Contributors

<table>
<thead>
<tr>
<th>Names</th>
<th>Organisation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alejandra Ruiz, María Carmen Palacios, Idoya del Río, Ángel López, Huáscar Espinoza</td>
<td>TECNALIA</td>
</tr>
<tr>
<td>Janusz Studzizba</td>
<td>PARASOFT</td>
</tr>
<tr>
<td>Yaping Luo, Alexander Mois Aroyo</td>
<td>Eindhoven University of Technology</td>
</tr>
<tr>
<td>Thomas Richardson</td>
<td>University of York</td>
</tr>
</tbody>
</table>

Document History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>V0.5</td>
<td>2013-07-19</td>
<td>Initial Version</td>
</tr>
<tr>
<td>V0.6</td>
<td>2013-08-07</td>
<td>Added: Mapping between RefFrameworks &amp; Process Management (Idoya and Huáscar)</td>
</tr>
<tr>
<td>V0.61</td>
<td>2013-09-02</td>
<td>Edition of Assurance Project and Baseline improved, as well as a new section for Compliance Map edition</td>
</tr>
<tr>
<td>V 0.7</td>
<td>2013-10-21</td>
<td>Changes to the version 0.6 of the First Prototype</td>
</tr>
<tr>
<td>V 0.8</td>
<td>2014-06-01</td>
<td>Update to describe the second prototype functionalities (Server and Client parts)</td>
</tr>
<tr>
<td>V 0.9</td>
<td>2014-09-10</td>
<td>Update to add the mapping management</td>
</tr>
<tr>
<td>V 1.0</td>
<td>2014-12-11</td>
<td>Described functionality added on client and server in 3rd prototype</td>
</tr>
</tbody>
</table>
# TABLE OF CONTENTS

1 Executive Summary........................................................................................................... 12

2 OPENCOSS platform - basic concepts .................................................................................. 13
   2.1 Client-Server architecture ............................................................................................. 13
   2.2 Naming conventions in OPENCOSS client ...................................................................... 13

3 Installation of OPENCOSS server and clients .................................................................... 15
   3.1 Installation of OPENCOSS platform server ................................................................... 15
      3.1.1 Installation of PostgreSQL database .......................................................................... 15
      3.1.1.1 Installation of PostgreSQL database on Windows machine ................................. 15
      3.1.2 Creating OPENCOSS database in PostreSQL .............................................................. 17
      3.1.3 Installation of OPENCOSS server distribution ........................................................... 18
   3.2 Installation of OPENCOSS platform client .................................................................... 20
      3.2.1 Client bundle download ........................................................................................... 20
      3.2.2 Client configuration .................................................................................................. 21
      3.2.3 Deleting Repository contents .................................................................................... 24

4 Prescriptive Knowledge Management .................................................................................. 25
   4.1 Create Reference Framework model ............................................................................. 26
   4.2 How to edit a Reference Framework model .................................................................... 29
      4.2.1 Add concepts to the diagram ................................................................................... 29
      4.2.2 Add links between concepts .................................................................................... 29
      4.2.3 Edit properties ......................................................................................................... 30
      4.2.4 Create multi-diagrams from a Reference Framework model ..................................... 30
      4.2.5 Non graphical editor .............................................................................................. 32
   4.3 Creating Equivalence Maps ........................................................................................... 33
      4.3.1 Equivalence Map using the editor ............................................................................ 33
      4.3.2 Equivalence Map using a tailored functionality ....................................................... 37
   4.4 Creating Applicability Tables ......................................................................................... 42

5 Assurance Project Management ........................................................................................... 46
   5.1 Create Assurance Project and Baseline .......................................................................... 47
   5.2 Create or update Project Baseline .................................................................................. 51
   5.3 Edit Project Baseline ...................................................................................................... 54
   5.4 Edit Compliance Maps .................................................................................................. 55
      5.4.1 Compliance Map using the editor ............................................................................. 55
      5.4.2 Compliance Map using a tailored functionality ......................................................... 60
   5.5 Cross-Domain reuse ...................................................................................................... 62
   5.6 Cross-Project reuse ........................................................................................................ 67

6 Safety Argumentation Management ...................................................................................... 71
   6.1 Preferences ................................................................................................................... 71
   6.2 Creating and Saving a Diagram ...................................................................................... 71
      6.2.1 Creating a New Diagram .......................................................................................... 71
      6.2.2 Creating a Diagram at the project creation time ...................................................... 74
      6.2.3 Opening a Diagram ................................................................................................. 74
      6.2.4 Saving a Diagram .................................................................................................... 75
   6.3 Editing Functions ............................................................................................................ 75
      6.3.1 Editing a Diagram .................................................................................................... 75
6.3.1.1 Copying and Pasting an Element ................................................. 79
6.3.1.2 Deleting a Node or a Link .......................................................... 79
6.3.2 Create multi-diagrams from an Argumentation model .................... 79
6.3.3 Editing a Diagram Using a Pattern or a Module .............................. 82
   6.3.3.1 Creating a New Pattern Diagram ............................................ 83
   6.3.3.2 Editing a Pattern Diagram .................................................... 83
   6.3.3.3 Adding Elements from Patterns to a Diagram (instantiating a Pattern) ... 85
   6.3.3.4 Creating a New Module Diagram .......................................... 85
   6.3.3.5 Editing a Module Diagram ................................................... 85
   6.3.3.6 Adding Elements from Modules to a Diagram (instantiating a Module) . 86

6.3.4 Connecting a Diagram to Artefacts .............................................. 87
6.4 Argumentation Contracts ................................................................. 89
   6.4.1 Preparation for Contracts .......................................................... 89
   6.4.2 Creating a Contract .................................................................. 92
6.5 Argumentation Integration ................................................................. 94
6.6 Argumentation Change Management .................................................. 96
6.7 Vocabulary ....................................................................................... 97
   6.7.1 Defining Vocabularies ............................................................... 97
6.8 Using Vocabularies in the Argument Editor ........................................ 99
6.9 Pattern Instantiation ......................................................................... 100
6.10 Structured Sentences ....................................................................... 101
6.11 Printing .......................................................................................... 103

7 Evidence Management .......................................................................... 104
   7.1 Define Artefact Repository Preferences ........................................... 104
   7.2 Artefact Definition .......................................................................... 108
      7.2.1 Add an artefact definition ......................................................... 108
      7.2.2 Delete Artefact Definition ....................................................... 110
   7.3 Artefact ......................................................................................... 111
      7.3.1 Add an artefact .................................................................. 111
      7.3.2 Delete an artefact ................................................................ 115
   7.4 Artefact Resource .......................................................................... 116
      7.4.1 Add an artefact resource to an artefact .................................. 116
      7.4.2 Delete an artefact resource .................................................... 119
   7.5 Artefact Property Value .................................................................. 120
      7.5.1 Add an artefact property value to an artefact .......................... 121
      7.5.2 Delete an artefact property value ........................................... 123
   7.6 Artefact Assurance Asset Evaluation .............................................. 124
      7.6.1 Add an artefact assurance asset evaluation to an artefact ........... 124
      7.6.2 Delete an artefact assurance asset evaluation .............................. 126
   7.7 Artefact Assurance Asset Events ..................................................... 127
      7.7.1 Add an artefact assurance asset event to an artefact ................. 127
      7.7.2 Delete an artefact assurance asset event .................................... 129
   7.8 Impact analysis ............................................................................... 130

8 Process Assurance Management ............................................................ 133
   8.1 Creating Process Assurance data ..................................................... 134
   8.2 Deleting Process Assurance Objects ............................................... 136

9 Properties Management ........................................................................ 138
   9.1 Property ....................................................................................... 140
      9.1.1 Add a property ................................................................... 140
      9.1.2 Delete a property .................................................................. 142
10  Mapping Management......................................................................................................................... 144
10.1  Map Group............................................................................................................................................ 146
  10.1.1  Add a map group........................................................................................................................... 146
  10.1.2  Delete a map group....................................................................................................................... 147
10.2  Map....................................................................................................................................................... 148
  10.2.1  Add a map....................................................................................................................................... 148
  10.2.2  Delete a map................................................................................................................................... 150
11  OPENCOSS server.................................................................................................................................... 152
11.1  OPENCOSS Web interface layout........................................................................................................ 152
11.2  Compliance report ................................................................................................................................. 153
  11.2.1  Goal of the report .......................................................................................................................... 153
  11.2.2  Viewing compliance data on the report ....................................................................................... 153
  11.2.3  Adding evidence and compliance data ....................................................................................... 156
  11.2.4  Generation of summary textual report ....................................................................................... 158
11.3  Compliance Estimation report ............................................................................................................. 159
11.4  Change Impact Analysis....................................................................................................................... 161
  11.4.1  Change Impact Analysis in OPENCOSS Client ........................................................................... 162
  11.4.2  Change Impact Analysis algorithm ............................................................................................. 162
  11.4.3  Impact Analysis result presentation on OPENCOSS server reports ........................................... 163
11.5  Gap Analysis report - Compliance Assessment and Evidence Evaluation ........................................ 164
  11.5.1  Gap Analysis report core functionality ....................................................................................... 165
  11.5.2  Viewing Evidence Evaluation in Gap Analysis report ............................................................... 167
11.6  Metrics reports..................................................................................................................................... 167
  11.6.1  Metrics Estimation Report .......................................................................................................... 168
  11.6.2  Equivalence Map Report ............................................................................................................ 169
11.7  Administration web GUI....................................................................................................................... 170
  11.7.1  Projects Administration .............................................................................................................. 170
  11.7.2  Create Sample Data .................................................................................................................... 171
  11.7.3  Configuration Settings.................................................................................................................. 172
List of Figures

Figure 1 - pgAdmin application .................................................................................................................. 16
Figure 2 - OPENCOSS platform main page ............................................................................................... 20
Figure 3 - Select the workspace menu ...................................................................................................... 21
Figure 4 - Preference menu ......................................................................................................................... 21
Figure 5 - Model Repository Configuration Page linked to the TUE repository ....................................... 22
Figure 6 - Show view menu ......................................................................................................................... 23
Figure 7 - Repository Explorer View ......................................................................................................... 23
Figure 8 - Repository Explorer content with configuration error ............................................................. 24
Figure 10 - Delete model menu .................................................................................................................. 24
Figure 11 - Reference Framework concepts ............................................................................................... 25
Figure 12 - New Reference Framework model ........................................................................................... 26
Figure 13 - Wizard Reference Framework model ....................................................................................... 26
Figure 14 - New Refframework Diagram .................................................................................................. 27
Figure 15 - New Refframework Domain Model ......................................................................................... 28
Figure 16 - Refframework editor perspective ............................................................................................ 29
Figure 17 - Show properties view .............................................................................................................. 30
Figure 18 - Deleted concept shown in a diagram ....................................................................................... 30
Figure 19 - Refframework Diagram wizard I .............................................................................................. 31
Figure 20 - Refframework Diagram wizard II ............................................................................................. 31
Figure 21 - Refframework Diagram wizard III ........................................................................................... 32
Figure 22 - Model tree editor ...................................................................................................................... 32
Figure 23 - Edit model from Outline .......................................................................................................... 33
Figure 24 - Load Resource I ....................................................................................................................... 34
Figure 27 - Activity Equivalence Map ....................................................................................................... 36
Figure 28 - Equivalence Map ....................................................................................................................... 37
Figure 29 - How to create Equivalence Map ............................................................................................... 38
Figure 30 - Equivalence Map form ............................................................................................................ 39
Figure 32 - Equivalence Map with Postcondition ..................................................................................... 41
Figure 34 - Applicability Table ISO 26262 .................................................................................................. 42
Figure 35 - Requirement Applicability Table ISO 26262 .......................................................................... 43
Figure 36 - Applicability Table DO-178C .................................................................................................. 43
Figure 37 - Activity Applicability Table DO-178C ...................................................................................... 44
Figure 38 - Requirement Applicability Table DO-178C .......................................................................... 44
Figure 39 - Summary Applicability Table DO-178C ................................................................................ 45
Figure 40 - Assurance Project structure .................................................................................................... 46
Figure 41 - New Assurance Project wizard ............................................................................................... 47
Figure 42 - Assurance Project name page .................................................................................................. 48
Figure 43 - Reframework selection ........................................................................................................... 49
Figure 44 - Assurance Project structure .................................................................................................... 50
Figure 45 - Assurance Project editor .......................................................................................................... 51
Figure 46 - Other kind of projects option ................................................................................................... 51
Figure 47 - Creates or Updates Baseline wizard ....................................................................................... 52
Figure 48 - Selection of the Assurance Project to update ......................................................................... 53
Figure 49 - Assurance Project with new baseline .................................................................................... 54
Figure 50 - Baseline editor ......................................................................................................................... 54
Figure 51 - Baseline graphical editor .......................................................................................................... 55
Figure 52 - Assets Package active .............................................................................................................. 56
Figure 53 - Baseline Config active ............................................................................................................ 57
Figure 106 "To Be Supported" and "Public" Property Values for Declaring a Claim Providing Support to Other Modules ................................................................. 91
Figure 107 New Contract Wizard .............................................................................................................. 92
Figure 108 Selecting an Inter-Module Dependency .................................................................................. 93
Figure 109 Selecting Elements to Resolve the Dependency ................................................................. 93
Figure 110 Justification of Dependency Resolution ................................................................................ 94
Figure 111 Argumentation Integration Wizard ....................................................................................... 95
Figure 113 Select Contracts that Resolve Inter-Module Dependencies of Modules Previously Selected .... 96
Figure 114 Argumentation Change Analysis ............................................................................................. 96
Figure 115 Select Argumentation Integration Model to Analyse ............................................................ 97
Figure 117 – Example Vocabulary Diagram ............................................................................................ 98
Figure 118 – Vocabulary Import .............................................................................................................. 98
Figure 119 – Mark-up Rendering ............................................................................................................. 99
Figure 120 – Tooltip ................................................................................................................................. 99
Figure 121 – Term Suggestions ................................................................................................................. 100
Figure 122 – Pattern Instantiation ........................................................................................................ 100
Figure 123 – New Claim Types Editor .................................................................................................... 101
Figure 124 – Claim Type Repository ....................................................................................................... 102
Figure 125 – Claim Type Tab in the Properties View .............................................................................. 103
Figure 126 - Preference menu.................................................................................................................. 104
Figure 127 - Artefact Repository Preferences ......................................................................................... 105
Figure 128 - New Evidence Model menu File -> New -> Other ............................................................ 105
Figure 129 - New Evidence Model I ...................................................................................................... 106
Figure 130 - New Evidence Model II ..................................................................................................... 106
Figure 131 - New Evidence Model III .................................................................................................... 107
Figure 132 - Evidence Model .................................................................................................................. 107
Figure 133 - Add New Artefact Definition (I) ......................................................................................... 108
Figure 134 - Add New Artefact Definition (II) ....................................................................................... 108
Figure 135 - Artefact Definition Description (I) .................................................................................... 109
Figure 136 - Artefact Definition Description (II) ................................................................................... 109
Figure 137 - Description Artefact Definition Artefact .......................................................................... 110
Figure 138 - Description Artefact Definition Evaluation ........................................................................ 110
Figure 139 - Description Artefact Definition Events ............................................................................ 110
Figure 140 - Delete Artefact Definition (I) ............................................................................................ 111
Figure 141 - Delete Artefact Definition (II) ........................................................................................... 111
Figure 142 - Add New Artefact (I) ......................................................................................................... 112
Figure 143 - Add New Artefact (II) ........................................................................................................ 112
Figure 144 - Artefact Description ........................................................................................................... 113
Figure 145 - Description Artefact Version ............................................................................................. 114
Figure 146 - Description Artefact Property Value .................................................................................. 114
Figure 147 - Description Artefact Evaluation ........................................................................................ 115
Figure 148 - Description Artefact Events ............................................................................................... 115
Figure 149 - Delete Artefact I ................................................................................................................. 115
Figure 150 - Delete Artefact II ............................................................................................................... 116
Figure 151 - Add Artefact Resource I .................................................................................................... 116
Figure 152 - Resource properties ........................................................................................................... 117
Figure 153 - Add Artefact Resource II .................................................................................................. 117
Figure 154 - Resource dialog box ........................................................................................................... 118
Figure 155 - Select Artefact from the local drive .................................................................................... 118
Figure 156 - Select Artefact from the SVN Remote Repository .............................................................. 119
Figure 157 - SVN History table of a File ............................................................................................... 119
Figure 158 - Delete Artefact Resource I ................................................................. 119
Figure 159 - Delete Artefact Resource II .............................................................. 120
Figure 160 - Load Resource Property model .......................................................... 121
Figure 161 - Select Property model .......................................................... 121
Figure 162 - Add Artefact Property Value I .......................................................... 122
Figure 163 - Artefact Value dialog box .......................................................... 122
Figure 164 - Add Artefact Property Value II ......................................................... 123
Figure 165 - Artefact Property properties ........................................................... 123
Figure 166 - Delete Artefact Property Value I ..................................................... 124
Figure 167 - Delete Artefact Property Value II ..................................................... 124
Figure 168 - Add Artefact Assurance Asset Evaluation I ........................................ 125
Figure 169 - Artefact Assurance Asset Evaluation dialog box .................................. 125
Figure 170 - Add Artefact Assurance Asset Evaluation II ...................................... 126
Figure 171 - Artefact Assurance Asset Evaluation properties ................................. 126
Figure 172 - Delete Artefact Assurance Asset Evaluation I .................................... 127
Figure 173 - Delete Artefact Assurance Asset Evaluation II ................................... 127
Figure 174 - Add Artefact Assurance Asset Event I ............................................. 128
Figure 175 - Artefact Assurance Asset Event dialog box ....................................... 128
Figure 176 - Add Artefact Assurance Asset Event II ............................................ 129
Figure 177 - Artefact Assurance Asset Event properties ....................................... 129
Figure 178 - Delete Artefact Assurance Asset Event I .......................................... 130
Figure 179 - Delete Artefact Assurance Asset Event II ......................................... 130
Figure 180 - Artefact modified with automatically generated events ..................... 131
Figure 181 - Artefact analyser confirmation windows ......................................... 131
Figure 182 - Artefact events created by Impact Analyser ..................................... 132
Figure 183 - New Process Model I ...................................................................... 133
Figure 184 - New Process Model II ..................................................................... 133
Figure 185 - New Process Model III .................................................................... 134
Figure 186 - Process Model .............................................................................. 134
Figure 187 - Create Process Model data using context menu ................................. 135
Figure 188 - Create Process Model data using properties View .............................. 135
Figure 189 - Delete Process Model data using context menu ............................... 136
Figure 190 - Delete Process Model data using properties view ............................. 137
Figure 191 - New Property Model menu File -> New -> Other ................................ 138
Figure 192 - New Property Model I .................................................................... 138
Figure 193 - New Property Model II ................................................................... 139
Figure 194 - New Property Model III .................................................................. 139
Figure 195 - Property Model ............................................................................. 140
Figure 196 - Add New Property (I) ..................................................................... 140
Figure 197 - Add New Property (II) .................................................................. 141
Figure 198 - Property properties ....................................................................... 141
Figure 199 - Add enum values .......................................................................... 142
Figure 200 - Delete Property I .......................................................................... 142
Figure 201 - Delete Property II .......................................................................... 143
Figure 202 - New Property Model menu File -> New -> Other .............................. 144
Figure 203 - New Mapping Model I ................................................................. 144
Figure 204 - New Mapping Model II .................................................................. 145
Figure 205 - New Mapping Model III ............................................................... 145
Figure 206 - Mapping Model ............................................................................ 146
Figure 207 - Add New Map Group (I) ................................................................. 146
Figure 208 - Add New Map Group (II) ............................................................... 147
List of Tables
1 Executive Summary

This document is a user manual of the OPENCOSS platform tools prototype implementation. In this document the user can find the installing instructions, the tool environment description, and the functionalities starting from the creation of Reference Frameworks (models representing Standards, Regulations, or Company-specific Processes), creation of Assurance Projects and the associated Baseline (subset of Reference Framework to be applied in a specific assurance project), Evidences model (Artefacts), Process model (Activities) Compliance Maps (so far, compliance maps from Reference Artefacts to Artefacts), Argumentation model and web interface reports.

Finally, functionality facilitated by OPENCOSS web UI server has been described.

This document has been elaborated as a Fast User Manual. Further questions must be directed to the TECNALIA team.
2 OPENCOSS platform - basic concepts

2.1 Client-Server architecture

OPENCOSS platform tools are designed to follow client-server architecture approach.

OPENCOSS platform deployment model consists of:

- **OPENCOSS server** - installed in a central host machine
  
  *Note:* Functionality facilitated by OPENCOSS server has been described in “OPENCOSS server” chapter.

- One or many **OPENCOSS clients** - each of which installed on specific user machines
  
  *Note:* Functionality facilitated by OPENCOSS clients has been described in several chapters: from “Prescriptive Knowledge Management” to “Mapping Management”.

2.2 Naming conventions in OPENCOSS client

In this document, the naming convention follows the following concepts:

- **Environment** supports (a large part of) the software tool process. OPENCOSS tool platform is the main environment in this document.

- **Workbenches** support only one or a few activities. Example: “Evidence Management” workbench.

- **Tool** support only specific tasks in the software tool process. Example: “Evidence Analysis”.

The OPENCOSS tool platform has the following Workbenches:

<table>
<thead>
<tr>
<th>Workbench</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prescriptive Knowledge Management</td>
<td>Functionality related to the management (edition, search, transfer, etc.) of standards information as well as any other information derived from them, such as interpretations about intents, mapping between standards, etc. This functional group maintain a knowledge database about “reference frameworks” (model of standards, regulations, etc.), which can be used by other OPENCOSS functionalities.</td>
</tr>
<tr>
<td>Assurance Project Lifecycle Management</td>
<td>This functionality factorizes aspects such as the creation of safety assurance projects locally in OPENCOSS and any project baseline information that may be shared by the different functional modules. A project baseline is a subset of reference framework (e.g., subset of a standard) that will be applied to a given assurance project.</td>
</tr>
<tr>
<td>Safety Argumentation Management</td>
<td>This group manages argumentation information in a modular fashion. It also includes mechanisms to support compositional safety assurance, and assurance patterns management.</td>
</tr>
<tr>
<td>Evidence Management</td>
<td>This module manages the full life-cycle of evidences and evidence chains. This includes evidence traceability management and impact analysis. In addition, this module is in charge of communicating with external engineering tools</td>
</tr>
<tr>
<td>Process Assurance Management</td>
<td>This functionality group handles every activity related to the specification, execution and validation of safety assurance processes in connection with engineering processes. It also manages compliance information related to functional safety standards. This module should be integrated with process-related tools managed by companies (ALM/PLMs, process workflows, etc.)</td>
</tr>
</tbody>
</table>

*Table 1 - Workbenches OPENCOSS tool platform*
3 Installation of OPENCOSS server and clients

As it has already been introduced, OPENCOSS platform tools are designed in client-server architecture approach.

The following chapters present how to install server on a central machine and clients on user machines respectively.

3.1 Installation of OPENCOSS platform server

OPENCOSS platform server should be installed on a corporate central server host, which is accessible from company user machines.

The OPENCOSS server infrastructure consists of the following main components:

- PostgreSQL database - which stores OPENCOSS data
- Apache Tomcat web server - which provides OPENCOSS server services
- OPENCOSS web applications deployed on the web server

Follow the steps below in order to install the OPENCOSS server infrastructure on a central host.

3.1.1 Installation of PostgreSQL database

OPENCOSS platform tools use PostgreSQL database.

Download it from PostgreSQL official website http://www.postgresql.org/download/.

Follow the PostgreSQL installation instructions to install and setup it.

For user convenience, PostgreSQL installation on Windows has been described in the chapter below.

3.1.1.1 Installation of PostgreSQL database on Windows machine

1. Download PostgreSQL installer
   a. Go to the download section for Windows http://www.postgresql.org/download/windows/
   b. Click on download installer from EnterpriseDB
   c. Choose the latest version and download it
2. Double click the installer file and follow the installation wizard.
3. Most important steps during the installation process are (among others):
   a. Definition of a password for the database super-user (his login is “postgres”).
   b. Definition of a port for PostgreSQL - the default 5432 is recommended.
   c. If the installation wizard asks you to launch Stack Builder to install additional tools, you may skip this step - no additional tools are needed.
   d. The installation may take a few minutes to complete.
4. Verify the database installation
   a. The quick way to verify the installation is to use pgAdmin application which has been installed together with PostgreSQL server.
      Please run “pgAdmin III”. The following GUI is displayed:
b. In the left panel, double click on PostgreSQL 9.3 tree node. The application will ask to enter a database super-user password. Please enter the password which has been defined during PostgreSQL server installation.

c. When database objects are displayed, your database server has been installed correctly.

d. You can now define login role with non-super-user rights. This account will be used to login to the database later on, as it is not advisable to work on a daily basis with super-privileged user account. In this example we use role “opencossdbms” with password “opencossdbms”. You can use pgAdmin to do so, or bring up the PostgreSQL textual console (psql) and type:

```bash
> create role opencossdbms with login createdb encrypted password 'opencossdbms';
```
3.1.2 Creating OPENCOSS database in PostreSQL

This step demonstrates how to create a database in PostreSQL which will store data tables used by OPENCOSS platform tools. In “pgAdmin III” application right-click Databases tree node and choose “New Database...” option.

In the dialog which opens, please provide the database name as “cdo-opencoss” and set “opencossdbms” as its owner. The database will be created.

Alternatively, when “pgAdmin III” application is not available, database can be created using “psql” command line. In order to do it, please run “psql” application and issue the following command:

```bash
> CREATE DATABASE "cdo-opencoss" with owner "opencossdbms";
```

Note:
“cdo-opencoss” is a default database name which is accessed by OPENCOSS server. This can be reconfigured later in OPENCOSS server configuration file (<entry key="dbName">). For details about the server configuration file, see Installation of OPENCOSS server distribution.

In “pgAdmin III” application right-click Schemas (for cdo-opencoss db) tree node and choose “New schema...” option.
Create schema name: **externaltools** with owner **opencossdbms**. The schema will be created.

Alternatively, when “pgAdmin III” application is not available, schema can be created using “psql” command line. In order to do it, please run “psql” application and issue the following command:

```
> CREATE SCHEMA externaltools AUTHORIZATION opencossdbms;
```

Then in pgAdmin application (or psql command line) execute query:

```
ALTER DATABASE "cdo-opencoss" set search_path=externaltools,public
```

### 3.1.3 Installation of OPENCOSS server distribution

1. Download and unpack OPENCOSS Server distribution.
   The latest build can be downloaded from:

   ![Download Link](http://77.252.162.49:8080/opencoss/server/OpencossServer.zip)

   The downloaded OPENCOSS server distribution consists of Apache Tomcat web server with deployed OPENCOSS server web applications.

   Please unzip it to some target installation folder. The folder will be referred to as [OPENCOSS_DIR] in this document.

   **Note for developers:**
   The distribution contains `!!!README.OPENCOSS!!!` file which describes what files are added to the default Apache Tomcat application.

2. OPENCOSS server requires Java JRE version 8. If it is not present on the machine, please install it. It can be downloaded from:
3. Configure Tomcat to use Java JRE 8 installed on your computer.
   In order to do this, please navigate to [OPENCOSS_DIR]/bin folder, and edit catalina.bat file (on Windows) or catalina.sh (on Linux).
   Scroll down in the code until you pass the end of the beginning “rem” comments, and adjust the following code pointing to your Java JRE root directory:

   - for example on Windows:
     ```bash
     set JRE_HOME=C:\Program Files\Java\jre8
     ```
   - for example on Linux:
     ```bash
     export JRE_HOME=/usr/java/jdk1.8
     ```

   Save your changes.

4. Adjust OPENCOSS server configuration file settings.

   Go to [OPENCOSS_DIR]/conf-opencoss folder and move opencoss-properties.xml file from this location to the operating system user home directory. This is the location from where OPENCOSS server reads opencoss-properties.xml settings file.
   The location, depending on the operating system, would be:
   - Windows XP: c:\Documents and Settings\<username>\opencoss-properties.xml
   - Windows 7/8: c:\Users\<username>\opencoss-properties.xml
   - Linux: /home/<username>/opencoss-properties.xml

   Edit opencoss-properties.xml settings file.
   The most important entries in this file are:
   - “dbUser” / “dbPassword”
     These are PostgreSQL user credentials. Please specify a valid user and password for your PostgreSQL server.
   - “serverAddress”
     This is CDO repository name which is broadcasted by the CDO server. The “localhost” value should be replaced with the specific server machine host name in order OPENCOSS tool clients are able to connect to this server repository from other hosts. Please modify the following entry:
     ```xml
     <entry key="serverAddress">localhost:2036</entry>
     ```
     by replacing “localhost” with the specific server host name, e.g.:
     ```xml
     <entry key="serverAddress">host-name.acme.com:2036</entry>
     ```

5. Start OPENCOSS server by going to [OPENCOSS_DIR]/bin folder and running startup.bat (Windows) or startup.sh (Linux).

6. Verify OPENCOSS platform web application

   OPENCOSS server web pages are served at 8080 port.
   Please run your web browser and go to the following location:
   http://<OPENCOSS-SERVER-HOST-NAME>:8080/
in particular when you are on the server machine:

http://localhost:8080/

The OPENCOSS platform server page should be displayed, presenting one of web reports.

![Image of OPENCOSS platform main page]

Figure 2 - OPENCOSS platform main page

Now some data should be entered to OPENCOSS server using OPENCOSS client tools, in order it can be visualized by web reports.

**NOTE:**
At this stage of installation process, OPENCOSS database tables have not been created yet.
In order to use OPENCOSS web reports, all database tables need to be created.
To do this, please install the OPENCOSS Eclipse client, as described in the subsequent chapter, connect to the CDO server and create some new Assurance Project.

### 3.2 Installation of OPENCOSS platform client

#### 3.2.1 Client bundle download

It is required to have installed (minimum) **Java Environment 1.6**.

To install the OPENCOSS platform client download it using one of the links below according to your OS Platform and uncompress it into your hard disk.

http://77.252.162.49:8080/opencoss/client/20141219_OpencossClient_Win_64.zip

http://77.252.162.49:8080/opencoss/client/20141219_OpencossClient_Win_32.zip

http://77.252.162.49:8080/opencoss/client/20141219_OpencossClient_Linux_64.tar.gz
3.2.2 Client configuration

To use the platform execute the eclipse.exe file and introduce a select a folder that will be used as workspace.

![Select workspace](image)

*Figure 3 - Select the workspace menu*

The first step after the installation process is to configure the connection settings with the CDO repository where all the models generated using the platform will be stored. This information must be introduced in the Model Repository Preference page inside the Opencoss category. Go to menu Window → Preferences to open this window.

![Preference menu](image)

*Figure 4 - Preference menu*
The information to introduce is:

- **Server IP**: The IP of the running CDO Server.
- **Server Port**: The port used by the running CDO Server.
- **Repository name**: The name of the repository where all the data will be stored (read only).
- **Protocol**: The protocol used to connect to the CDO Server.

After introducing this data the Repository Explorer View can be used inside the Opencoss category. The objective of this view is showing the contents of the repository configured. To open it go to menu Window → Show View → Other.
If the provided connection settings with the repository are incorrect or the server is not running, this view will display the error in the screenshot below instead of the contents of the repository. To solve it, check the server is running and the configuration settings are correct, close the view and open it again.
After that, if the server is not installed in the same machine, copy opencoss-properties.xml from server to your local drive. The destination path, depending on the operating system, should be:

- Windows XP: `c:\Documents and Settings\<username>\opencoss-properties.xml`
- Windows 7/8: `c:\Users\<username>\opencoss-properties.xml`
- Linux: `/home/<username>/opencoss-properties.xml`

**3.2.3 Deleting Repository contents**

To delete a folder and its contents, right click over it and left click the “DELETE Folder” menu.

To delete a model, right click over it and left click the “DELETE Model” menu.
4 Prescriptive Knowledge Management

The current version of the OPENCOSS prototype only covers Edition of the Reference Framework models. It does not cover CCL Vocabulary specification.

Users can use the Reference Framework Editor to model Standards (IEC 61508, ISO 26262, DO-178C, EN 50126, and the like), any Regulations (either as additional Requirements or model elements in a given model representing a Standard or a new Reference Framework), and Company-specific processes (e.g., the Alstom, Thales or Fiat process to develop safety-critical systems).

Each Reference Framework model can be also mapped to other Reference Framework models by using the concept of Equivalence Map.

Finally, Reference Frameworks can be used to create Assurance Project Baselines. Baseline Models represent the subset of the Reference Frameworks tailored for individual Assurance Projects. For more details on Baselines, please see the Assurance Project management Section.

Figure 11 - Reference Framework concepts
4.1 Create Reference Framework model

In order to create a new Reference Framework model, follow the next steps:

- From the File menu, choose New -> Other ...

![Figure 12 - New Reference Framework model](image)

- In the Wizard dialog, open the Opencoss category, and select Refframework Diagram, and press the Next button.

![Figure 13 - Wizard Reference Framework model](image)

- In the New Refframework Diagram dialog, select or enter the parent folder, the name of the diagram to be created, and press the Next button.
In the New Refframework Domain model page, select or enter the same name as in the previous step as parent folder, enter the name of the diagram to be created, and press the Finish button.

Figure 14 - New Refframework Diagram
Figure 15 - New Refframework Domain Model
4.2 How to edit a Reference Framework model

After completing the Refframework Diagram creation wizard, the perspective of the tool will be opened composed by five views:

1. The Repository Explorer shows the contents of the repository.
2. The Outline shows the elements of the model and permits their edition.
3. The Diagram Editor permits the graphical modelling of a subset of concepts of the Reference Framework.
4. The Palette is a toolbox with the concepts of the model and the connections between them to add to the diagram.
5. The Properties to edit the properties of the element of the model selected.

![Figure 16 - Refframework editor perspective](image)

4.2.1 Add concepts to the diagram

To add concepts to the diagram, left click in a category Object of the palette and move the cursor over the diagram zone. This cursor appears if it is possible to add this object in the target diagram location according to the modelling rules (Reference Framework metamodel), if not this other will appear. A figure representing the concept will be displayed in the diagram.

4.2.2 Add links between concepts

To add link between concepts, select it from the Connections category of the palette. This cursor appears if this object can be the origin of the connection, according to the modelling rules (Reference Framework metamodel).
Framework metamodel), if not this other will appear. Maintain the left mouse clicked, the cursor will become , and move to the destination object, the same icons will appear if the destination is correct or not.

### 4.2.3 Edit properties

Some model elements from Reference Framework cannot be edited graphically (RefRequirements, RefApplicability tables, among others). These model elements can be edited by using the Properties view.

If the properties view is not visible, you can open it by using the contextual menu of the figures “Show Properties View” of the figures.

![Properties View](image)

*Figure 17 - Show properties view*

### 4.2.4 Create multi-diagrams from a Reference Framework model

The tool allows managing different views of a model through a set of diagrams. Once a model is available, a new diagram view can be created and special edition functionalities are available as follows:

1. Thanks to the Outline view, it is possible to drag and drop concepts from the model to the diagram.
2. Once a concept has been selected, it can be hidden through the “Delete from diagram” option available in the contextual menu. This option does not delete the concept from the model.
3. Once a concept has been selected, it can be deleted through the “Delete from model” option available in the contextual menu. This option delete the concept from the model permanently. If this deleted concept is visible in another diagram files, this concepts will be shown with a cross icon in the upper right corner to show that it does not exist anymore.

![Deleted Concept](image)

*Figure 18 - Deleted concept shown in a diagram*
Once a model is available, a new diagram view can be created following the procedure below.

Select “New” → “Other…” from the File menu of eclipse

Select “Refframework Diagram” in the Opencoss category and click the “Next” button.

Select the folder to store the new diagram. Then type the name of the diagram to be created and click the “Next” button.

Type the name of the model created previously and click the “Finish” button.

Figure 19 - Refframework Diagram wizard I

Figure 20 - Refframework Diagram wizard II
After that, the diagram is ready for edition.

4.2.5 Non graphical editor

Alternatively to the graphical Editor, the Reference Framework model can be edited by using a purely Form Editor. To do so, open the file created together with the Diagram file (extension: `xxx.refframework`).
It is also possible to use the Outline view to create new model elements, as shown below.

![Figure 23 - Edit model from Outline](image)

### 4.3 Creating Equivalence Maps

It's possible to create Equivalence Maps in two ways:
- One way, using the editor,
- Another way, using a tailored functionality for it.

#### 4.3.1 Equivalence Map using the editor.

To create Equivalence Maps using the editor, it’s necessary to load two CDO resources: the reference framework model (.refframework) and the mapping model (.mapping).

So, press the editing window and select “Load Resource” in the context menu.
Then select the refframework model and mapping model using the “Browse Repository” button to obtain the URI of any model stored in the repository.
Figure 25 - Load Resource Reference Framework II
It’s possible to create equivalence maps for activities, artefacts, requirements, roles and techniques. Then, first select the object in the tree and after click on the tab “Activity Equivalence Map” and press the button “Add”.

Figure 26 - Load Resource Map Group III

Figure 27 - Activity Equivalence Map
Finally, enter the information requested:

![Equivalence Map](image)

*Figure 28 - Equivalence Map*

If the user adds as target of the equivalence an element of the source refframework, this target element will be considered as postCondition. The postConditions are mandatory extra activities, not included in the standard, that must be performed in case of reusing the target element from one assurance project based in the target refframework in another assurance project based in the source refframework using the Cross-Domain functionality that will be explained in the section put section reference.

### 4.3.2 Equivalence Map using a tailored functionality.

To create Equivalence Maps using the tailored functionality, first of all, it’s necessary to press the button “Mapping Set” on the properties form of the reference framework using the tree view editor (not available using the diagram editor). This window automatically saves the mappings when checking or unchecking elements of the target refframework tree.
Figure 29 - How to create Equivalence Map.
The Equivalence Map form is organized in three zones:

- **The left zone** shows the actual reference framework, and it loads the type of elements for which we want to make the equivalence maps. For default, activities.

- **The middle zone** allows to make different filters like:
  
  o **Filter Mapping Model** lists all the mapping models stored in the database, and it will be necessary to select one of them and one group model. It’s also possible to create a new map group pressing the button “New group”.

  o **Filter Map Element**. It’s possible to create equivalence maps for activities, artefacts, requirements, roles and techniques. When these filter change, also it changes the information showed by the reference framework. For example:

    ▪ If the filter “Requirement” is selected only the requirements of both refframeworks will be shown:
Figure 31 - Equivalence Map, select map element (I)

- *Filter Equivalence Map*. This filter allows making different equivalence maps for the same refinement framework element.
- The mapping information must also be introduced in the middle part by the user; this information is the ID, the name, the type and a justification text.

- **The right zone** shows two lists and a combo box.
  - *The combo box*. It shows all the database refinement frameworks to select the reference framework that will be the target of the equivalence map to create.
  - The upper list loads the elements, according to the filter selected, of the refinement framework chosen in the combo box that will be the target of the equivalence map to create.
  - The lower list displays the full content (not filtered) of the source refinement framework that will be postConditions in case of reusing. The postConditions are mandatory extra activities, not included in the standard, that must be performed in case of reusing the target element from one assurance project based in the target refinement framework in another assurance project based in the source refinement framework using the Cross-Domain functionality that will be explained in the section 5.5 Cross-Domain reuse. If the user double-clicks any element of this list, the source refinement framework could be modified to create new element to be used as postconditions (the save button must be pressed to save the changes).
For making an equivalence map, this window saves automatically the mapping information, follow the next steps:

1. Select a mapping model and a map group (or create it if needed).
2. Select the target reference framework.
3. Select the filter map element.
4. Select the element from the source reference framework.
5. Select or create the equivalence map and introduce the mapping information (ID, name, type and justification).
6. Check or uncheck the element from the target reference framework.
7. Create the postconditions if needed and check or uncheck the postconditions elements.
4.4 Creating Applicability Tables

To create Applicability Tables (the naming is to cover various standards, for IEC 61508 derived standards would be Recommendation Tables) such as this one:

Table 1 — System design analysis

<table>
<thead>
<tr>
<th>Methods</th>
<th>ASIL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A</td>
</tr>
<tr>
<td>Deductive analysis(^a)</td>
<td>0</td>
</tr>
<tr>
<td>Inductive analysis(^b)</td>
<td>++</td>
</tr>
</tbody>
</table>

\(^a\) Deductive analysis methods include FTA, reliability block diagrams, Ishikawa diagram.
\(^b\) Inductive analysis methods include FMEA, STA, Markov modeling.

For ISO26262 and IEC 61508 standards, you must select the desired “Requirement” from the Standard, go to Tab called RequirementApplicability (to go to a Requirement Form, you must first select the Activity which contains the Requirement, and double click on the Requirement properties), and you will see:
Then you can add rows by defining the target Method (technique), and select the Criticality Level and the Recommendation level.

NOTE: You must first create Criticality Levels (SIL) and Recommendation Levels (+++, +, o, or others as required) by clicking on the Diagram (in some blank space) where you will see Properties of the RefFramework model element (the Standard).

In the case of the DO-178C standard, to create Applicability Table such as the next figure:

![Applicability Table DO-178C](image)

**Figure 36 - Applicability Table DO-178C**

It needs to make two steps:
1. First, it needs to define the applicability table for the activities: select the desired “Activity” from the Standard, go to Tab called ActivityApplicability.

Then you can add rows by defining the target Requirement, and select the Criticality Level and the Recommendation level.

![Figure 37 - Activity Applicability Table DO-178C](image)

1. And finally, it needs to define the applicability table for the requirements: select the desired “Requirement” from the Standard, go to Tab called RequirementApplicability.

Then you can add rows by defining the target Artefact, and select the Criticality Level and the Recommendation level.

![Figure 38 - Requirement Applicability Table DO-178C](image)

In summary:
### Table A.1 Software Planning Process

<table>
<thead>
<tr>
<th>Objective</th>
<th>Ref</th>
<th>Applicability by Software Level</th>
<th>Output</th>
<th>Control Category by Software Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Ref</td>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td>The activities of the software life cycle processes are defined.</td>
<td>4.1.8</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>The software life cycle(s), including the inter-relationships between the processes, their sequencing, feedback mechanisms, and transition criteria, is defined.</td>
<td>4.1.9</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>Software life cycle environment is selected and defined.</td>
<td>4.1.6</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>Additional considerations are addressed.</td>
<td>4.1.6</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>Software development standards are defined.</td>
<td>4.1.8</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>Software plans comply with this document.</td>
<td>4.1.8</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>Development and revision of software plans are coordinated.</td>
<td>4.1.8</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
</tbody>
</table>

**Figure 39 - Summary Applicability Table DO-178C**

- **Activity Applicability Table**
- **Requirement Applicability Table**
- **Target: Requirement**
- **Target: Artefact**
5 Assurance Project Management

An Assurance Project has three main elements:

1. **Baseline Configuration**: a Baseline Configuration has a set of Baseline Models. Each baseline model results from importing (copying) a Reference Framework model and adding information about its Selection in the current project (it answers to the question: does a given Reference Framework model element apply to the current Assurance Project?). A Baseline model represents what is planned to do or to comply with, in a specific assurance project.

2. **Permissions Configuration**: This has not been implemented yet. It will support profile creation to enable restricted access to OPENCOSS functionality and data.

3. **Assurance Assets Package**: This is a pointer to project-specific Artefacts models, and Argumentation models, and Process models. These three models represent what has been done in a specific assurance project. The mapping of these three models with Baseline Models is modelled using the concept of Compliance Map.

One Assurance Project can have multiple Baseline Configurations, Permissions Configurations and Assurance Assets Package, but only one is active at once.

The next figure illustrates the elements of an Assurance Project:

![Figure 40 - Assurance Project structure](image)
5.1 Create Assurance Project and Baseline

To create a new assurance project go to the menu File → New → Project or use the button 📚 in the top button bar and select New Assurance Project inside the Opencoss category.

![New Project dialog]

*Figure 41 - New Assurance Project wizard*

The first page is to enter the name of Assurance project.
The next page of the wizard will show in the left the list of reference framework model in the whole repository. Select the desired refframework and in the right list will appear its contents in form of checkable tree for the generation of the baseline. Select the nodes of the tree that will be applied to the project are creating, give a name to the baseline and click the Finish button to generate all the project information (This process could take several seconds). Only is possible to Finish the process if you give a name and select almost one concept.
Now in the Repository Explorer the new project will be displayed.
The project is composed by 4 folders:

- Argumentation for storing the argumentation models with and argumentation model (with diagram) generated automatically based on the baseline’s selected entities. (For details, see section 6.2.2 Creating a Diagram at the project creation time)
- Assurance Project folder that has the project information in .assuranceproject model, the baseline information in .baseline model(with diagram) and the .mapping model to store the compliance mapping information.
- Evidence for saving the evidence models
- Process for the processes execution.

To edit the Assurance Project information double clicks over the model and its editor will appear. By default the assurance project has related all the models generated automatically, the baseline and mapping models in the active BaselineConfig and the argumentation model in the active AssetsPackage. If the user generates new models related to the assurance project, for example evidence model, he must select the right folder (EVIDENCE following with the example) of the assurance project as destination for the new model and update manually the assurance project model to reference the new models inside the project (AssuranceAssets following with the example).
5.2 Create or update Project Baseline

To create a new assurance project baseline or update and exiting one select menu File → New → Other or use the arrow in the right of the button in the top button bar and select Other.

Choose the wizard Creates or Updates Baseline behind the OpenCoss category.
The first page of the wizard requests the selection of the assurance project model to update.
The following steps are exactly the same than for the generation of a new assurance project. Select the desired reference framework model to be used as source for the generation of the baseline in the left list, then in the right list will appear its contents in form of checkable tree for the generation of the baseline. Select the nodes of the tree that will be applied to this baseline and give a name to the baseline taking into account that if the given name is the same as previous existing baseline, the contents of the previous one will be replaced with the information selected and the same will occur with the argumentation model.

Finally click the Finish button to generate the new baseline and argumentation models that will be added to the assurance project model and stored in the appropriate Assurance Project Folders.
5.3  Edit Project Baseline

To edit the baseline information double click over the .baseline model and its editor View will appear. The not selected elements will be displayed in the upper tree with a different icon.
The baseline model can be edited also be means of a graphical editor, to use it double click in the .baseline_diagram model. The way of using this editor is exactly equal than the refframework's editor explained in the chapter 4.2

5.4 Edit Compliance Maps

It is possible to create Compliance Maps in two ways:

- One way using the editor,
- Another way using a tailored functionality for it.

5.4.1 Compliance Map using the editor

To create Compliance Maps using the editor, we must load four CDO resources: the artefact model (.evidence), process model (.process), argumentation model (.arg) and the mapping model (.mapping).

It is important remind that these models have to be part of the active BaselineConfig and AssetsPackage of the Assurance Project, in other words,

- The artefact model, process model and argumentation model have to be part of the active AssetsPackage of the project:
The map group of the mapping model has to be part of the active BaselineConfig of the project.
So, press the editing window and select “Load Resource” in the context menu.

Figure 53 - Baseline Config active

Figure 54 - Load Resource
Other way is select the entry of the menu Baseline Editor -> Load Resource

![Figure 55 - Load Resource II](image)

Then select the resource model browsing the repository using the “Browse Repository” button.

![Figure 56 - Load Resource Evidence, Process or Argumentation model](image)
It is possible to create compliance maps for activities, artefacts, requirements, roles and techniques. To do so, first select the object in the tree and after click on the tab “Compliance Map” and press the button “Add”.

Finally, enter the information requested:
5.4.2 Compliance Map using a tailored functionality.

To create Compliance Maps using the tailored functionality, first of all, it is necessary to press the button “Mapping Set” on the properties form of the baseline using the tree view editor (not available using the diagram editor). This window automatically saves the mappings when checking or unchecking elements of the target baseline tree.
The Compliance Map form is organized in three zones:

- The **left zone** shows the actual baseline, and it loads the type of elements for which we want to make the compliance maps. For default, activities.

- The **middle zone** allows to make different filters like:
  - *Filter Mapping Model* lists all the mapping models stored in the database, and it will be necessary to select one of them and one group model. It’s also possible to create a new...
map group pressing the button “New group”. This map group has to be part of the active Baseline Config of the project.

- **Filter Map Element.** It’s possible to create compliance maps for activities, artefacts, requirements, roles and techniques, and the allowed maps are:
  - BaseArtefact -> Artefact
  - BaseRequirement -> Artefact, Claim or Activity
  - BaseActivity -> Activity
  - BaseRole -> Participant
  - BaseTechnique -> Technique

When the filter changes, also it changes the information showed by the reference framework. For example:

- If the filter “Artefact” is selected:

![Figure 62 - Compliance Map, select map element](image)

Remember that these models have to be part of the active Assets Package of the project.

- **Filter Compliance Map.** This filter allows making different compliance maps for the same element.

- The **right zone** shows the list of models, depend on the map filter, stored in our database. We should select one of them. This selected model will be the target of the compliance map to create.

For making a compliance map, follow the next steps:

1. Select a mapping model and a map group.
2. Select the target reference framework.
3. Select the filter map element.
4. Select the element from the source reference framework.
5. Select to create the compliance map.
6. And for last, check or uncheck the element from the target model.

### 5.5 Cross-Domain reuse

The cross-domain window objective is reusing the evidences from one source assurance project of one domain in a target assurance project of other domain. It’s mandatory that the target assurance project is based in a refframework with equivalence maps with the refframework in which is based the source assurance project and, logically, the source project must have and evidence model.
To access this functionality open the target assurance project model and press the button “Cross Domain” on the properties form of the Assurance Project element of the model.

![Figure 63 - Cross Domain button](image)

If the target project doesn’t have an evidence model a confirmation message will be displayed asking the user confirmation to create it.

If the user is agree, a new evidence model will be generated automatically based in the contents of the target assurance project baseline (for each BaseArtefact in the target baseline, one Artefact Definition with one Artefact will be included in the target evidence model), the evidence model will be related with the target assurance project and also the compliance maps between the baseline and the evidences are automatically created. After this the data generation process, the cross domain window will be opened.

If the user refuses, the data is not created and the cross domain window is not opened because is mandatory to have and evidence model as destination model of the reuse.

![Figure 64 - Create a new evidence model message](image)
If the target model already has an evidence model, the user is asked if he wants to use it as destination model of source Artefacts to reuse. If the user answers “Yes” the existing evidence model will be used as the target model of the reuse and the cross domain window will be opened. If the user says “No” the previously explained message will appear (see Figure above).

The Cross Domain window is organized in three zones:

- The **left zone** shows information about the target project. In the top part the URL of the target assurance project, above a tree with the target baseline contents, above the compliance map information of the target baseline element selected and the contents of the target evidence model in other tree.

- The **middle zone** displays equivalence map information. It includes controls to select the equivalence mapping model and the equivalence map group, display the equivalence map details of the target baseline element selected and its postconditions in a list (to see the ID, Name and description one postcondition must be selected).

- The **right zone** presents information about the source project. In the top part the URL of the source assurance project, above a tree with the source baseline contents, above the compliance map information of the source baseline element selected and the contents of the source evidence model.

The user can obtain detailed information of any element displayed in the trees in a popup window by double-clicking over it. In the case of the target evidence model, the model can be edited directly in the popup window and the changes will be saved after clicking the “Save target evidences” button.

Also is possible to create new Compliance Maps between the target baseline and the target evidence model using the “New CM” button connected with the tailored Compliance map window explained in Section 5.4.2
The user has to choose the source project of the reuse using the “Search button” and the source baseline and evidence model tree will be loaded. After this, has to select the equivalence model and the equivalence group. The next step is to select the target base element that will receive the evidences to be reused and its compliance and equivalence map information will be loaded, highlighting in green its target elements in the trees. Finally the user has to select the target Artefact and press the “Reuse” button to start the copy of the checked source Artefact/s to the target selected Artefact (only one can be selected).

As example the screenshot above shows that the selected BaseArtefact “ArtfA” of the target project “Project2” has a Compliance Map with the target Artefact “Artefact version 1” of the evidence model of the target project. Also has an equivalence map, inside the map group “MG-Artefacts”, with the source baseline BaseArtefact “Artf1” of the source assurance project “Project1” and as postcondition the BaseActivity “Extra Activity caused by reusing” (the equivalence maps are created at refframework level.
and are copied to the baselines during the assurance project generation process). The source BaseArtefact “Artf1” has a Compliance Map with the target Artefact “Artefact 1” of the evidence model of the source project. Therefore, the “Artefact 1” is a good candidate to be reused, according to the existing equivalence and compliance mapping information, and appears checked and highlighted in green.

This window checks the integrity of the data before start the copy process, for example:

![Cross domain information messages about integrity](image)

*Figure 68 - Cross domain information messages about integrity*

If the user wants to copy artefacts without equivalence between them, a confirmation message is showed.

![Reuse not equivalence artefacts confirmation message](image)

*Figure 69 - Reuse not equivalence artefacts confirmation message*

Finally, if all is correct another confirmation message with the resume of the data that will be copy is displayed.

![Cross domain final confirmation message](image)

*Figure 70 - Cross domain final confirmation message*

If the user continues the copy process will begin. The source repository configuration information inside the Artefact Model Object, the Resource objects of the checked source Artefacts and the repository files related to these resources will be copied to the target evidence model. Additionally, the postconditions will be selected in the target baseline model.
5.6 Cross-Project reuse

The cross-project window objective is reusing models from one source assurance project to a target assurance project, and also the diagrams will be copied to the target project if exists.

This window allows reusing only the selected source evidence models associated to the active Assets Package, because evidences are not related to any other model of the project, or all the baselines associated to the active Baseline Config and all the evidence, argumentation and process models of the active Assets Package. In this second option all the models will be cloned to assure the integrity of the data, for example, a baseline could be related with argumentations, evidences and/or processes and in this way we are sure all the related information is copied avoiding inconsistencies.

To access this functionality open the target assurance project model and press the button “Cross Project” on the properties form of the Assurance Project element of the model.

![Cross Project button](image)

The user has to select the source project and after its models will be displayed in the window. As said before, only the models related to the active Baseline Config and active Asset Package of the source assurance project will appear.
Figure 72 - Cross project: Source project selection

By default the “copy all models” option is selected but the user can uncheck the “copy baseline models” control to indicate that only wants to copy evidences. In this case, the user can select the desired models to copy from the evidence model list. To go back to the previous option only is necessary to check again the “copy baseline models” option.
Figure 73 – Cross project: Copy all models

Figure 74 – Cross project: Copy only evidences
To begin the copy process the user must click over the Reuse button. In case of copying all the models, the information message in the screenshot below will be show to clarify to the user that the active Assets Package and Baseline Configuration of the target process will be changed. In case of reusing evidences, this message won’t appear.

![Changes to target Assurance Project](image)

*Figure 75 - Cross Project information message*

When the copy process ends a message window will be shown.

In the next screenshot we can see, boxed in red, the new models copied and the new information added to the target assurance project model.

![Cross project reuse result](image)

*Figure 76 - Cross project reuse result*
6 Safety Argumentation Management

6.1 Preferences

Set some configuration Parameters in Window ➔ Preferences ➔ Opencoss ➔ Argumentation. In the section, you can define parameters required by the Argumentation diagram editor. The parameters which can be defined are below:

- Modules directory preference. This folder contains all argumentation modules stored from previous argumentation.
- Patterns directory preference. This folder contains all argumentation patterns templates.
- Agreements directory preference. This folder contains all agreements.

These folders are Eclipse Projects that should be created previously.

![Figure 77 - Argumentation Preferences](image)

6.2 Creating and Saving a Diagram

6.2.1 Creating a New Diagram

**In File Format**

To create a new file-based argumentation diagram, follow the procedure below and generate a new diagram in the project folder.

Select “New” ➔ “Other...” from the File menu of eclipse ➔ Select “Arg Diagram to File” in the Opencoss category and click the “Next” button. ➔ Select the folder to store the new diagram ➔ Enter the name of the diagram to be created and click the “Finish” button.
Figure 78 - File-based Argumentation Diagram wizard I
In Database Format

To create a new database-based argumentation diagram, follow the procedure below and generate a new diagram in the project folder.

1. Select “New” → “Other...” from the File menu of eclipse.
2. Select “Arg Diagram to Repository” in the Opencoss category and click.
3. Select the folder to store the new diagram.
4. Enter the name of the diagram to be created and click the “Finish” button.

---

**Figure 79 - File-based Argumentation Diagram wizard II**

**Figure 80 - Database-based Argumentation Diagram wizard I**
6.2.2 Creating a Diagram at the project creation time

When we create a new assurance project (See section 5.1 Create an assurance project and a Baseline) a new argumentation is created. This argumentation is created applying a transformation to the baseline. The rules that apply to the transformation are the following:

- Every reference activity is transform into a claim. The id, name and description of the reference activity become also the id, name and description of the claim.
- Every reference requirement is transform into a claim. The id, name and description of the reference requirement become also the id, name and description of the claim.
- Every reference artefact is transform into an information element with the property type marked as solution. The id, name and description of the reference artefact become also the id, name and description of the information element.
- When a reference activity has sub activities, then an asserted inference relationship is created, the source is the claim transformed from the top activity and the target the claim transformed from the sub activity
- When a reference activity has reference requirements, then an asserted inference relationship is created, the source is the claim transformed from the reference activity and the target the claim transformed from the requirement

6.2.3 Opening a Diagram

In File Format

Double click on the project folder on the Package Explorer tab in order to expand the folder. The stored diagrams will be shown. Double click on Argumentation Diagram information file (.arg_diagram) to open a diagram in the editing window. The diagram can then be edited.

In Database Format

Double click on the project folder on the Repository Explorer tab in order to expand the folder. The stored diagrams will be shown. Double click on Argumentation Diagram information file (.arg_diagram) to open a diagram in the editing window. The diagram can then be edited.
6.2.4 Saving a Diagram

To save a diagram, select one of the following items in the “File” menu.

1. “Save” item
   This utility is only available for file-based Argumentation Diagram. The contents of the selected editing window will be saved in the model information file and the diagram information file.

2. “Save As...” item
   This utility is only available for file-based Argumentation Diagram. The contents of the selected editing window will be saved in the model information file and the diagram information file with a different name.

3. “Save All” item
   The contents of all editing windows will be saved in the corresponding model and the diagram information files/database.

6.3 Editing Functions

6.3.1 Editing a Diagram

Nodes and relationships (or links) selected from Palette can be added to the canvas. Just select the node from the Palette, go to the editing window and select the place and size of the element.
The palette is structured into three different sections. Section “Argumentation core” includes the main nodes for argumentation. These nodes implement the GSN graphical notation. The “Argumentation relationships” includes all the different links between the different nodes. “Argumentation modular extensions” includes those nodes specific for the modular argumentation.

<table>
<thead>
<tr>
<th>Graphical notation</th>
<th>GSN concept</th>
<th>Argumentation editor</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="" alt="Goal" /></td>
<td>Goal</td>
<td>Claim</td>
</tr>
<tr>
<td><img src="" alt="Context" /></td>
<td>Context</td>
<td>InformationElementCitation Property type=&quot;context&quot;</td>
</tr>
<tr>
<td><img src="" alt="Strategy" /></td>
<td>Strategy</td>
<td>ArgumentReasoning</td>
</tr>
<tr>
<td><img src="" alt="Solution" /></td>
<td>Solution</td>
<td>InformationElementCitation Property type=&quot;solution&quot;</td>
</tr>
<tr>
<td><img src="" alt="SolvedBy" /></td>
<td>SolvedBy</td>
<td>AssertedInference, but only if the target is not a solution</td>
</tr>
<tr>
<td><img src="" alt="SolvedBy" /></td>
<td>SolvedBy</td>
<td>AssertedEvidence, but only if the target is a solution</td>
</tr>
<tr>
<td><img src="" alt="InTheContextOf" /></td>
<td>InTheContextOf</td>
<td>Asserted</td>
</tr>
<tr>
<td><img src="" alt="Underdeveloped" /></td>
<td>Underdeveloped</td>
<td>Property toBeSupported=&quot;true&quot;</td>
</tr>
<tr>
<td><img src="" alt="To be instantiated" /></td>
<td>To be instantiated</td>
<td>Property toBeInstantiated=&quot;true&quot;</td>
</tr>
<tr>
<td>ArgumentElementCitation</td>
<td>type=&quot;claim&quot;</td>
<td></td>
</tr>
<tr>
<td>--------------------------</td>
<td>--------------</td>
<td></td>
</tr>
<tr>
<td>ArgumentElementCitation</td>
<td>type=&quot;context&quot;</td>
<td></td>
</tr>
<tr>
<td>ArgumentElementCitation</td>
<td>type=&quot;solution&quot;</td>
<td></td>
</tr>
<tr>
<td>Argumentation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Agreement</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Claim</td>
<td>assumed=true</td>
<td></td>
</tr>
<tr>
<td>InformationElementCitation</td>
<td>type=&quot;justification&quot;</td>
<td></td>
</tr>
<tr>
<td>AssertedCounterEvidence</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AssertedChallenge</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Public Goal</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Optionality</td>
<td></td>
<td></td>
</tr>
<tr>
<td>multiplicity=optional</td>
<td></td>
<td></td>
</tr>
<tr>
<td>multiplicity=multi</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Choice</td>
<td>Choice</td>
<td></td>
</tr>
</tbody>
</table>

**Table 2 - Argumentation graphical notation**

In case of the links, just select the link from the palette, then on the editing window click on the source of the link and then release the click on the target of the link.
Popup menu opens when the mouse cursor is placed in the graphic-object editing area and is kept still for a moment. A node can be created by selecting the corresponding icon in the menu.

On the other hand, Properties View manages the properties of the current element under edition.

**Figure 85 - Claim properties**

**Claims**
These elements have the graphical notation as Goals on GSN. The identifier property indicates uniquely the claim on the argumentation. Do not use the same identification on different elements on the same argumentation.

Assumed property references the assumption concept and has the same graphical notation assumptions in GSN. It indicates an assumption in relation of a claim.

To be supported property indicates that the claim will be further developed on a future.

**Argument Reasoning**
This element has the same graphical notation as strategies on GSN. The identifier property indicates uniquely the element on the argumentation. Do not use the same identification on different elements on the same argumentation.

To be supported property indicated that the argument reasoning will be further developed on a future.

**Information Element Citation**
This element could have different graphical notations depending to the concept which is referencing. The property “Type” could have the values

- Justification: it references the justification concept and has the same graphical notation as justifications in GSN. It justifies the validity of a claim.
- Context: it references the context concept and has the same graphical notation as contexts in GSN. It indicates the context of a claim.
- Solution: it references the evidence concept and has the same graphical notation as solutions in GSN. It supports the validity of a claim.

The identifier property indicates uniquely the element on the argumentation. Do not use the same identification on different elements on the same argumentation.
Information elements references to a specific artefact. The url property indicates the location of the artefact associated. See section “Connecting a Diagram to Artefacts” for further details for information on these artefacts.

**Argumentation**

This element has the same graphical notation as modules on GSN. The identifier property indicates uniquely the element on the argumentation. Do not use the same identification on different elements on the same argumentation.

The Location attribute indicates where is stored the argumentation diagram file with the content of the module. By default is should be stored on the places indicated on the preferences.

### 6.3.1.1 Copying and Pasting an Element

Elements in a diagram can be copied and pasted. To copy an element, select the element and click “Copy” in the “Edit” menu. The copied element can be pasted by clicking “Paste” in the “Edit” menu without selecting any element.

### 6.3.1.2 Deleting a Node or a Link

Please, do not press the Del key. The element will not be completely deleted.

To delete a node or a link, click to select the node/link and perform either of the following steps.

1. Press the “BS” key.
2. Right click the item and select “Delete from Model” from the context menu.

PS: An argument can be deleted. But child elements are also deleted and the argument cannot be edited ever.

### 6.3.2 Create multi-diagrams from an Argumentation model

The tool allows managing different views of a model through a set of diagrams. Once a model is available, a new diagram view can be created and special edition functionalities are available as follows:

1. Thanks to the Outline view, it is possible to drag and drop concepts from the model to the diagram.
2. Once a concept has been selected, it can be hidden through the “Delete from diagram” option available in the contextual menu. This option does not delete the concept from the model.
3. Once a concept has been selected, it can be deleted through the “Delete from model” option available in the contextual menu. This option delete the concept from the model permanently. If this deleted concept is visible in another diagram files, this concepts will be shown with a cross icon in the upper right corner to show that it does not exist anymore.

**Create multi-diagrams in File Format**

Once a model is available, a new diagram view can be created by using the “Initialize arg_diagram diagram file” option in the contextual menu.
This option launches a wizard that at the first step requires the folder and the name of the new diagram file. Next, the root element of the model (Case type) must be selected as the root element of the new diagram. After that, the diagram is ready for edition.

Figure 86 - Initialize a diagram file

Figure 87 - Selection of the Case root element
Create multi-diagrams in Database Format

Once a model is available, a new diagram view can be created following the procedure below.

Select “New” → “Other...” from the File menu of eclipse

Select “Arg Diagram to Repository” in the Opencoss category and click the “Next” button.

Select the folder to store the new diagram. Then type the name of the diagram to be created and click the “Next” button.

Type the name of the model created previously and click the “Finish” button.

---

**Figure 88 - Database-based Argumentation Diagram wizard I**

**Figure 89 - Database-based Argumentation Diagram wizard II**
After that, the diagram is ready for edition.

### 6.3.3 Editing a Diagram Using a Pattern or a Module

In order to display the Template View, follow the next steps: Go to window ➔ Show View ➔ Other
On the Menu expand the Opencoss category and select the Templates view.
Double click on a folder will expand the folder. Double click on an Argumentation Diagram file (.arg_diagram) will open the diagram in the editing window.
Modules and Patterns are stored on places determined by the preferences (see previous section). The Patterns and Modules should be stored as ready to be reused. From the Template view, the drag and drop operation is activated in order to reuse the pattern or module on the actual argumentation.

6.3.3.1 Creating a New Pattern Diagram
To create a new Argumentation Pattern diagram, follow the procedure of “Creating a New Diagram”. The only difference with other argumentation diagrams is that Patterns need to be stored on the places designed by the preferences. By default preferences point to a project called “Patterns” on the workspace.

6.3.3.2 Editing a Pattern Diagram
Proceed as explained in the “Editing a Diagram” section. Remember only Pattern Diagrams support structural and entity abstraction. Structural abstractions like multiplicity or optionality are available through “Multiextension” and “Cardinality” properties of relationships. In addition, structural options are addressed by a “Choice” relationship. Entity abstraction can be accomplished by “To Be Instantiated” and “To Be Developed” properties. While editing a claim one of the properties than can be change is “To be instantiated”, when its value it “true” then the node will change its graphics.

---

1 At creation time the “Choice” link must be placed inside its source node.
The “AssertedInference” has a property called “multiextension” with three different values:

- Normal: behaves as a regular connection. It has the “supported by” graphical notation
- Optional: indicates that this connection is optional or alternative connections between the nodes
- Multi: indicates the generalised n-ary relationships between the nodes. When this option is selected, the attribute “cardinality” should also be modified indicating the “n” value

The “choice” relationship is specific for patterns edition. For editing, select the Choice at the Palette and then place it inside its associated top Claim. The graphic for the choice relationship will appear affixed to it. Then add as many asserted inferences as the number of possible choices.
6.3.3.3 Adding Elements from Patterns to a Diagram (instantiating a Pattern)

An Argument Pattern can be instantiated (thus all its content copied) into the diagram under edition. To proceed, drag and drop a Pattern Diagram file into the diagram under edition. You need to open the argumentation diagram uncompleted that needs to include the pattern. Go to the templates view, select the pattern you are interested in, once selected you can drag and drop it into the editing part of the diagram. Once you drop you will see the new elements that have been copied into your argumentation diagram.

Once the drop is done, the “Arrange Selection” feature can be used to move to all the nodes and links. This feature can be found on the top menus as a button.

6.3.3.4 Creating a New Module Diagram

To create a new argumentation Pattern diagram, follow the procedure of “Creating a New Diagram” but generating the new diagram into the Modules directories. The only difference with other argumentation diagrams is that Modules need to be stored on the places designed by the preferences. By default preferences point to a project called “Modules” on the workspace.

6.3.3.5 Editing a Module Diagram

Proceed as explained in the “Editing a Diagram” section. Remember Modules Diagrams (“Argumentation”) allows representing interrelated modules of arguments. An “Argument Element Citation” repeats an element presented in another argumentation module which is used to support the argument in the local module. “Public” property indicates that an element is visible to other modules where it can be referenced. While an “Agreement” element represents the agreed relationship between modules. To indicate that a claim is Public, just indicate on the Claim properties view, the attribute “Public” be “true”. The Public activation is also noticed on the graphic notation.
Argument Element Citations can represent different concepts; they can reference a claim, a context or a solution. On the properties view, an attribute called “cited type” should be informed and consequently the graphic notation might change.

The “Argumentation Reference” property indicates the reference to the module in which this citation element is described.

6.3.3.6 Adding Elements from Modules to a Diagram (instantiating a Module)
An Argumentation Module can be instantiated (thus all its content copied) into the diagram under edition. To proceed, drag and drop a Module Diagram file into the diagram under edition. Once is its instantiated an “argumentation” graphic notation will appear on your diagram.
By double clicking on this Argumentation another diagram will appear with the argumentation context of this module. The url property indicates the location of this diagram.

6.3.4 Connecting a Diagram to Artefacts

Firstly, proceed to load the evidences model (.evidence) from the repository. So, press the Outline and select “Load Resource” in the context menu.

![Load Resource to Argumentation Diagram](image)

*Figure 98 - Load Resource to Argumentation Diagram*

Then browse into the workspace to select the evidences model resource.

![Select Evidence model as resource](image)

*Figure 99 - Select Evidence model as resource*

Now, information element citations can be related to its artefacts. To carry out, select an “Information Element Citation” and press “+” operator in Artefact section.
Figure 100 - Artefact selection as Information Element Citation

A pop-up window will show the Artefacts Model at “All Resources” tab. Finally, choose the required artefact.

Figure 101 - Artefact selection from resources

In addition, it is possible to launch the artefact editor by double click on one artefact instance as shown below.
6.4 Argumentation Contracts

A contract is essentially a recording of an agreement between two or more entities. In the case of argumentation modules in OPENCOSS, a contract records how an inter-module dependency (a public claim to be supported in a module) is resolved by the claims and evidence provided by other argumentation modules.

6.4.1 Preparation for Contracts

Create two or more argumentation diagrams (see Section 6.2.1 Database Format). In each of these diagrams (or their accompanying models), an "Argumentation" (GSN module symbol in the palette) should be created (Figure 103). Argument elements such as Claims, Information Element Citations, and Argument Reasoning should then be added to the Argumentation (module) element. This is achieved by right clicking on the Argumentation element and selecting "Show Properties View", clicking on the "+" button on the "Consist Of" property, and selecting the model elements that should be added to the argumentation element. The "+" button for the "Consist Of" property in the properties view for an Argumentation (module) is shown in Figure 104.
**Figure 103** Creating an Argumentation (Module)

**Figure 104** "Consist Of" Property for an Argumentation (Module)
To declare that an inter-module dependency exists, a Claim element in a module should have the properties "public" and "to be supported" set to true (Figure 105). There can be multiple dependencies.

![Figure 105 "To Be Supported" and "Public" Property Values for Declaring a Claim Requiring Support from Another Module](image)

To declare that a Claim can provide support to resolving inter-module dependencies, the "Public" property should be set to "true" and the "To Be Supported" property should be set to "false" (Figure 106).

![Figure 106 "To Be Supported" and "Public" Property Values for Declaring a Claim Providing Support to Other Modules](image)
Note, Information Element Citations, when used to model "Solutions", can also be used to provide support to inter-module dependencies. The "To Be Supported" property should be set to "False" in this case.

### 6.4.2 Creating a Contract

Having created argumentation models which consist of modules of model elements that require and/or provide support to other modules, contracts can be created to link such elements (thus resolving the inter-module dependencies). To create a contract, go to File → New → Other → Opencoss → Argumentation Contract to start the Contract Wizard (Figure 107).

![Figure 107 New Contract Wizard](image)

In the first page of the wizard, selecting an argumentation model from the dropdown box will display a list of all inter-module dependencies (Claims that are public and to be supported) of all modules contained in the model. One of the Claims should be double clicked to select this as the inter-module dependency to be resolved by the contract being created (Figure 108). The second wizard page is similar to the first, except now the user should double click the claims and/or solutions that will resolve the dependency selected in the first wizard page (Figure 109). Note that one or more claims and/or solutions may be selected on this page. In the third page of the wizard, an explanation may be entered as to how the selected claims and/or solutions resolve the selected inter-module dependency (Figure 110). After completing the wizard, an argument diagram representing the contract is created and opened. This can be edited like any other argument diagram.
Figure 108 Selecting an Inter-Module Dependency

Figure 109 Selecting Elements to Resolve the Dependency
6.5 Argumentation Integration

After creating contracts to resolve all argumentation inter-module dependencies, a system integration argument should be created to pull together the strands of argument from all of the modules of software components that comprise the system. To create a system integration argument go to File —> New —> Other —> Opencoss —> Argumentation Integration (Figure 111).
The first wizard page, the user selects all of the argumentation modules that will comprise the system. The second page is where the user selects all of the contracts that resolve all inter-module dependencies of modules that will comprise the system.
6.6 Argumentation Change Management

After integrating argumentation (Section 6.5), it is possible to check whether argument modules which comprise the system have changed in ways which might compromise the integrity of the contracts that have been created. The user should click Argumentation → Argumentation Change Analysis.

The user should then select the system integration argumentation model that was generated from Section 6.5. The results of the analysis will then be displayed.
6.7 Vocabulary

The safety argumentation supports the usage of terms and term categories which have been defined in a vocabulary. The meaning of some terms and categories is specific to legislative regulations, standards and or the project they are used in. Vocabularies capture the meaning of such terms and categories by providing a definition.

6.7.1 Defining Vocabularies

The simplest way to define a vocabulary is to create one in the project where it going to be used. They can be stored either locally in a file or inside a remote repository.

The vocabulary can be visualized in a diagram to which shows how terms are related to each other. The diagram is a visualization of the vocabulary model but can also be used to edit the model.
Writing vocabularies is a time-consuming process. To save some or all of that work, vocabulary data can be imported from files in a custom XML format. In order to talk about elements of the CCL model in the argumentation, it is not necessary to duplicate the model elements as vocabulary terms. Instead, they can be imported into a vocabulary.

**Figure 117 – Example Vocabulary Diagram**

**Figure 118 – Vocabulary Import**
6.8 Using Vocabularies in the Argument Editor

Terms and term categories from the vocabulary can explicitly be used in the argumentation. In order to do so, some mark-up is required. The mark-up is visible while editing text, otherwise it will not be.

One of the following mark-up variants can be used:

- `voc:term`
- `voc:“term”` – Usually for terms with spaces or when the sentence ends after the term.
- `voc:term|terms` – Provide a natural language expression to be rendered instead of the term name, e.g. the plural form of the term. Add quotes if the term or the expression contains spaces.

Categories use the “var” prefix instead of “voc”.

Syntax highlighting and tooltips for vocabulary elements are available inside the argumentation editor.

While editing text, pressing Ctrl + Space will display a list of available vocabulary items. The list gets smaller when the user types the starting letters of the item he is looking for. Pressing enter inserts the item at the cursor.
6.9 Pattern Instantiation

Argumentation templates are likely to contain text placeholders. Typically, a template containing such a placeholder is inserted in the argumentation from the templates view using drag and drop. If the placeholder is correctly marked as a (general) category from the vocabulary, it can be quickly instantiated with one of the (specific) vocabulary terms from that category. The instantiation action is available in the context menu when right-clicking on the placeholder.

Figure 121 – Term Suggestions

Figure 122 – Pattern Instantiation
6.10 Structured Sentences

Oftentimes, statements inside of safety claims fall into different categories, e.g. risk claims or timing claims. For each of these claim types there is only a small number commonly used sentences if the sentences are appropriately parameterized using general categories from the vocabulary. An example of such a parameterized sentence structure might be:

The *(risk)* has been mitigated by *(risk mitigation technique)*.

Predefined sentence structures, grouped into a set of claim types such as the risk claim type, offer the user typical statements he likely wants to make in his safety argumentation. They are even more useful when defining argumentation templates because large parts of the template will use parameterized sentences. The Claim Types Editor allows the user to define claim types and structured sentences. Create a new list of claim types under File -> New -> Other.

![New Claim Types Editor](image)

*Figure 123 – New Claim Types Editor*

In the following example, the sentence structures have been parameterized with categories from the vocabulary like “risk” and “risk mitigation technique”.
The properties view of the argumentation editor has been extended to list the available claim types. Select any of the goals, navigate to the appropriate sentence structure and assign the sentence structure to the goal. The color associated with the parent claim type is used to colorize the goal as a visual reminder of the claim type, e.g. blue represents any claim about risk.
6.11 Printing

The following operations regarding printing are available under the “File” menu.
1. “Print Preview” item: Print preview
2. “Print...” item: Print
3. “Page Setup...” item: Print settings
7 Evidence Management

7.1 Define Artefact Repository Preferences

The first step before creating an Evidence Model is to indicate the SVN Repository configuration information to store the artefact files using the menu Windows → Preferences

![Preference menu](image)

Then select the Opencoss → Artefact Repository Preferences category and introduce the information required. If you want to use a local directory as Artefact Repository you have to check the “Use Local Repository” or uncheck it to use a SVN Server, the path of the local folder used as Local repository, the URL of the remote SVN server and the user and password of the SVN server.
The management of Evidences must be made through the creation of a new model of the type Evidence Model.

In order to generate a new Evidence Model, the following steps need to be done:

- First, select the entry of the menu File -> New -> Other:

![Figure 127 - Artefact Repository Preferences](image1)

![Figure 128 - New Evidence Model menu File -> New -> Other](image2)
• Inside the category of the wizard *Opencoss*, select the *Evidence Model* to the Repository and press the *Next* button:

![Image](image1.png)

*Figure 129 - New Evidence Model I*

• Enter o select the parent folder, the resource name and press the Next button:

![Image](image2.png)

*Figure 130 - New Evidence Model II*

• And finally, select the “Artefact Model” object to create.
Once the Evidence Model has been created, the first item is presented to the user.

The “Copy Preferences” button will copy the Artefact Repository Preferences data to this model and will be saved in the model to be used to store the Artefact files of this evidence model. If this information is empty, then the data specified in the Artefact Repository Preferences will be used to store the artifact files.
7.2  Artefact Definition

7.2.1 Add an artefact definition

It is possible to add artefacts definition to an artefact model in two ways:

- One way, select the artefact model and press the button

![Figure 133 - Add New Artefact Definition (I)](image1)

- Another way, click on the branch Artefact Model, press the right mouse button and select the contextual menu New Child → Artefact Definition.

![Figure 134 - Add New Artefact Definition (II)](image2)
In the properties zone, the framework presents several fields to describe the new Artefact Definition divided in tabs:

1. Artefact Definition (base)
   - **Id:** Artefact Definition identifier.
   - **Name:** Artefact Definition name.
   - **Description:** Artefact Definition description.

2. Artefact Definition Artefact
   - **Name:** Artefact name. This field is read-only.
   - **Version ID:** Identifier of the artefact version.
   - **Date:** Date of the artefact version.
   - **Last Version:** This field shows what artefact is the version in use.
   - **File ID:** Identifier of the file associated with the artefact.
   - **Name:** Name of the file associated with the artefact.
3. Artefact Definition Evaluation
   - References to the assurance asset evaluations that specify the outcome of evaluating the artefact.

4. Artefact Definition Events:
   - References to the assurance asset events of which the lifecycle of the artefact consists.

7.2.2 Delete Artefact Definition.

To delete an artefact definition:
- Select the artefact definition, press the right mouse button and select the contextual menu Delete:
Or, select the artefact model, select the artefact definition model to delete and press the button : 

7.3 Artefact

7.3.1 Add an artefact
It is possible to add artefacts to an artefact definition in two ways:
• Select the artefact definition, press the right button of the mouse and select the contextual menu \textit{New Child $\rightarrow$ Artefact}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig142.png}
\caption{Add New Artefact (I)}
\end{figure}

• Or, select the artefact definition, select the Artefact Definition Artefact tab Properties, and press the button \textit{Add}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig143.png}
\caption{Add New Artefact (II)}
\end{figure}

\textit{When user modifies one Artefact, the system automatically adds to it an AssuranceAssetEvent of type \textit{Modification}.}

In the properties zone, the framework presents several fields to describe the new Artefact divided in tabs:
1. Artefact Definition (base)
   - Id: Artefact Definition identifier.
   - Name: Artefact Definition name.
   - Description: Artefact Definition description.

2. Artefact Version
   - Version ID: Identifier of the artefact version.
   - Date: Date of the artefact version.
   - Changes: Changes made in the artefact version.
   - Last Version: This field shows what artefact is the version in use.
   - Is Template: Check if the artefact is a template.
   - Is Configurable: Check if the artefact is configurable.
   - Resource: List of resources associated to the artefact,
3. Artefact Property Value
   - Property: Property name.
   - Value: Property value.

4. Artefact Evaluation
   - Evaluation: References to the assurance asset evaluations that specify the outcome of evaluating the artefact. When a user introduces Evaluation information to an Artefact, and AssuranceAssetEvent of type Evaluation is added automatically to the Artefact.
5. Artefact Events
   - LifecycleEvent: References to the assurance asset events of which the lifecycle of the artefact consists.

![Artefact Events](image)

**Figure 148 - Description Artefact Events**

When adding child Artefact to other Artefact, it’s created automatically ArtefactRel information with modificationEffect=MODIFY and revocationEffect=MODIFY with source= parentArtefact and target=child Artefact.

### 7.3.2 Delete an artefact.

To delete an artefact:
- Selecting the artefact, press the right mouse button and select the contextual menu **Delete**.

![Delete Artefact](image)

**Figure 149 - Delete Artefact I.**
• Or select the branch Artefact Definition that contains the artefact to delete, select the ArtefactDefinitionArtefact tab, select the artefact and press the button Delete.

![Image of OpenCOSS Platform Tools User Manual page 116]

**Figure 150 - Delete Artefact II.**

### 7.4 Artefact Resource

#### 7.4.1 Add an artefact resource to an artefact

Once selected the artefact:

- Press the right mouse button and select the contextual menu New Child -> Resource to bring up the Artefact File properties.

![Image of Artefact Resource Addition]

**Figure 151 - Add Artefact Resource I**
• Or, select the Artefact Version tab and press the button.

Figure 153 - Add Artefact Resource II
In case of using a Local Repository to add the file, press the button Location or Assign, and select the file that will be added to the artefact resource from the local drive. The URL of the repository will be displayed in bold.

1. In case of using a Remote Repository to add the file, press the button Location to select the file that will be added to the artefact resource from the local drive and after press the “Commit” button to upload it to the SVN server. If the already file exists in the SVN Server use the “Assign”
button to select and assign it to the artifact version. Finally the SVN history of the file will be displayed in the table below.

![Select File from repository to assign](image1)

**Figure 156 - Select Artefact from the SVN Remote Repository**

<table>
<thead>
<tr>
<th>Location:</th>
<th>doc/Opencoss_SDVS.pdf</th>
</tr>
</thead>
<tbody>
<tr>
<td>Commit</td>
<td></td>
</tr>
<tr>
<td>Assign</td>
<td></td>
</tr>
<tr>
<td>Delete</td>
<td></td>
</tr>
<tr>
<td>Open</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Revision</th>
<th>Date</th>
<th>Author</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1408</td>
<td>10/21/2013</td>
<td>angel.lopez</td>
<td>New artefact Opencoss_SDVS</td>
</tr>
</tbody>
</table>

![SVN History table of a File](image2)

**Figure 157 - SVN History table of a File**

2. Clicking the “Open” button will launch the corresponding application to open the file. In case of remote repository, the file will be downloaded in a local temporarily file.
3. Clicking the “Delete” button will delete the file from the repository, local or remote.

### 7.4.2 Delete an artefact resource.

To delete an artefact resource:

1. After selecting the artefact resource, press the right mouse button and select the contextual menu **Delete**.

![Delete Artefact Resource](image3)

**Figure 158 - Delete Artefact Resource**
2. Or, select the artefact version tab, select the Resource to remove, and press the icon button.

![Figure 159 - Delete Artefact Resource II.](image)

### 7.5 Artefact Property Value

Firstly, it’s necessary to load the CDO resource property model (.property). So, press the editing window and select “Load Resource” in the context menu.
Then introduce the URI of the property model.

![Figure 160 - Load Resource Property model.]

**Figure 160 - Load Resource Property model.**

7.5.1 **Add an artefact property value to an artefact**

Once the artefact is selected, it is possible to add an artefact property in two ways:

1. One way, selecting the tab Artefact Property Value and pressing the button Add to bring up the Property Value dialog box:
2. Another way, pressing the right mouse button and selecting the contextual menu New Child -> Value to bring up the Artefact Property properties.
It is possible to delete an artefact property in two ways:

1. One way, select the artefact property in the tree, press the right mouse button and select the contextual menu **Delete**.
2. Another way, select the parent artefact of the artefact property to remove in the tree, select the Artefact Property value tab, select the artefact property and select the button Delete.

### 7.6 Artefact Assurance Asset Evaluation

#### 7.6.1 Add an artefact assurance asset evaluation to an artefact

Once the artefact is selected, it is possible to add an assurance asset evaluation in two ways:
1. One way, selecting the tab Artefact Evaluation and pressing the button Add to bring up the an Assurance Asset Evaluation dialog box:

![Image 1](image1.png)

**Figure 168 - Add Artefact Assurance Asset Evaluation I**

![Image 2](image2.png)

**Figure 169 - Artefact Assurance Asset Evaluation dialog box**

2. Another way, pressing the right mouse button and selecting the contextual menu New Child -> Assurance Asset Evaluation to bring up the Assurance Asset Evaluation properties.
7.6.2 Delete an artefact assurance asset evaluation

It is possible to delete an assurance assets evaluation in two ways:

1. One way, select the assurance assets evaluation in the tree, press the right mouse button and select the contextual menu *Delete*.
Another way, select the parent artefact of the assurance asset evaluation to remove in the tree, select the Artefact Evaluation tab, select assurance asset evaluation and select the button Delete.

![Figure 172 - Delete Artefact Assurance Asset Evaluation I](image1)

**Figure 172 - Delete Artefact Assurance Asset Evaluation I**

7. Another way, select the parent artefact of the assurance asset evaluation to remove in the tree, select the Artefact Evaluation tab, select assurance asset evaluation and select the button Delete.

![Figure 173 - Delete Artefact Assurance Asset Evaluation II](image2)

**Figure 173 - Delete Artefact Assurance Asset Evaluation II**

### 7.7 Artefact Assurance Asset Events.

#### 7.7.1 Add an artefact assurance asset event to an artefact

Once the artefact is selected, it is possible to add an assurance asset event in two ways:

1. One way, selecting the tab Artefact Events and pressing the button Add to bring up the an Assurance Asset Event dialog box:
2. Another way, pressing the right mouse button and selecting the contextual menu New Child -> Assurance Asset Evaluation to bring up the Assurance Asset Evaluation properties.
7.7.2 Delete an artefact assurance asset event

It is possible to delete an assurance assets event in two ways:

1. One way, select the assurance assets event in the tree, press the right mouse button and select the contextual menu **Delete**.
2. Another way, select the parent artefact of the assurance asset event to remove in the tree, select the Artefact Event tab, select assurance asset event and select the button Delete

![Figure 178 - Delete Artefact Assurance Asset Event I](image)

![Figure 179 - Delete Artefact Assurance Asset Event II](image)

### 7.8 Impact analysis

This functionality informs to the user about the impact of the changes in one Artefact that affect to others taking into account the relations between the modified and the impacted artefacts ones.

When the user clicks the save button to store the modifications made in one evidence model, for each Artefact modified with impact to another, the user will be asked in a confirmation dialog if he is agree or not with the showed impact information in form of tree or not.
If the user accepts the impact showed new assurance assets events will be generated to the modified and impacted artefacts.

**Figure 180 - Artefact modified with automatically generated events**

**Figure 181 - Artefact analyser confirmation windows**
Figure 182 - Artefact events created by Impact Analyser

See also Impact Analysis described in server section in Change Impact Analysis chapter.
8 Process Assurance Management

The management of Process Assurances must be made through the creation of a new model of the type Process Model.

In order to generate a new Process Model, the following steps need to be done:

- First, select the entry of the menu File -> New -> Other.
- Inside the category Opencoss, select the Process Model and press the Next button.

**Figure 183 - New Process Model I**

- Enter o select the parent folder, name the model and press the Next button.

**Figure 184 - New Process Model II**
• And finally, select the “Model” object to create

![Figure 185 - New Process Model III](image)

Once the Property Model has been created, the first item is presented to the user

![Figure 186 - Process Model](image)

### 8.1 Creating Process Assurance data

The Process Model allows defining activity, participant, person, tool, organization or technique objects.
To create these objects, in the Model zone, click on the branch **Model** and press the right mouse button and select the contextual menu New Child or use its properties view:

![Figure 187 - Create Process Model data using context menu](image1)

![Figure 188 - Create Process Model data using properties View](image2)
8.2 Deleting Process Assurance Objects

To delete a Process Assurance Object, select the object to remove, press the right mouse button and select the contextual menu Delete or select the information to delete using the properties view:

Figure 189 - Delete Process Model data using context menu
Figure 190 - Delete Process Model data using properties view
9 Properties Management.

The management of Properties must be made through the creation of a new model of the type Property Model.

In order to generate a new Property Model, the following steps need to be done:

- First, select the entry of the menu File -> New -> Other:

![Figure 191 - New Property Model menu File -> New -> Other](image)

- Inside the category of the wizard Opencoss, select the Property Model to the Repository and press the Next button:

![Figure 192 - New Property Model I](image)
• Enter o select the parent folder, the resource name and press the Next button:

![Figure 193 - New Property Model II](image)

• And finally, select the “Model” object to create.

![Figure 194 - New Property Model III](image)

Once the Property Model has been created, the first item is presented to the user.
9.1 Property

9.1.1 Add a property

It is possible to add properties to a property model in two ways:

- Select the model element, press the right button of the mouse and select the contextual menu "New Child -> Property"
• Or, select the model element, and press the icon button in the base tab

**Figure 197 - Add New Property (II)**

After these actions, in the properties zone, the framework presents several fields to describe the new property:

**Figure 198 - Property properties**

- Id: Property identifier.
- Name: Property name.
- Datatype: Property data type. Possible values: enumeration, string, integer and float.
- Enum values: values of an enumeration data type property. To add this values, press the button “Enum value”
9.1.2 Delete a property.

To delete a property:

- Select the property, press the right mouse button and select the contextual menu **Delete**.
- Or select the branch **Model that contains the property to delete, select the property and press the icon button**.
Figure 201 - Delete Property II.
10 Mapping Management

The management of Mapping must be made through the creation of a new model of the type Mapping Model.

In order to generate a new Mapping Model, the following steps need to be done:

- First, select the entry of the menu *File* -> *New* -> *Other*:

  ![Figure 202 - New Property Model menu File -> New -> Other](image)

- Inside the category of the wizard *Opencoss*, select the Mapping Model and press the Next button:

  ![Figure 203 - New Mapping Model I](image)

- Enter or select the parent folder, the resource name and press the Next button:
And finally, select the “Map Model” object to create.

Once the Mapping Model has been created, the first item is presented to the user.
10.1 Map Group.

10.1.1 Add a map group.

It is possible to add map groups to a mapping model in two ways:

- Select the model element, press the right button of the mouse and select the contextual menu *New Child → Map Group*

- Or, select the model element, and press the icon button ![icon](image) in the base tab
After these actions, in the properties zone, the framework presents several fields to describe the new map group:

- **Id**: Map group identifier.
- **Name**: Map group name.
- **Description**: Map group description

### 10.1.2 Delete a map group.

To delete a map group:
- Select the map group, press the right mouse button and select the contextual menu *Delete*. 
I. 

10.2 Map.

10.2.1 Add a map.

It is possible to add maps to a mapping model in two ways:

- Select the model element, press the right button of the mouse and select the contextual menu

  *New Child* $\rightarrow$ Map
Or, select the model element, and press the icon button in the base tab associated to the label Map Model:

Figure 213 - Add New Map (II)

After these actions, in the properties zone, the framework presents several fields to describe the new map:
10.2.2 Delete a map.

To delete a map:

- Select the map, press the right mouse button and select the contextual menu Delete:

![Figure 215 - Delete Map I.](image)

- Or select the branch Model that contains the map to delete, select the map and press the icon button associated to the label MapModel:
Figure 216 - Delete Map II.
11 OPENCOSS server

As described in “Client-Server architecture” chapter, OPENCOSS platform tools consist of:

- OPENCOSS server - installed in a central host machine.
- One or many OPENCOSS clients - each of which installed on specific user machines.

The role of the central OPENCOSS server is threefold:

- It hosts CDO server, which facilitates a **common storage** for OPENCOSS server applications and OPENCOSS clients.
- It provides web interface with **OPENCOSS reports** presenting common storage assurance data from different angles.
- It hosts **OPENCOSS API** services, e.g. evidence service or process service.

This chapter describes functionality provided by OPENCOSS server web reports - which facilitate a OPENCOSS server front-end for OPENCOSS platform users.

11.1 OPENCOSS Web interface layout

OPENCOSS server web pages are served at 8080 port by default.

In order to view the web pages, please run your web browser and go to the following location:

```
http://<OPENCOSS-SERVER-HOST-NAMES>:8080/
```

![Web interface layout](image)

*Figure 217 - Web interface layout*

A typical OPENCOSS server web page consists of the following panels:

- **Top panel**
  - It contains links to User Manual documents and links to server administration pages.
- **Project and Menu panel**
  - It contains:
    - A select box with assurance projects which have been created in OPENCOSS platform.
    - Main menu with links to OPENCOSS server reports.
    - There are several reports presenting analytical view from assurance data stored in OPENCOSS platform.
Each of the reports is described in the subsequent chapters.

- Main panel
  It presents the main content of the page - depending on the current report or page selected from the menu and the given assurance project.

## 11.2 Compliance report

### 11.2.1 Goal of the report

Compliance report provides extensive functionality which helps OPENCOSS platform users to assess the current compliance of their project to the selected safety standard (i.e., baseline).

The functionality is intended to be used by:

- Project **team members**, for example developers, when the project is in progress, in order to have up-to-date insights into which of the baseline framework items are already satisfied and to what extent.
- Project **safety manager** in order to monitor the project general compliance, observe the compliance details and add, assign, or un-assign specific evidence resources to/from the given requirement of the safety standard which is followed by the project.
- Independent **safety assessor**, when the project draws to an end, in order to browse the assigned safety evidence, evaluate it and independently assess the actual project compliance to the specific safety standard.

Two modes of the report can be distinguished:

- An **interactive mode**, where user can actively browse the report, select the specific baseline items, view their properties, their compliance mapping, and the associated evidence, and add or remove the evidence resources mapped to the specific baseline element.
- A **printer friendly report** - which is a textual output presenting all the information of the current compliance of the selected project.

The compliance report can be accessed via the following OPENCOSS web server menu item.

![Figure 218 - Menu item directing to “Compliance report”](image)

### 11.2.2 Viewing compliance data on the report

The compliance report allow users to see the overall compliance of the selected project to the specific safety standard.
When a specific OPENCOSS assurance project is selected in the top panel, its defined baselines are presented in the middle panel select box.

![Baseline Frameworks combo box for the specific project](image1)

**Figure 219 - Baseline Frameworks combo box for the specific project**

The report data section is divided into 4 panels.

![4 panels of "Compliance report"](image2)

**Figure 220 - 4 panels of "Compliance report"**

The “**Project Compliance**” table, which is placed in the left, presents base artefacts and base activities of the selected safety standard. The most important column is the “**Compliance Status**” one, which presents the overall compliance status of a project to the specific safety standard item. The column can be sorted by value, thus allowing user to assess the project compliance at one glance.

In case base activities or base artefacts are defined to have a parent-child hierarchy, this relation is presented accordingly in a tree structure of the table.

**Note:**

“**IA Status**” column presents the current status of specific baseline element from Impact Analysis point of view. This functionality has been described in a separate chapter: Impact Analysis result presentation on OPENCOSS server reports.

When a specific baseline element item (i.e. table row) is selected, its description and properties, as defined in OPENCOSS storage, are presented in the bottom-left panel.
Upon the selection of the specific safety standard item in the “Project Compliance” table on the left of the screen, the compliance mapping details are presented in the “Base Asset Compliance Details” panel at the right side of the page.

The extensive compliance information is presented, including:

- **Compliance justification** explanation (as specified in OPENCOSS client editor or on this report).
- For the specific justification: the associated artefact or activity.
- For the specific artefact, its associated evidence resource files. These resource files are committed to the appropriate SVN repository. Users can press the [Download] link next to each resource tree node in order to download the specific file from the SVN and view it.
The above tree can be expanded or collapsed quickly to the desired level by pressing buttons above it, allowing tailoring the presented details to the level needed by a user at a given moment. When any of the above tree levels is selected (justification, artefact, or resource), its description and properties are presented in the right-bottom panel of the report.

![Figure 223 - Compliance evidence of the specific baseline asset](image)

Figure 224 - Specific evidence details description presented at the bottom

### 11.2.3 Adding evidence and compliance data

Additionally to browsing the project evidence pieces, the report allows users to **add, modify and remove evidence** resources and **define a compliance mapping**.

“**Base Asset Compliance Details**” panel on the right-hand side of the report contains an **Upload** panel which allows users to add a specific file resource (containing the evidence), specify the associated artefact and define compliance justification text and its type.

After user presses the upload button or drag and drops the file resource to the panel, the following **“New Resource Definition”** dialog appears:
User can enter the desired **compliance justification** in the text area, change the names to be created (default names are suggested) and define the **compliance mapping type**. Additionally it is possible to specify a **SVN URL** location where the evidence file will be committed. User has a possibility to add new location or select already defined one from the select-box.

After pressing **Assign** button, the following actions are performed by the OPENCOSS platform:

- The resource file gets **committed to** the given **SVN** repository so that it is securely stored and can be retrieved on demand.
- A **resource** CCL object (associated with the above file) is created with the specific name.
- An **artefact** CCL object (associated with the specific resource) gets created.
- A **compliance justification**, which maps the artefact to the selected baseline framework items, gets created in OPENCOSS storage.

Additionally, there are **[Modify]** and **[Unassign]** buttons, which allow user to update or revoke the evidence file and compliance mapping created above.
11.2.4 Generation of summary textual report

The interactive mode presented in the preceding chapters is very comfortable for users to browse and filter data, and view their details. Upon each user selection, appropriate details are presented. However, there is often a need to generate an overall report, containing all the information visualized in one place.

This can be easily done using Export to MS Word button. Upon pressing it, a default an docx template report gets filled with the all the Compliance report data presented for the specific safety project.

*Note for OPENCOSS administrators:*

The template docx used for textual report generation can be changed on OPENCOSS server side in order to adjust it the given company standards.

The textual report can then be printed to pdf or on paper, signed digitally or manually, and stored for future reference.
11.3 Compliance Estimation report

Compliance Estimation report is an enhanced version of the Compliance report as it additionally provides “External tool” connectors functionality. There is “External Tools” column presented in the Project Compliance table.
External Tool Connector is a module which is capable of connecting to the defined external tool (e.g. DOORS, JIRA or Bugzilla which may store the project evidence items), send a query to the tool and present up-to-date result returned.

External tool connector can be specified for the given baseline element of the safety standard. For example, if the baseline item says that all the requirements implemented should be tested, a connector to corporate Bugzilla could be added (where the company stores their requirement definitions and their statuses). The connector could query the Bugzilla to learn how many requirements are there with a status marked as Not Tested.

In order to specify a connector for the selected baseline element, [Add Connector] button should be pressed. It presents a connector settings window:

A connector to Bugzilla tool has been implemented as a proof of concept. Additional connectors can be implemented in future release of OPENCOSS platform.

The connector settings are grouped into 2 sections:

- **Instance Settings** - it specifies the external tool connection and the query parameters
- **Instance Execution Result Processing** - it specifies the thresholds for the value returned by the connector querying the external tool. Three value ranges can be defined, for each range green, yellow or red colour would be presented on the report respectively. User can additionally define message for the specific value range.
Pressing [Execute Connector] button triggers the connector to query the external tool and render the result on the report.

To sum up, External tools connectors are modules which can be defined to connect to external tools and present numerical result from a defined data query. The result is presented on the Compliance Estimation report. This way, the report users have up-to-date numeric status regarding their evidence items stored in the external tools. This functionality can be used when the project is in progress (in contrary to the phase when the projects is finishing and is evaluated for compliance) to facilitate automated up-to-date external evidence status presentation.

11.4 Change Impact Analysis

OPENCOSS platform facilitates a prototype of Change Impact Analysis algorithm implementation. It is triggered when any artefact stored in OPENCOSS database is modified by the user. The algorithm traverses all the related artefacts in order to check if they should be marked as affected by the change, depending on the artefact relation type.
11.4.1 Change Impact Analysis in OPENCOSS Client

OPENCOSS Eclipse client editor is the main OPENCOSS tool where user adds, modifies, removes artefacts and relations between them. Similarly it is the place where Impact Analysis algorithm is triggered when any artefact is changed. Similarly the IA results are presented there. This has been described in Impact analysis section.

11.4.2 Change Impact Analysis algorithm

This chapter presents technical details describing how IA algorithm traverses relations between artefacts.

The main pieces of information used by the IA engine are relations between Artefacts objects stored in ArtefactRel CCL entity.

Two Artefacts are considered related when there is an ArtefactRel instance pointing to one of them as a source and another of them as a target. Please note that ArtefactRel has modificationEffect and revocationEffect attributes.

**Note:** An ArtefactRel object for specific two artefacts can be added in the following ways:
- A user can add this entity manually in the Evidence Editor of OPENCOSS platform client
- ArtefactRel entity is added automatically when a parent-child relation is established between two artefacts. When adding artefactPart to parentArtefact, a new ArtefactRel object is created, with modificationEffect=MODIFY and revocationEffect=MODIFY, source pointing to parentArtefact and target to artefactPart.

It has been arranged that a direction of analysis flow is the following: ArtefactRel “target affects the source”. When impact analysis is started:
- It starts from artefactCDOId for the specific EventKind (either Modify or Revoke),
- It looks into the related ArtefactRel (for which the artefactCDOId is a target) object
- It traverses to the artefact pointed by ArtefactRel source
- Depending on the initial EventKind (either Modify or Revoke), it takes the value of modificationEffect or revocationEffect from the ArtefactRel and assumes the appropriate AssuranceAssetEvent on the reached source artefact.

For example, let’s assume that there are the following Artefact and ArtefactRel dependencies:
- ArtefactA ---- ArtefactRelA(ModificationEffect:MODIFY, RevocationEffect:REVOKE) ----> ArtefactB
- ArtefactB ---- ArtefactRelB(ModificationEffect:REVOKE, RevocationEffect:REVOKE) ----> ArtefactC
- ArtefactC ---- ArtefactRelC(ModificationEffect:MODIFY, RevocationEffect:VALIDATE) ----> ArtefactD
ArtefactD ---- ArtefactRelA(ModificationEffect:MODIFY, RevocationEffect:REVOKE) ---> ArtefactE

- The engine starts with EventKind.MODIFICATION for ArtefactA and navigates via ArtefactRelA to ArtefactB and because ArtefactRelA:ModificationEffect equals MODIFY, it reaches ArtefactB with EventKind.MODIFICATION change effect.

**Note:** this change effect event is not saved in storage yet. Now it is only used for further traversal, and will be returned as part of the result of listArtefactsRelationImpacts() method.

- The engine continues from ArtefactB with EventKind.MODIFICATION and navigates to ArtefactC and because ArtefactRelB:ModificationEffect equals REVOKE, it reaches ArtefactC with EventKind.REVOKE change effect.

- The engine continues from ArtefactC with EventKind.REVOCATION and traversal path ends here because of ArtefactRelC:RevocationEffect equals VALIDATE.

- Thus the result is:
  - ArtefactRelA - ModificationEffect.MODIFY
  - ArtefactRelB - ModificationEffect.REVOKE

Above algorithm affects the Artefact lifecycle. States of this lifecycle are presented on the figure below.

![Figure 234 - Artefact lifecycle from the IA point of view](image)

Some of these states require action from user – like “To validate” and “To modify”. To address some restrictions of CCL these two states are recognized by the presence of given event date or lack of it. This signals that action from the user is required and after this action date of the event is set.

### 11.4.3 Impact Analysis result presentation on OPENCOSS server reports

IA-induced user actions that need to be performed in the assurance project are presented on Compliance Estimation Report and Compliance Report.

These reports are described in Compliance Estimation report and Compliance report in general. Hereafter is a description of Impact Analysis results being presented on these reports.
In “Project Compliance” table on the left panel, “IA Status” column presents the status of the specific baseline element from Impact Analysis point of view. The following information is presented:

- Grey color means that there is no artefact compliant to this baseline element thus there is no entity on which AI can work.
- Red color means that some of the artefacts compliant with this baseline element where affected by AI and they require attention from the user. The displayed number represents the amount of such affected artefacts.
- Green color means that there is no action required by IA from user after IA traversed the associated artefacts.

On “Base Asset Compliance Details” panel the compliant artefacts and their details are presented. The information also includes IA results in case when AI algorithm detected that some action need to be performed by a user.

As it regards the above screenshot, IA execution resulted in detection of two actions required by the user: “To validate” and “To modify”. This information is presented next to the respective artefacts.

When the user takes the according measures (i.e. validates or modifies the respective artefacts) he can simply clicks “Modified” and “Validated” action buttons to report that the requested activity has been performed.

11.5 Gap Analysis report - Compliance Assessment and Evidence Evaluation

Gap Analysis report facilitates the following pieces of functionality:

- Compliance Assessment by viewing a Gap Analysis
- Viewing Evidence Evaluation results
In order to view Gap Analysis report, please go to OPENCOSS platform web server page in your web browser at http://<OPENCOSS-SERVER-HOST-NAMESPACE>:8080/ and select Reports > Gap Analysis report from menu.

![Gap Analysis report](image)

**Figure 236 - Gap Analysis report**

### 11.5.1 Gap Analysis report core functionality

Gap Analysis report presents summary and details regarding specific assurance project base artefacts and base activities and their compliance mapping to the actual evidence and activities. For the selected assurance project its baseline frameworks are presented in a select box.

![Baseline frameworks for the specific assurance project](image)

**Figure 237 - Baseline frameworks for the specific assurance project**

After choosing the specific baseline framework, the following gap analysis data is presented:

- In a left pane, called “Project Baseline Compliance”, base artefacts and base activities of the selected project baseline framework are shown. They are displayed in a tree structure to express parent-child hierarchy of these items. For each base artefact or base activity the total numbers of fully- and partially-compliant assets are presented. The “Project Baseline Compliance” table can be filtered to show only base artefacts or base activities and can be sorted by any column.
When user selects the specific cell in the left pane table (e.g. specific base artefact or the number of fully- or partially-compliant assets), the right panel is refreshed with the details regarding the selected item.

For specific base artefact or base activity selected in the left panel, the right details panel presents the following information:

- Summary of the specific base artefact or base activity compliance mapping
- For the given base artefact and base activity: Compliance Justification elements from its compliance mapping
- For each Compliance Justification: the actual assets i.e. artefacts or activities
- For each artefact or activity: its description and properties are presented on the tooltip
- For each artefact or activity: its evaluation is shown - in case the asset has been evaluated.

The “Compliance Details” tree can be expanded to the specific level by pressing links at the top of the panel.
For example, if user is interested only in justification or assets or their evaluation, he can press [Expand to Justification], [Expand To Asset] or [Expand to Evaluation] links respectively. The details tree is expanded accordingly.

### 11.5.2 Viewing Evidence Evaluation in Gap Analysis report

In case a specific evidence item has been evaluated in OPENCOSS client tool, its evaluation data are shown on the Gap Analysis report.

All the evaluation properties (Id, Criterion, Evaluation result, etc.), and the evaluation date and time are presented.

![Figure 241 - Evidence evaluation details](image)

### 11.6 Metrics reports

This chapter presents the implemented metric reports, their functionality and layout.
11.6.1 Metrics Estimation Report

The Metrics Estimation Report can be accessed via the following OPENCOSS web server menu item:

![Menu item directing to “Metrics Estimation report”](image)

When a specific OPENCOSS safety project is selected in the top panel, its defined baselines are presented in the middle panel select box, as shown in Figure 211. The report data is divided into two panels. The first one, on the left, is a static menu panel in which the user can select a type of metrics to analyse.

![Metrics Estimation report](image)

Once a specific metrics is selected on the Metrics menu, the metrics menu details are presented in the right part of the report with a description of the main goal and all the different types of charts related to that metric.

Also, the user has the option to the selected metric to a Word Document with more detailed information. As a small example, the following figure:
11.6.2 Equivalence Map Report

The Equivalence Map Report can be accessed via the following OPENCOSS web server menu item as showed beforehand.

In this case the equivalence metrics are between two Reference frameworks, not specific OPENCOSS safety project needs to be selected in the top panel. The only possible configurations are between reference frameworks as shown in figure below

![Selection of reference frameworks](image)

After the selection, the metrics of the equivalence maps and a detailed description are showed on the screen. There is also a possibility to export this information to a document.
11.7 Administration web GUI

OPENCOSS server web pages provide a few basic pages for server administration. They are accessible through “Administration” menu.

11.7.1 Projects Administration

This administration page facilitates project editing basic actions. For more advanced project editing functionality, OPENCOSS Eclipse Client Editor should be used.

The page allows the following basic actions:

- Project name and description editing.
- Project baseline name editing.
  
  Note that only one baseline of the project is presented.
- Project baseline element editing.
  
  User can add, remove and modify baseline elements.
  
  For the specific baseline element, its name and description can be specified.
11.7.2 Create Sample Data

This OPENCOSS server administration web page provides a functionality, which allows user to generate sample data in the database. The sample data can be generated for example in order to demonstrate Gap Analysis report.

OPENCOSS user can generate the following sample data for the selected assurance project:

- baseline framework
- base artefacts
- base activities
- artefacts and activities being mapped with compliance mapping to the base artefacts and base activities.
After pressing [Generate Data], sample data will be created and inserted into the selected assurance project.

11.7.3 Configuration Settings

This page presents the main configuration settings of the OPENCOSS server. These settings are stored and can be modified on OPENCOSS server host, in `opencoss-properties.xml` file which is present in the operating system user home directory.