evidence item		
05_03	Tool information about the artefact associated to imported evidence item information		
05_04	Evidence traceability link import		
05_05	Automatic artefact association to imported evidence traceability link		
05_06	Tool information about the artefact associated to imported evidence traceability link		
05_07	Evidence traceability link export		
05_08	SAEM import		
05_09	SAEM export		
05_10	Automatic update of evidence items in the OPENCOSS platform		
05_11	Automatic update of evidence items in external tools		
05_12	Notification of change of imported evidence item in the OPENCOSS platform		
05_13	Notification of change of imported evidence item in external tool		
05_14	Highlight for change of imported evidence item in the OPENCOSS platform		
05_15	Highlight for change of imported evidence item in external tool		
05_16	Automatic update of evidence traceability links in the OPENCOSS platform		
05_17	Automatic update of evidence traceability links in external tools		
05_18	Notification of change of imported evidence traceability link in the OPENCOSS platform		
05_19	Notification of change of imported evidence traceability link in external tool		
05_20	Highlight for change of imported evidence traceability links in the OPENCOSS platform		
05_21	Highlight for change of imported evidence traceability link in external tools		