*Report*



**OpenES Project**

***Project Full Title:***

**“Open ESL Technologies for Next Generation Embedded Systems”**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | WP no. | | Deliverable no. | Lead participant |
| **WP1** | | **D1.2.2.b** | **ST** |
| **System modeling and uses cases v2** | | | | |
| Prepared by | | **ST: Alain CLOUARD, Sebastien REVOL (editor) CEA: Arnaud CUCCURU TCS: Aurélien BERHAULT CISC: Ralph WEISSNEGGER** | | |
| Issued by | | **Sebastien REVOL (ST)** | | |
| Document Number | | **OpenES/WP 1/D1.2.2.b** | | |
| Nature  Dissemination Level | | **Report**  **CO** | | |
| Submission Date | | **2014-10-03** | | |
| Due Date | | **2014-09-30** | | |
| **Project co-funded by eureka-logo.jpg CATRENE Programme (2013-2016)** | | | | |

© Copyright 2013 STMicroelectronics srl., Thales Communications & Security and Thales Research & Technology SA, NXP Semiconductors, Synopsys, CEA-LIST, Magillem Design Services SAS, CISC Semiconductor GmbH, Docea Power, Vector Fabrics B.V., University of Technology Eindhoven, UJF-Verimag, European Electronic Chips & Systems design Initiative.

**History of Changes**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ED. | REV. | DATE | PAGES | REASON FOR CHANGES |
| AP | 1.0 | 2013-10-09 | 10 | Initial version |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

* The “Revision” is the document version number:
* V0.5 – project internal review
* V0.7 – modifications after project internal review
* V1.0 – initial draft released to the consortium
* V2.0 – final version delivered to the CATRENE review.

**<config services=`ExtConfigService`>**

**<output path='${outputPath}/Deliverable\_1\_3\_\_1\_4.docx' />**

**<param key='model' value='${inputPath}'/>**

**</config>**

**<config services=`ExtLogger`/><drop/>**

**<context model='${model}' />**

<bookmarks>

<alias source='reqID' target='[req.getReqID()/]'/>

</bookmarks>

Contents

[1 Introduction 4](#_Toc369093560)

[2 General formatting rules 5](#_Toc369093561)

[3 Empty Chapter 8](#_Toc369093562)

[4 References 9](#_Toc369093563)

[A. Nothing interesting here 10](#_Toc369093564)

# T1.3

The next pages contain T1.3 specific generated sections

# OSMK requirements traceability

The goal this last section is to provide a demonstration that T1.3 specific requirements have been addressed and illustrated by at least one of the example models delivered with this report. Hence the remaining part this section is fully generated from the “traceability” models contributed by each of the partners (cf D1.2.3a for the flow description).

The first section contains a summary table, whose generation was possible thanks to the “RequirementScope” OSMK annotation, and coming from comments specified in the input excel table.

Then, each of the satisfied requirement is reminded in this document, with link to the satisfying model and the reason why we considered that it was satisfied.

## Requirement analysis summary

<fragment name=’getModelLink’><arg name=’abs’ type=’Abstraction’/><arg name=’uri’ type=’String’/>[if (uri.contains('CISC'))]CISC Model[elseif (uri.contains('OSMK'))]OSMKProfile[elseif (uri.contains('requirements'))]Requirement Model[elseif (uri.contains('CEA'))]CEA Model[elseif (uri.contains('TCS'))]TCS Model[elseif (uri.contains('ST'))]ST Model[elseif (uri.contains('common'))]Common Model[/if]</fragment>

<gendoc>

|  |  |  |
| --- | --- | --- |
| Requirements satisfied by T1.3 | [getT13SatisfiedRatio()/] | [for (req : Class | getT13SatisfiedReqs ())separator (‘,’) ] [[req.getReqID()/]](#reqID)[/for] |
| Requirements that should be satisfied by task T1.3 | [getT13NotSatisfiedRatio()/] | [for (req : Class | getT13NotSatisfiedReqs ())separator (‘,’) ] [[req.getReqID()](#_Requirement__[req.getReqID()/])/][/for] |
| Requirements related to task T1.2 | [getT12Ratio()/] | [for (req : Class | getT12Reqs ())separator (‘,’) ] [req.getReqID()/][/for] |
| Requirements related to task T1.4 | [getT14Ratio()/] | [for (req : Class | getT14Reqs ())separator (‘,’) ] [req.getReqID()/][/for] |
| Requirements that are out of OSMK scope | [getNotOSMKRatio()/] | [for (req : Class | getNotOSMKReqs ())separator (‘,’) ] [req.getReqID()/][/for] |

## Satisfied requirements

[for (req : Class | getT13SatisfiedReqs ())]<drop/>

### Requirement [req.getReqID()/]

Requirement definition :

*“[req.getReqText()/]”*

[if(not req.getReqComment().oclIsUndefined())]<drop/>

**Comment :** [req.getReqComment()/]

[/if]<drop/>s

Satisfaction traceability :

[for (abs : Abstraction | req.getSatisfyLinks())]

[getTextIntro(i)/] satisfied in [abs.getModelLink(abs.getSatisfyingModel())/] by :

<list><drop/>

[for (elem : NamedElement | abs.getSatisfyingElements())]<drop/>

* [elem.getSatifyingElementText()/]

[/for]

</list>

**Rational :** [abs.ownedComment->asSequence()->at(1).\_body/]

[/for]<drop/>

[/for]<drop/>

## Not satisfied requirements

[for (req : Class | getT13NotSatisfiedReqs ())]<drop/>

### Requirement [req.getReqID()/]

Requirement definition :

*“[req.getReqText()/]”*

[if(not req.getReqComment().oclIsUndefined())]<drop/>

**Comment :** [req.getReqComment()/]

[/if]<drop/>s

[/for]<drop/>

</gendoc>

# T1.4

The next pages contain T1.4 specific generated sections

# OSMK requirements traceability

The goal this last section is to provide a demonstration that T1.4 specific requirements have been addressed and illustrated by at least one of the example models delivered with this report. Hence the remaining part this section is fully generated from the “traceability” models contributed by each of the partners (cf D1.2.3a for the flow description).

The first section contains a summary table, whose generation was possible thanks to the “RequirementScope” OSMK annotation, and coming from comments specified in the input excel table.

Then, each of the satisfied requirement is reminded in this document, with link to the satisfying model and the reason why we considered that it was satisfied.

## Requirement analysis summary

<gendoc>

|  |  |  |
| --- | --- | --- |
| Requirements satisfied by T1.4 | [getT14SatisfiedRatio()/] | [for (req : Class | getT14SatisfiedReqs ())separator (‘,’) ] [[req.getReqID()/]](#reqID)[/for] |
| Requirements that should be satisfied by task T1.4 | [getT14NotSatisfiedRatio()/] | [for (req : Class | getT14NotSatisfiedReqs ())separator (‘,’) ] [[req.getReqID()](#_Requirement__[req.getReqID()/])/][/for] |
| Requirements related to task T1.2 | [getT12Ratio()/] | [for (req : Class | getT12Reqs ())separator (‘,’) ] [req.getReqID()/][/for] |
| Requirements related to task T1.3 | [getT13Ratio()/] | [for (req : Class | getT13Reqs ())separator (‘,’) ] [req.getReqID()/][/for] |
| Requirements that are out of OSMK scope | [getNotOSMKRatio()/] | [for (req : Class | getNotOSMKReqs ())separator (‘,’) ] [req.getReqID()/][/for] |

## Satisfied requirements

[for (req : Class | getT14SatisfiedReqs ())]<drop/>

### Requirement [req.getReqID()/]

Requirement definition :

*“[req.getReqText()/]”*

[if(not req.getReqComment().oclIsUndefined())]<drop/>

**Comment :** [req.getReqComment()/]

[/if]<drop/>s

Satisfaction traceability :

[for (abs : Abstraction | req.getSatisfyLinks())]

[getTextIntro(i)/] satisfied in [abs.getModelLink(abs.getSatisfyingModel())/] by :

<list><drop/>

[for (elem : NamedElement | abs.getSatisfyingElements())]<drop/>

* [elem.getSatifyingElementText()/]

[/for]

</list>

**Rational :** [abs.ownedComment->asSequence()->at(1).\_body/]

[/for]<drop/>

[/for]<drop/>

## Not satisfied requirements

[for (req : Class | getT14NotSatisfiedReqs ())]<drop/>

### Requirement [req.getReqID()/]

Requirement definition :

*“[req.getReqText()/]”*

[if(not req.getReqComment().oclIsUndefined())]<drop/>

**Comment :** [req.getReqComment()/]

[/if]<drop/>s

[/for]<drop/>

</gendoc>

1. UML modelling environment setup

In the context of Task 1.2,1.3,1.4 to fulfil the requirements mentioning the main System Level modelling environment should be open and free, we have chosen to rely on the Eclipse-based UML modelling tool named Papyrus ([https://www.**eclipse**.org/**papyrus**](https://www.eclipse.org/papyrus/)/). This tool is the official UML editor of the eclipse community, and its project leader is the CEA laboratory involved in several tasks of OpenES (T1.3 leader). Moreover, this tool is also used for products modelling and development at STMicroelectronics.

To setup this tool, several steps are required, which will be detailed further:

* Download a base eclipse installation
* Install Papyrus and additional MARTE Profile
* Install additional profiles (from ST) for IPXACT and IP API related notations
* Import the model project in Papyrus

## Eclipse download and install

To run the latest version of Papyrus (1.0.2) delivered as plugins on the official Eclipse repository, it is first required to install a base bundle of Eclipse Luna.

Several distributions are available for the official eclipse website : <http://www.eclipse.org/downloads/> . The most appropriated distribution, already including some required dependencies is named “*Eclipse Modelling Tools*”.

It can be directly downloaded from this page: <https://www.eclipse.org/downloads/packages/eclipse-modeling-tools/lunasr1>, choosing the download link according the target OS architecture to which the package is intended to be installed (Linux/Windows/Mac, 32/64 bits).

Once downloaded, just unpack the archive.

Eclipse can be directly started with the ***eclipse/eclipse.exe*** executable contained in the target directory.

Notice that a Java Runtime Environment is required to run eclipse. If not present on the target machine, it can be downloaded and installed from: <http://www.java.com/fr/download/>

## Papyrus install

Papyrus is not present by default in the modelling package. Once this eclipse bundle is downloaded, installed and started, you can install additional plugins from eclipse user interface with the following steps described below. Notice that it requires eclipse to connect to Internet. If you use a proxy, you should configure it accordingly.

* Step0 (optional): if you connect to internet through a proxy, configure proxy information from Window ->Preferences -> General ->Network connections
* Step1: click on Help -> Install new software (Figure 1)

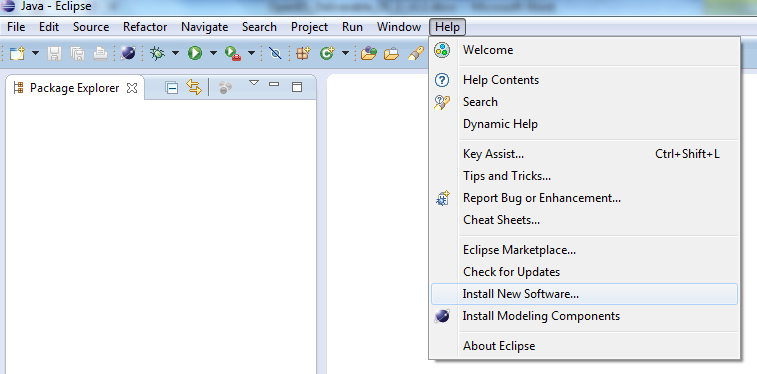


Figure 1 : Install new Software menu

* Step2 : Select Kepler main repository

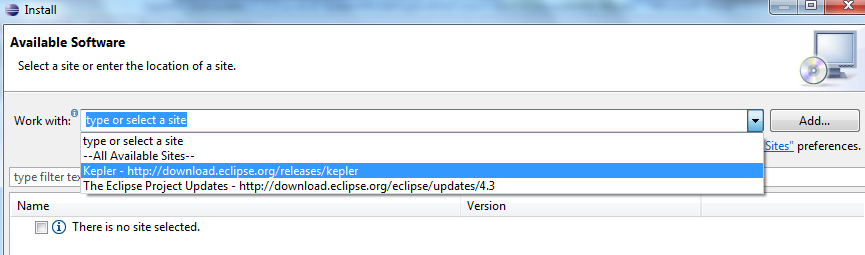


Figure 2 Select Luna Repository

* Step3 : Browse and Select Papyrus in Modelling sub project and click “next”:

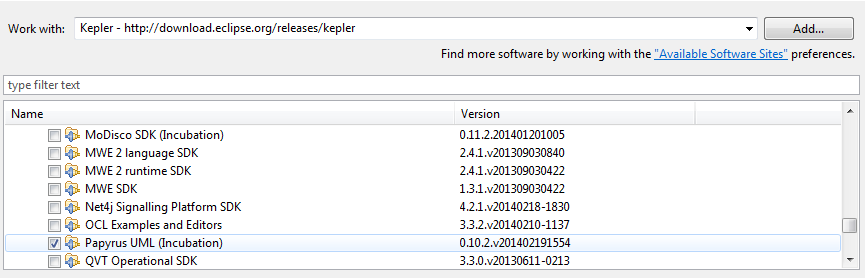


Figure 3 Select Papyrus in Modelling subproject

* Step4 : Accept licence agreement (Eclipse Plugin Licence), install bundles and restart eclipse.
* Step 5 : Installation of Papyrus extensions (MARTE profile): after eclipse restart, enter in workbench and right click on help-> install Papyrus additional components. Select MARTE and click on finish. Restart eclipse.

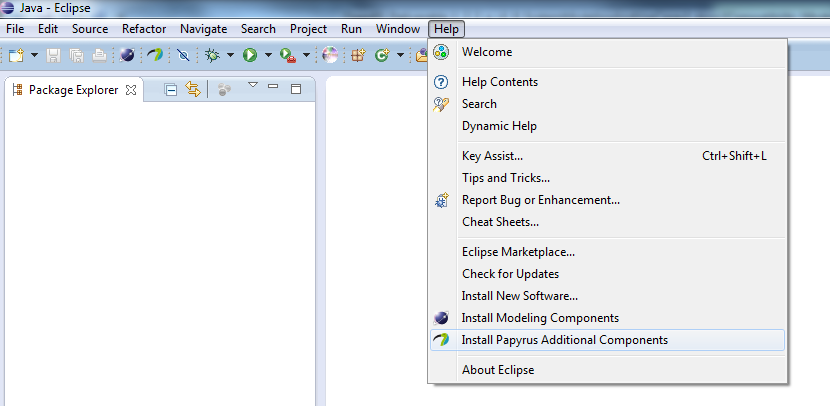


Figure 4 Install Papyrus Additional Components

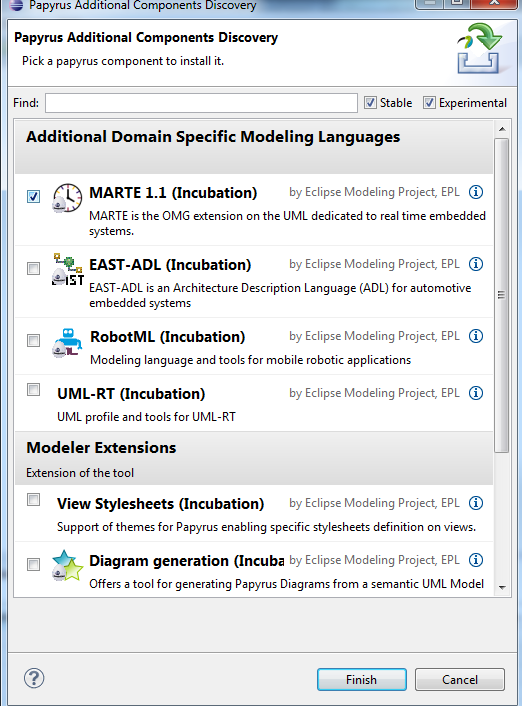


Figure 5 Select MARTE and click on finish

## Install additional profiles (from ST) for IPXACT and IP API related notations

As an input to the Task T1.2, ST is providing an in house UML profile allowing to describe IPXact related information. This profile is packaged in an additional eclipse plugin that should be installed in eclipse installation. It can be simply done by the following steps:

* **Copy** the archive **com.st\*.jar** files from /Deliverables/WP1/OSMKPlugins on OpenES FTP into **eclipse/dropins** directory.
* Restart eclipse.

## Import the model in the workspace

The model of this deliverable is packaged as a zipped eclipse project. It can be easily imported into eclipse with the following procedure:

* Click on File->Import

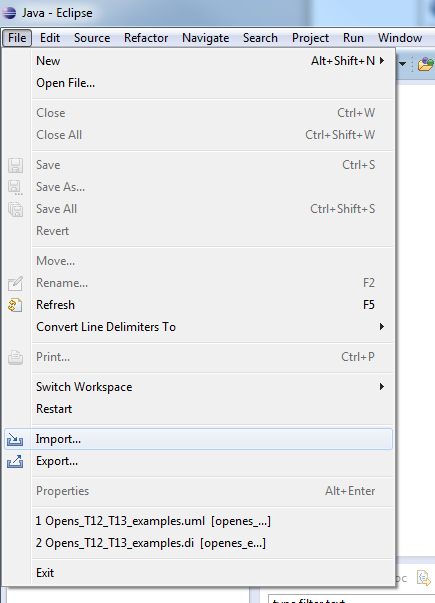


Figure 6 Click on File -> Import

* Select General/Existing Project into workspace

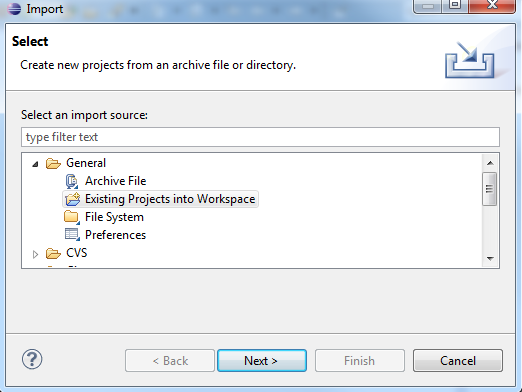


Figure 7 Select General/Existing projects into Workspace

* Select **Archive File** checkbox (! Important, don’t select ‘root directory’!) and browse to the OpenESExamplesD122a-D13a.zip archive. Then click on finish.

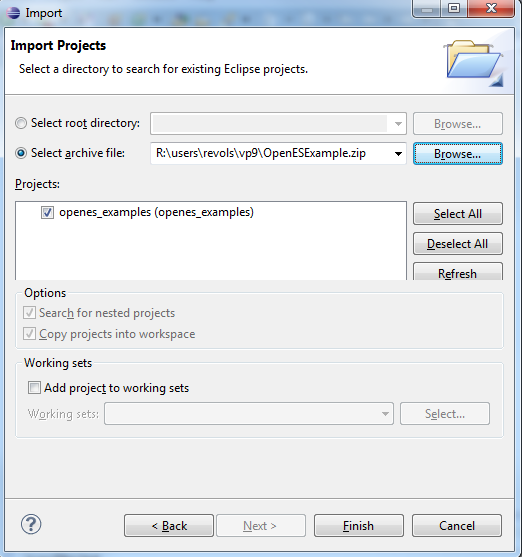


Figure 8 Zipped project selection

* Activate the Payrus perspective : click on window -> open perspective ->others and select Papyrus

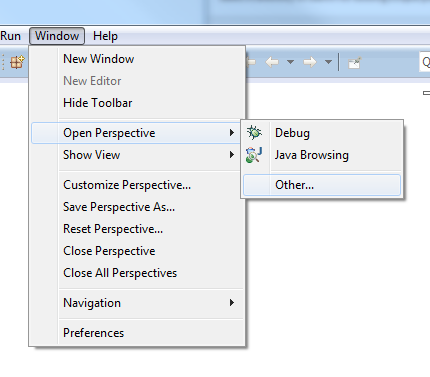


Figure 9 Path to activate Papyrus perspective

* Open the model named Openes\_T12\_T13\_examples contained in the the top left view,in the openes\_examples project.