

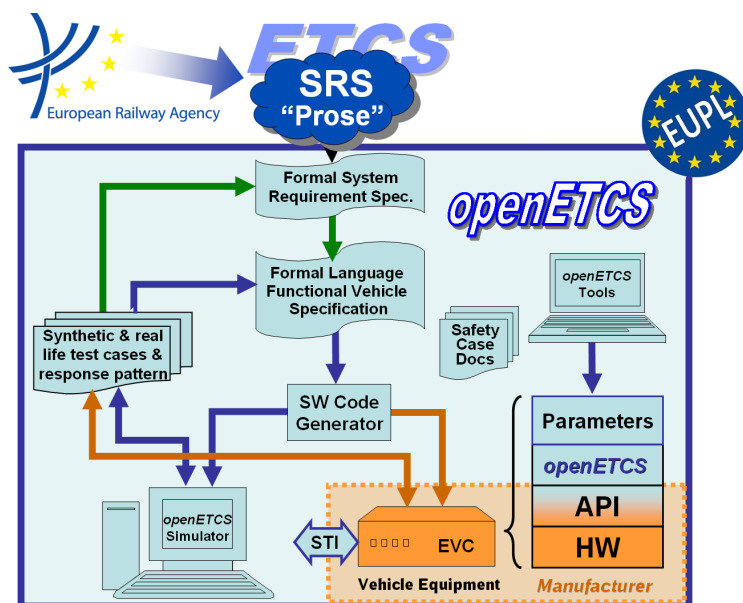
Work-Package 7: “Toolchain”

Traceability Architecture alternate solutions

WP7 Proposition

Cecile Braunstein and Raphaël Faudou

November 2015



Funded by:


 Federal Ministry
 of Education
 and Research

 Région de
 Bruxelles-
 Capitale

 GOBIERNO
 DE ESPAÑA
 MINISTERIO
 DE INDUSTRIA, ENERGÍA
 Y TURISMO

This page is intentionally left blank

Work-Package 7: “Toolchain”**OETCS/WP7/07.3.5
November 2015**

Traceability Architecture alternate solutions

WP7 Proposition

Document approbation

Lead author:	Technical assessor:	Quality assessor:	Project lead:
location / date	location / date	location / date	location / date
signature	signature	signature	signature
Raphael Faudou (University Bremen)	()	()	()

Cecile Braunstein

University Bremen

Raphaël Faudou

Samares Engineering on behalf of ENSEEIHT

OpenETCS : Position Paper on traceability

Prepared for openETCS@ITEA2 Project

Abstract: This document presents alternate solutions to the current tool chain traceability architecture.

Disclaimer: This work is licensed under the "openETCS Open License Terms" (oOLT) dual Licensing: European Union Public Licence (EURL v.1.1+) AND Creative Commons Attribution-ShareAlike 3.0 – (cc by-sa 3.0)

THE WORK IS PROVIDED UNDER openETCS OPEN LICENSE TERMS (oOLT) WHICH IS A DUAL LICENSE AGREEMENT INCLUDING THE TERMS OF THE EUROPEAN UNION PUBLIC LICENSE (VERSION 1.1 OR ANY LATER VERSION) AND THE TERMS OF THE CREATIVE COMMONS PUBLIC LICENSE ("CCPL"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS OLT LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. TO THE EXTENT THIS LICENSE MAY BE CONSIDERED TO BE A CONTRACT, THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

<http://creativecommons.org/licenses/by-sa/3.0/>
<http://joinup.ec.europa.eu/software/page/eupl/licence-eupl>

Table of Contents

Document Information

Document information	
Work Package	WP7
Deliverable ID or doc. ref.	O7.3.5
Document title	Traceability Architecture in OpenETCS
Document version	00.02
Document authors (org.)	Cécile Braunstein (Uni.Bremen)

Review information	
Last version reviewed	
Main reviewers	

Approbation			
	Name	Role	Date
Written by	Cécile Braunstein	WP7-T7.3 Sub-Task Leader	06.02.2014
Approved by			

Document evolution			
Version	Date	Author(s)	Justification
00.00	17.12.2014	C. Braunstein	Document creation
00.00	23.10.2015	R. Faudou	Precisions concerning OpenETCS requirements and models and update of tool chain traceability requirements

1 Introduction

1.1 Document purpose

This document presents a proposition concerning support of openETCS project traceability activity by openETCS tool chain. This proposition is a consensus between traceability needs, project policy (open source solutions), time and efforts (focus on existing tools and features) and risks about tool maturity.

1.2 Document scope

This proposition is based on the needs and priorities about traceability, captured from different work packages (mainly WP3 and WP4) during October 2015, either from interviews or from reading of available project documents, especially:

- Project Quality Assurance Plan - D1.3 - ?.
- Definition of OpenETCS Development Process - D2.3a v02 - ?
- OpenETCS Architecture and Design Specification - D3.5.0 - ? and D3.5.1 - ?
- Safety plan - D4.2.3 - ?

The proposition done in this document is based on existing tools and existing tool capabilities with verified maturity. Document will explain limits of current proposition and will suggest axes to investigate in the future of project (roadmap). But detailed evaluations of tools and features based on those investigations is not part of this document.

1.3 Document organization

Chapter 2 recalls openETCS project scope, development process, different needs and requirements to address, architecture and design specification.

Chapter 3 highlights main priorities concerning traceability and detailed expectations about traceability activities to support (including design, code, verification and validation).

Chapter 4 defines tool chain requirements concerning traceability.

Chapter 5 presents current traceability architecture defined to support tool chain requirements identified in chapter 4. This architecture is based on use of existing mature tools.

Chapter 6 summarizes limits of the current traceability architecture and suggests axes to investigate in order to improve traceability support by the OpenETCS tool chain.

1.4 Investigation axes for improvements

Requirement management and traceability tools Concerning requirement management and traceability there exist a lot of solutions. If we focus on open source solutions and especially those able to integrate into Eclipse platform, we find:

- Eclipse Requirement Modeling Framework (RMF). As mentioned on Eclipse web site, RMF is a framework to manage textual requirements through ReqIF standard requirement exchange format. There is a powerful GUI called "ProR" to configure, visualize and edit ReqIF requirements in RMF. RMF/ProR is the natural choice for OpenETCS to support requirement management.

Alternatively to deal with the closed source path the requirement management may be done by ReqTify Gateway of SCADE (also called "RM Gateway").

ProR can also be used to support traceability between requirements (native functionality) and between requirements and Papyrus models through a "proxy" connector.

- PolarSys ReqCycle. It is a solution that focuses on requirement traceability. It can allow managing requirements (creation, edition, queries) with classification (scopes) and custom requirement data models but with very limited requirement editor. Strength is its ability to capture and aggregate traceability links coming from different sources (EMF models, C code, Java code, SysML models...) and to define own traceability links based on custom link types with potential attributes.

If we extend search to proprietary solutions, we find IBM DOORS and IBM DOORS Next requirement management products and Dassault Systems ReqTify product as leading solutions to support respectively requirement management and requirement traceability.

This second solution consists in using only one centralized Requirement database (as in solution 1) managed by one eclipse-based requirement management solution (ProR) and only one eclipse-based technology to support requirement traceability for all models (Papyrus and SCADE): PolarSys ReqCycle.

With that approach, there is still one master reference requirements data base entirely managed by ProR. Requirement database is initialized by import from the subset-026 word document, as in solution 1 (see ??) and by import from OpenETCS API. All new OpenETCS requirements are added in this requirement database through ProR tool.

Traceability is managed by ReqCycle tool. In order to ease visual selection of requirements in ReqCycle, there is a copy (could be a link) of reference requirements hierarchy into a ReqCycle database. Then ReqCycle manages links with Papyrus and links with SCADE. Finally, traceability can be exported by ReqCycle and processed by Gendoc tool to deliver documentation.

Interactions through the tool chain are detailed in figure ?? below with focus put on traceability between requirements and functional model (traceability with SysML model, creation of tests and export of traceability are not shown here).

Note: this solution requires synchronization from ProR to ReqCycle each time requirements change in the reference.

Most complex commands are detailed in next paragraphs.

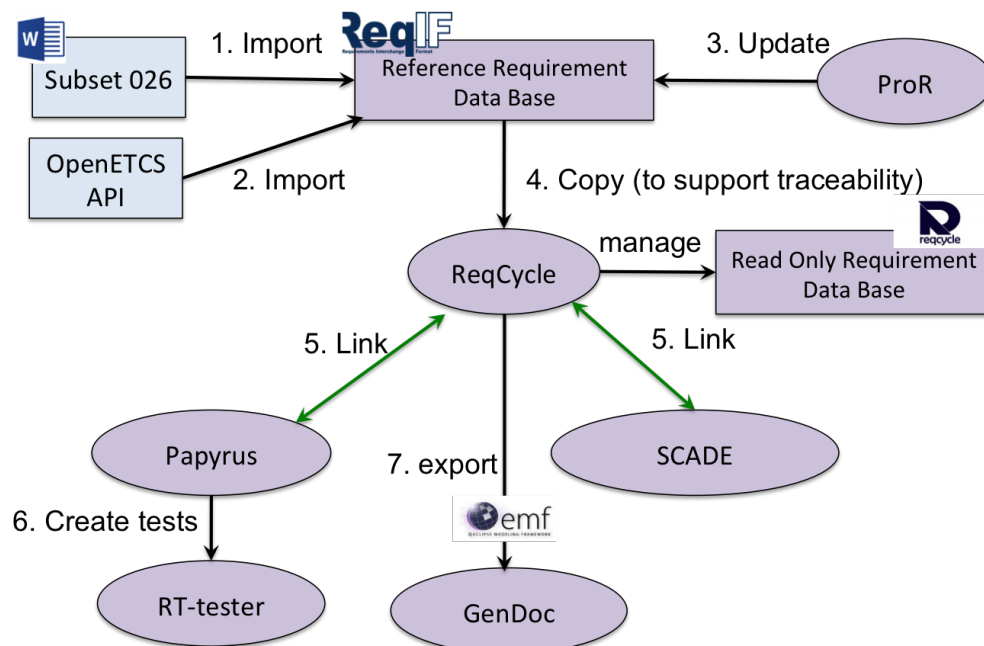


Figure 1. Traceability architecture second solution

1.4.1 Subset-026 import

The subset-026 import can be realized by two means:

- Either by direct import from the subset-026 word document (ReqCycle Document import connector)
- Or by a two steps process with transformation from word document to .ReqIF files (1.B1 path in the figure) and then import of those .ReqIF files into ReqCycle requirement database (1.B2 path).
 - 1.A1 transformation can be realized with same script than the one described in first traceability solution - see ??
 - 1.A2 transformation (import) can be done by ReqCycle ReqIF import connector.

1.4.2 Creation of additional OpenETCS requirements

ProR can create new requirements by decomposition or by derivation of existing requirements. In case of derivation, engineering level (scope) changes (for instance from System to OBU or to OBU Kernel).

Regularly, when there have been new requirements created, there must be a synchronization from ReqIF reference requirement database to ReqCycle internal database so that ReqCycle can manage traceability to those requirements. This synchronization is done by a ReqCycle "import or update" command from .reqIF files: first time it is an import (creation of requirements) and next times this is an update of existing requirements with ability to highlights associated traceability links (that have to be checked).

Note: in the future, we could have a link (reference) between ProR requirement database (ReqIF format) and ReqCycle database, rather than a copy.

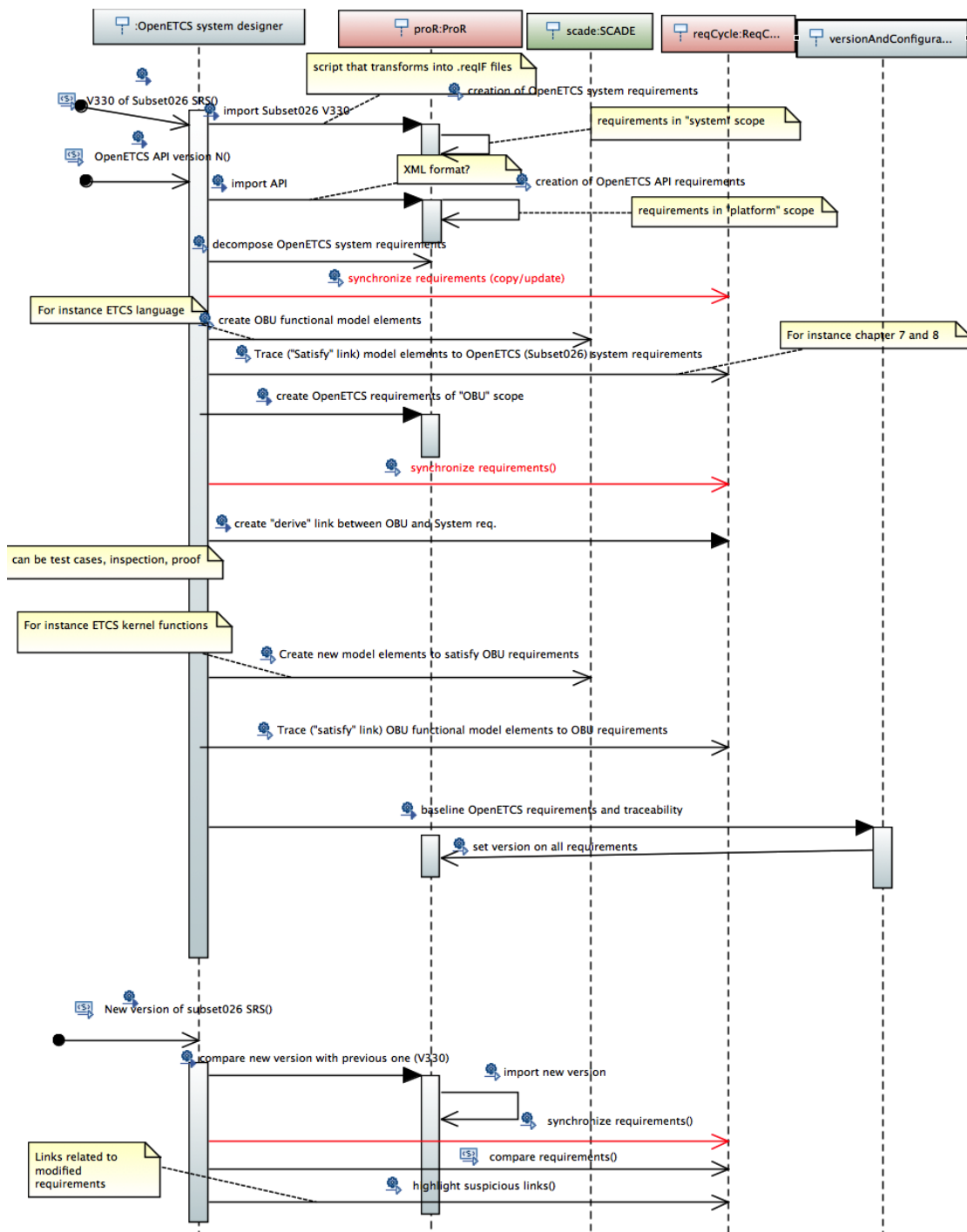


Figure 2. second solution - interactions between tool chain components

We could also have traceability driven by ReqCycle but selection of requirement done in ProR tool.

1.4.3 Creation of links between SysML model elements and requirements

ReqCycle provides a view that allows creating a link between two selected objects (traceability Creator). See figure ??.

Concerning selection of requirements, ReqCycle provides a view (Requirement View) from which it is possible to select one source object. Once selection is done, it can be propagated in traceability creator view (source area).

Concerning selection of SysML model elements, Papyrus provides a view with model elements (Model explorer) from which it is possible to select a destination object. Once selection is done, it can be propagated to the traceability creator view (destination area).

When both source and destination objects are selected, ReqCycle provides a command (button) to create links between selected objects if a link type definition is compliant with such source and destination (Refinement in the illustration on figure ?? below).

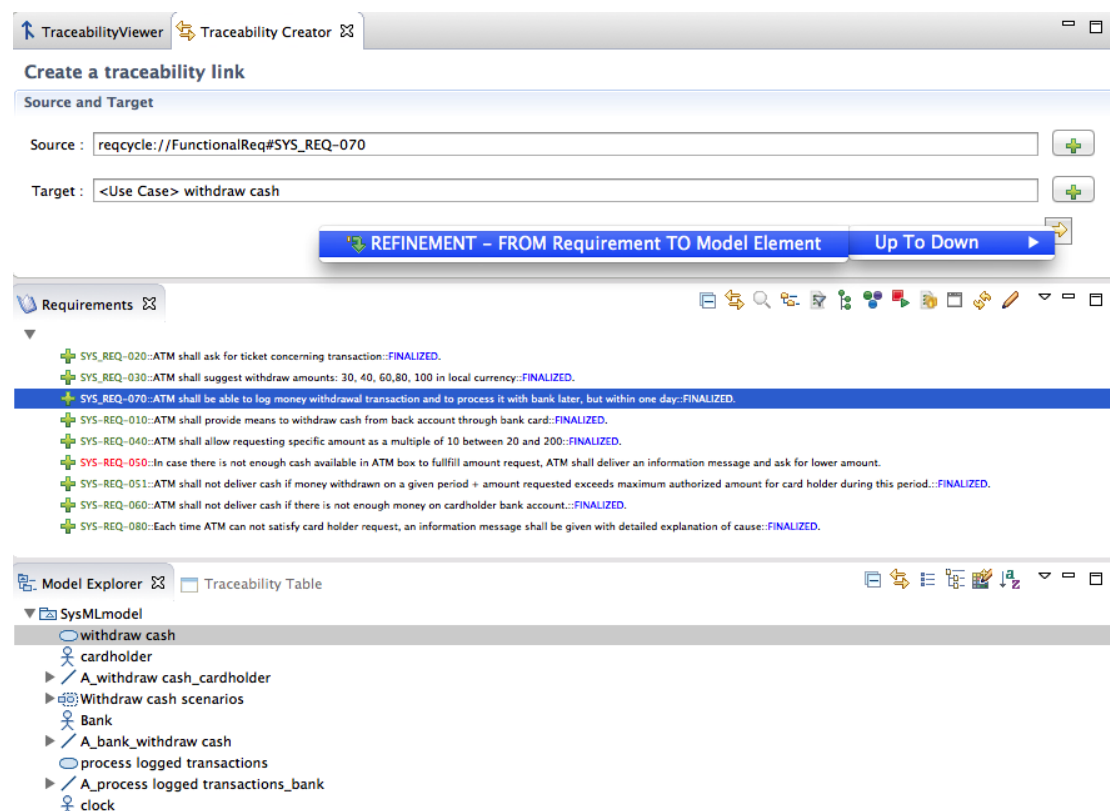


Figure 3. ReqCycle - traceability creator

1.4.4 Creation of links between SCADE model and requirements

There are two approaches that can be used:

1. use same approach than for creation of trace links between SysML model elements and OpenETCS requirements (see ??).

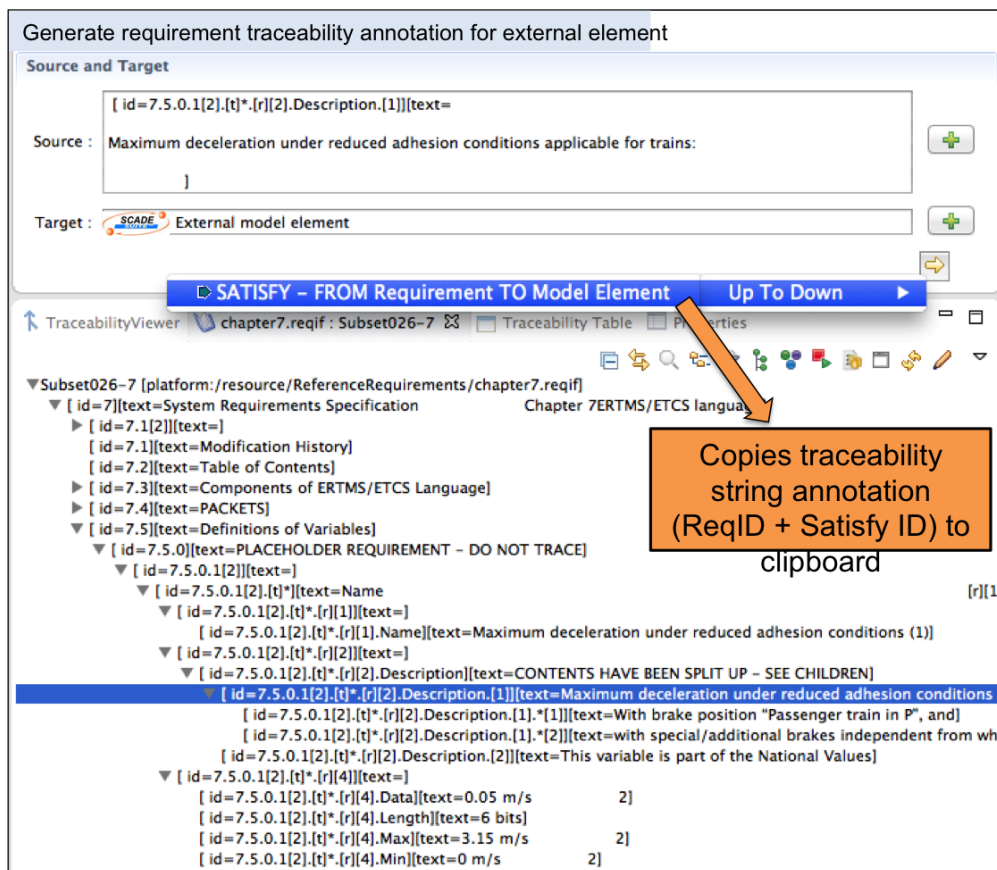


Figure 4. ReqCycle - generate traceability annotation for external model element (SCADE for instance)

Note: there is some investigation to check that it is (easily) feasible to propagate selection of a SCAD model element in ReqCycle traceability creator view (source or destination area).

2. use SCAD comment area available for any model element, fill it with traceability link data and let ReqCycle capture those traceability links by analysing SCAD model file (dedicated SCAD traceability analysis engine to implement).

In order to avoid mistakes on editing link annotation in comment area in SCAD, idea is to generate requirement traceability link annotation from ReqCycle command ("Generate requirement traceability annotation for external element") and copy it into the clipboard.

Then, in SCAD, system designer has just to "paste" string (from clipboard) in the concerned model element comment area and let ReqCycle analyse traceability: ReqCycle will parse SCAD model file, will capture all comments with associated model elements and will build the traceability links from requirement ID, trace link type and SCAD model element.

Figure ?? illustrates such command.

1.5 Aggregation of traceability links and export

It is done by ReqCycle that has all information (copy of the requirement database and associated traceability between requirements + all traceability links with models). It is exported into EMF format that can be processed by Gendoc to render expected documentation including traceability.

1.5.1 Third alternative traceability solution: ReqCycle

This third solution consists in using only one centralized Requirement database (as in previous solutions) managed by one eclipse-based solution (ReqCycle) also used to support requirement traceability for all models (Papyrus and SCADE).

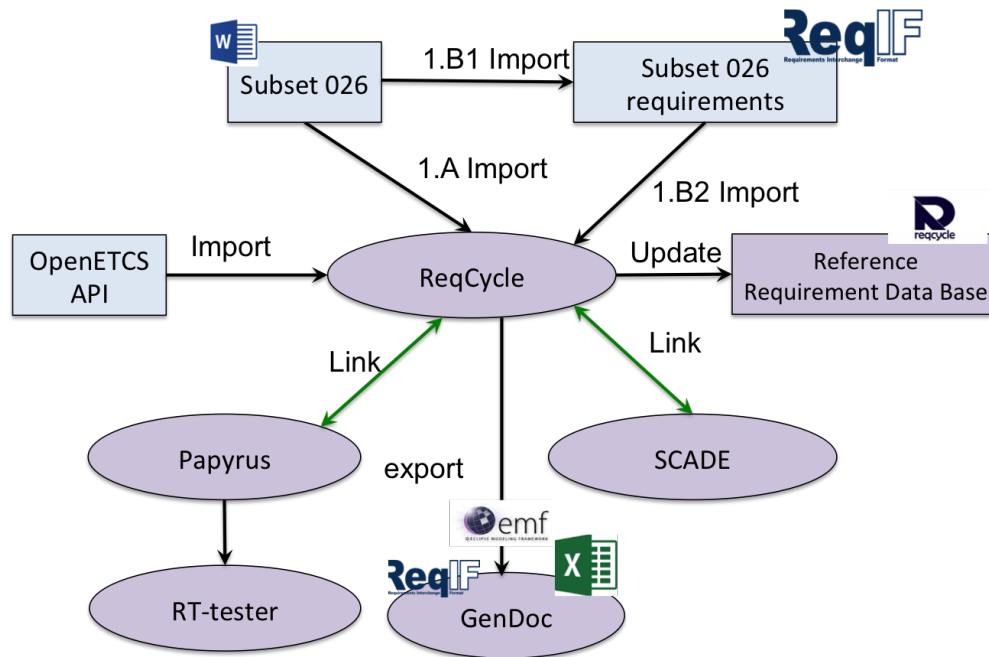


Figure 5. Traceability architecture third solution with ReqCycle only

With that approach, there is still one master reference requirements data base entirely managed by ReqCycle. Requirement database is initialized by import from the subset-026 word document through two possible means (see next section) and by import from OpenETCS API. All new OpenETCS requirements are added in this requirement database through ReqCycle tool.

Traceability is managed by ReqCycle tool. In order to ease visual selection of requirements in ReqCycle, there is a copy (could be a link) of reference requirements hierarchy into a ReqCycle database. Then ReqCycle manages links with Papyrus and links with SCADE. Finally, traceability can be exported by ReqCycle and processed by Gendoc tool to deliver documentation.

Most complex commands are detailed in next paragraphs.

1.5.2 Subset-026 import

The subset-026 import can be realized by two means:

- either by direct import from the subset-026 word document (ReqCycle Document import connector)
 - else by a two steps process with transformation from word document to .ReqIF files (1.B1 path in the figure) and then import of those .ReqIF files into ReqCycle requirement database (1.B2 path).
- 1.A1 transformation can be realized with same script than the one described in first traceability solution - see ??
 - 1.A2 transformation (import) can be done by ReqCycle ReqIF import connector.

1.6 Creation of additional OpenETCS requirements

ReqCycle provides a "local" connector to create new requirements and those requirements can be linked to existing other requirements through different kinds of links (derivation, refinement, decomposition). Those link types have to be defined first in ReqCycle configuration.

To complete

1.7 Creation of links between SysML model elements and requirements

Same solution than in second solution. see ??

1.8 Creation of links between SCADE model and requirements

Same solution than in second solution. see ??

1.9 Aggregation of traceability links and export

Same solution than in second solution. see ??

2 Tool evaluations

Will be done in a separate document.

2.0.1 TODO ReqCycle Evaluation

2.0.2 TODO Reqtify Evaluation