*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.

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

|  |  |  |
| --- | --- | --- |
| Requirements satisfied by T1.3 | 20.4% | [19](#19),[20](#20),[24](#24),[25](#25),[26](#26),[27](#27),[28](#28),[29](#29),[32](#32),[36](#36),[37](#37),[38](#38),[39](#39),[40](#40),[52](#52),[61](#61),[62](#62),[69](#69),[70](#70),[71](#71),[96](#96),[97](#97) |
| Requirements that should be satisfied by task T1.3 | 2.8% | [34,[35,[95](#_Requirement__95)](#_Requirement__35)](#_Requirement__34) |
| Requirements related to task T1.2 | 24.1% | 1,2,3,4,5,6,7,10,11,18,41,43,44,47,50,54,55,56,58,63,65,67,92,94,105,107 |
| Requirements related to task T1.4 | 8.3% | 8,21,22,23,42,45,46,49,101 |
| Requirements that are out of OSMK scope | 44.4% | 9,12,13,14,15,16,17,30,31,33,48,51,53,57,59,60,64,66,68,72,73,74,75,76,77,78,79,80,81,82,83,84,85,86,87,88,89,90,91,93,98,99,100,102,103,104,106,108 |

## Satisfied requirements

### Requirement 19

Requirement definition :

*"The OSMK shall include the extra-functional performance requirements of all application components on all resources in the platform."*

**Comment :** The performance requirements should (preferably) be specified in the XML format used by our proprietary tools (SDF3 and CompSOC). The actual implementation of the application running on the platform should be provided in the C programming language following additional programming constraints such that the program adheres to the dataflow paradigm that is used in our analysis and mapping tools. The performance requirements that need to be specified are: - WCET of each component in the application - Worst-case memory requirement of each component in the application - Latency and bandwidth requirement for each connection between components in the application - Throughput and latency requirements of the application - Energy budget of the application

Satisfaction traceability :

This requirement is satisfied in CEA Model by :

* CEA\_Examples::Req\_11\_example::A NFPs (Class)

**Rational :** The requirement 19 is satisfied by defining NFPs using MARTE (i.e., UML properties with MARTE stereotype applied on those properties), and then specifying NFP\_Constraints on those NFPs. Constraints are specified using NFP\_Constraints. Examples are given through the "A NFPs" class.

This requirement is also satisfied in TCS Model by :

* PHY\_function\_overview::WF application SOFT (Class)
* PHY\_function\_overview::WF application SYS (Interface)

**Rational :** The requirement 19 is satisfied by TCS case study because Capella is able to add viewpoints based on the MARTE profile and it is extended to UML.

### Requirement 20

Requirement definition :

*"OSMK shall support specification of non-functional requirements for correct operation of a subsystem in a system context. For example, requirements on bandwidth and latency for performing accesses to system memory."*

**Comment :** Facilitate efficient and correct integration of subsystems.

Satisfaction traceability :

This requirement is satisfied in CEA Model by :

* CEA\_Examples::Req\_11\_example::A NFPs (Class)

**Rational :** The requirement 20 is satisfied by defining NFPs using MARTE (i.e., UML properties with MARTE stereotype applied on those properties), and then specifying NFP\_Constraints on those NFPs. The specific context for the evaluation of those constraints (as requested by the requirement), are identified using the "mode" attribute NFP\_Constraint, which refers to a particular operational mode specified as a State in a StateMachine. The constrained element is identified through regular properties of the underlying UML Constraint. Examples are given through the "A NFPs" class.

### Requirement 24

Requirement definition :

*"OSMK may enable to model real-time constraints attached to setting and querying of attributes in "IDLE" mode."*

**Comment :** Not fundamental if one considers that reactivity will always be largely better than a human operator perception. Attribute values changes with time constraint

Satisfaction traceability :

This requirement is satisfied in CEA Model by :

* CEA\_Examples::Req\_11\_example::A NFPs (Class)

**Rational :** The requirement 24 is satisfied by defining NFPs using MARTE (i.e., UML properties with MARTE stereotype applied on those properties), and then specifying NFP\_Constraints on those NFPs. The IDLE mode is identified using the MARTE::Mode stereotype, and apply it on a State of a StateMachine (in the provided example, the state machine belongs to class A, from package Req\_11\_example in the model provided by CEA). The timing constraints are specified using NFP\_Constraints. The property on which the read / write acces time constraint apply are identified through regular properties of the underlying UML Constraint. Examples are given through the "A NFPs" class.

### Requirement 25

Requirement definition :

*"OSMK shall enable to model real-time constraints attached to exhange of information among components in "RUNNING" mode."*

**Comment :** This is a fundamental aspect in order to make consistent system models for real-time solutions.

Satisfaction traceability :

This requirement is satisfied in CEA Model by :

* CEA\_Examples::Req\_11\_example::A NFPs (Class)

**Rational :** The requirement 25 is satisfied by defining NFPs using MARTE (i.e., UML properties with MARTE stereotype applied on those properties), and then specifying NFP\_Constraints on those NFPs. The RUNNING mode is identified using the MARTE::Mode stereotype, and apply it on a State of a StateMachine (in the provided example, the state machine belongs to class A, from package Req\_11\_example in the model provided by CEA). The timing constraints are specified using NFP\_Constraints. The Operation on which the WCET constraint applies is identified through regular properties of the underlying UML Constraint. Examples are given through the "A NFPs" class.

### Requirement 26

Requirement definition :

*"OSMK shall enable to formally attach real-time constraints within the context of real-time critical scenarios expressed in the form of sequence diagrams."*

Satisfaction traceability :

This requirement is satisfied in Common Model by :

* OPENES\_T12\_T13\_Examples::Decoding Scenario (Interaction)

**Rational :** The requirement 26 is satisfied by the Decoding Scenario in the OpenES common model. This scenario is containing timing constraints on operation execution durations.

This requirement is also satisfied in TCS Model by :

* PHY\_function\_overview::WF application SYS (Interface)

**Rational :** The requirement 26 is satisfied by TCS case study because it is defined on the Sequence diagram of Capella with the MARTE Profile and it could be exported to UML

### Requirement 27

Requirement definition :

*"OSMK shall enable to attach Functionnal/Extra-Functionnal properties to components."*

Satisfaction traceability :

This requirement is satisfied in CISC Model by :

* SysMLmodel::Battery::v1 (Property)
* SysMLmodel::Battery::t1 (Property)

**Rational :** The requirement 27 is satisfied by sensor instances in CISC model, with frequency information specified using MARTE HWSensor stereotype.

This requirement is also satisfied in TCS Model by :

* PHY\_function\_overview::WF application SYS (Interface)

**Rational :** The requirement 27 is satisfied by TCS case study because it is defined on the Sequence diagram of Capella with the MARTE Profile and it could be exported to UML

### Requirement 28

Requirement definition :

*"OSMK shall enable to attach real-time constraints to the information exchange process between components "*

Satisfaction traceability :

This requirement is satisfied in CISC Model by :

* SysMLmodel::Battery to BMU (InformationFlow)

**Rational :** The requirement 28 is satisfied by the information flow from Battery to Battery Management Unit to which a timed constraint has been applied.

This requirement is also satisfied in TCS Model by :

* PHY\_function\_overview::WF application SYS (Interface)

**Rational :** The requirement 28 is satisfied by TCS case study because it is defined on the Sequence diagram of Capella with the MARTE Profile and it could be exported to UML

### Requirement 29

Requirement definition :

*"OSMK shall enable to attach real-time constraints to the information exchange process between components "*

Satisfaction traceability :

This requirement is satisfied in CISC Model by :

* SysMLmodel::Battery to BMU (InformationFlow)

**Rational :** The requirement 29 is satisfied by the information flow from Battery to Battery Management Unit to which a timed constraint has been applied.

This requirement is also satisfied in TCS Model by :

* PHY\_function\_overview::WF application SYS (Interface)

**Rational :** The requirement 29 is satisfied by TCS case study because it is defined on the Sequence diagram of Capella with the MARTE Profile and it could be exported to UML

### Requirement 32

Requirement definition :

*"OSMK shall support clear separation between functional model, performance model and power model"*

**Comment :** To reenforce modularity

Satisfaction traceability :

This requirement is satisfied in Common Model by :

* OPENES\_T12\_T13\_Examples::Extra Functional Characterization (Package)

**Rational :** The requirement 32 is satisfied in this example model where all the NFP related information has been put in a separate package, linked to the constrained elements thanks to MARTE Assign mechanism.

### Requirement 36

Requirement definition :

*"System components of OSMK should be parameterizable so that Power-related parameters as throughput and latency should be assigned to them"*

Satisfaction traceability :

This requirement is satisfied in Common Model by :

* OPENES\_T12\_T13\_Examples::Scenario NFP Requirements::Scenario NFP Requirements (Class)

**Rational :** The requirement 36 by defining NFPs using MARTE (i.e., UML properties with MARTE stereotype applied on those properties), and then specifying NFP\_Constraints on those NFPs (just like other NFP related requirements, such as requirement 20). An example of latency constraint (which is a power related requirement) is given with class Scenario NFP Requirement.

### Requirement 37

Requirement definition :

*"The OSMK IP definition shall allow configurable power related parameters"*

**Comment :** e.g Voltage, frequency

Satisfaction traceability :

This requirement is satisfied in CEA Model by :

* Docea CEA Model Examples::PowerInputs (Package)

**Rational :** The requirement 37 is satisfied by the definition of reusable classes (DigitalBlockPowerInput and SMPSPowerInput), specifying a set of power related non-functional properties. These properties have been specified according to inputs required by DOCEA power and thermal analysis tools. Specific values are specified by instantiating these class, and assigning the instances (using the Marte Assign relationship) to model elements which represent Systems, sub-systems or IPs.

### Requirement 38

Requirement definition :

*"The OSMK Sub-System definition shall allow configurable power related parameters"*

**Comment :** e.g picture size

Satisfaction traceability :

This requirement is satisfied in CEA Model by :

* Docea CEA Model Examples::PowerInputs (Package)

**Rational :** The requirement 38 is satisfied by the definition of reusable classes (DigitalBlockPowerInput and SMPSPowerInput), specifying a set of power related non-functional properties. These properties have been specified according to inputs required by DOCEA power and thermal analysis tools. Specific values are specified by instantiating these class, and assigning the instances (using the Marte Assign relationship) to model elements which represent Systems, sub-systems or IPs.

### Requirement 39

Requirement definition :

*"The OSMK System definition shall allow configurable power related parameters "*

**Comment :** e.g battery life time, critical temperature...

Satisfaction traceability :

This requirement is satisfied in CEA Model by :

* Docea CEA Model Examples::PowerInputs (Package)

**Rational :** The requirement 39 is satisfied by the definition of reusable classes (DigitalBlockPowerInput and SMPSPowerInput), specifying a set of power related non-functional properties. These properties have been specified according to inputs required by DOCEA power and thermal analysis tools. Specific values are specified by instantiating these class, and assigning the instances (using the Marte Assign relationship) to model elements which represent Systems, sub-systems or IPs.

### Requirement 40

Requirement definition :

*"OSMK shall support modeling of subsystem performance. These include the subsystem latency, processing requirements, and power consumption for performing a specific use case."*

**Comment :** Facilitate efficient and correct integration of subsystems.

Satisfaction traceability :

This requirement is satisfied in Common Model by :

* OPENES\_T12\_T13\_Examples::Scenario NFP Requirements::Scenario NFP Requirements (Class)

**Rational :** The requirement 40 by defining NFPs using MARTE (i.e., UML properties with MARTE stereotype applied on those properties), and then specifying NFP\_Constraints on those NFPs (just like other NFP related requirements, such as requirement 20). The specific use case on which the constraints apply is identified though a specific scenario, which is modeled using a sequence diagram. If a constraint applies on the overall scenario, then the scenario shall be the constrained element of the underlying UML constraint. If the constraint only concerns a sub-system within the scenario, then the constrained element of the underlying constraint shall be a LifeLine of the context scenario (where this life line represents the specific sub-system). An example is given with class Scenario NFP Requirement.

### Requirement 52

Requirement definition :

*"OSMK shall support expression of relationships between model elements, expressing that a model contains abstracted and/or synthesis information from other models, in regards with a specific usage."*

**Comment :** For instance, a timing property may be expressed in a model as a quantity of time. Another model may contain the same information in a refined form, i.e. the full communication path with detailed components timing information.

Satisfaction traceability :

This requirement is satisfied in Common Model by :

* OPENES\_T12\_T13\_Examples::constraint refinement (Abstraction)

**Rational :** The requirement 52 is satisfied by the "Refine" abstraction in OpenES example model. This "Refine" relationship is linking two timing constraints coming from two different interactions, describing the same scenario at two different refinement levels.

### Requirement 61

Requirement definition :

*"The real-time constraints capture may be compliant with MARTE."*

Satisfaction traceability :

This requirement is satisfied in CISC Model by :

* SysMLmodel::SafeState1 (StateMachine)

**Rational :** The requirement 61 is satisfied by CISC case study since it contains model elements with MARTE real-time constraints. For instance the SafeState state machine is having the TimedProcessing stereotype applied.

This requirement is also satisfied in TCS Model by :

* PHY\_function\_overview::WF application SYS (Interface)

**Rational :** The requirement 61 is satisfied by TCS case study because it is defined on the Sequence diagram of Capella with the MARTE Profile and it could be exported to UML

### Requirement 62

Requirement definition :

*"OSMK shall enable to capture real-time constraints in the structural model."*

Satisfaction traceability :

This requirement is satisfied in CISC Model by :

* SysMLmodel::Battery::v1 (Property)
* SysMLmodel::Battery::t1 (Property)

**Rational :** The requirement 62 is satisfied by the sensor instances in CISC model, with frequency information specified using MARTE HWSensor stereotype.

This requirement is also satisfied in TCS Model by :

* PHY\_function\_overview::WF application SYS (Interface)

**Rational :** The requirement 62 is satisfied by TCS case study because it is defined on the structural diagram of Capella with the MARTE Profile and it could be exported to UML

### Requirement 69

Requirement definition :

*"The OSMK shall support the specification of a power policy. "*

**Comment :** Power management

Satisfaction traceability :

This requirement is satisfied in CEA Model by :

* Docea CEA Model Examples::PowerInputs (Package)

**Rational :** The requirement 69 is partially satisfied by the definition of reusable classes (DigitalBlockPowerInput and SMPSPowerInput), specifying a set of power related non-functional properties. The complete details of the power policy are not directly captured by these classes, but with references to external configuration files (e.g., leak\_ref\_table) which are specific to Docea tools.

### Requirement 70

Requirement definition :

*"The OSMK shall support the specification and implementation of a thermal management policy"*

**Comment :** Thermal management/throttling

Satisfaction traceability :

This requirement is satisfied in OSMKProfile by :

* OSMK::Thermal (Profile)
* Docea CEA Model Examples::ThermalInputs (Package)

**Rational :** The requirement 70 is satisfied by the definition of OSMK::Thermal sub-profile. This profile captures information related to thermal analysis, and has been designed according to information required by Docea thermal analysis tools. A usage example is given in package ThermalInputs of the DOCEA example model.

### Requirement 71

Requirement definition :

*"IP, sub-system, package and board thermal constraints shall be specified as input to OTC"*

Satisfaction traceability :

This requirement is satisfied in OSMKProfile by :

* OSMK::Thermal (Profile)
* Docea CEA Model Examples::ThermalInputs (Package)

**Rational :** The requirement 71 is satisfied by the definition of OSMK::Thermal sub-profile. This profile captures information related to thermal analysis, and has been designed according to information required by Docea thermal analysis tools. A usage example is given in package ThermalInputs of the DOCEA example model.

### Requirement 96

Requirement definition :

*"OSMK methodology shall support the specification of Power related System constraints"*

**Comment :** e.g battery life time

Satisfaction traceability :

This requirement is satisfied in Common Model by :

* OPENES\_T12\_T13\_Examples::Scenario NFP Requirements::Scenario NFP Requirements (Class)

**Rational :** The requirement 96 by defining NFPs using MARTE (i.e., UML properties with MARTE stereotype applied on those properties), and then specifying NFP\_Constraints on those NFPs (just like other NFP related requirements, such as requirement 20). An example of power constraint is given with class Scenario NFP Requirement. In the example, the constraint refers to a specific usage scenario. This is done by having a lifeline as the constrained element of the underlying UML constraint. The same mechanism can be used to constrain more generally a system (or sub-system, or specific IP) by constraining the appropriate model element.

### Requirement 97

Requirement definition :

*"OSMK methodology shall support the specification of Power related IP constraints "*

**Comment :** e.g regulator efficiency

Satisfaction traceability :

This requirement is satisfied in Common Model by :

* OPENES\_T12\_T13\_Examples::Scenario NFP Requirements::Scenario NFP Requirements (Class)

**Rational :** The requirement 97 by defining NFPs using MARTE (i.e., UML properties with MARTE stereotype applied on those properties), and then specifying NFP\_Constraints on those NFPs (just like other NFP related requirements, such as requirement 20). An example of power constraint is given with class Scenario NFP Requirement. In the example, the constraint refers to a specific usage scenario. This is done by having a lifeline as the constrained element of the underlying UML constraint. The same mechanism can be used to constrain more generally a system (or sub-system, or specific IP) by constraining the appropriate model element.

## Not satisfied requirements

### Requirement 34

Requirement definition :

*"OSMK shall specify the voltage intent"*

**Comment :** How the power is distributed from battery or wall plug down to IP => used for power estimation

### Requirement 35

Requirement definition :

*"OSMK shall specify the clock intent "*

**Comment :** How the clock is distributed from quartz or PLL down to Ips => used for power estimation

### Requirement 95

Requirement definition :

*"OSMK methodology shall support UPF or CPF language for power intent description"*

**Comment :** To be standard compliant

# 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

|  |  |  |
| --- | --- | --- |
| Requirements satisfied by T1.4 | 8.3% | [8](#8),[21](#21),[22](#22),[23](#23),[42](#42),[45](#45),[46](#46),[49](#49),[101](#101) |
| Requirements that should be satisfied by task T1.4 | 0.0% |  |
| Requirements related to task T1.2 | 24.1% | 1,2,3,4,5,6,7,10,11,18,41,43,44,47,50,54,55,56,58,63,65,67,92,94,105,107 |
| Requirements related to task T1.3 | 23.1% | 19,20,24,25,26,27,28,29,32,34,35,36,37,38,39,40,52,61,62,69,70,71,95,96,97 |
| Requirements that are out of OSMK scope | 44.4% | 9,12,13,14,15,16,17,30,31,33,48,51,53,57,59,60,64,66,68,72,73,74,75,76,77,78,79,80,81,82,83,84,85,86,87,88,89,90,91,93,98,99,100,102,103,104,106,108 |

## Satisfied requirements

### Requirement 8

Requirement definition :

*"OSMK shall authorize a same component to interact with other components with different connections such as message/flow oriented connections."*

**Comment :** Same idea as what SysML authorize with streams. This is an essential dimension compared to most executable modelling solutions, where all interactions are either message-based (UML, IDL, …) or flow-based (Simulink), depending on the underlying execution mechanism.

Satisfaction traceability :

This requirement is satisfied in TCS Model by :

* PHY\_function\_overview::WF application SYS (Interface)

**Rational :** The requirement 8 is satisified by TCS case study because the interactions in Capella can be specified in flow or message oriented. It could be easily exported to an UML tool.

### Requirement 21

Requirement definition :

*"OSMK shall support modeling of subsystem functions / services and how they can be combined into use cases."*

**Comment :** Capture functional behavior of subsystems.

Satisfaction traceability :

This requirement is satisfied in ST Model by :

* SimplifiedGBSAPI::UseCase::SetRectangleAndRotation (Interaction)

**Rational :** The interaction "SetRectangleAndRotation" is satisfying the requirement 21 since it provide an example of the a subsystem use case, defined a sequence diagram showing a sequence API function calls.

### Requirement 22

Requirement definition :

*"OSMK shall support modeling of subsystems that contain multiple processor cores."*

**Comment :** Do not limit techniques to single core only.

Satisfaction traceability :

This requirement is satisfied in Common Model by :

* OPENES\_T12\_T13\_Examples::CortexA9 Architecture::CA9 (Class)

**Rational :** The requirement 22 is satisfied by the OpenES common model which contains a hierarchical description of ARM Cortex A9 quadri core processor.

### Requirement 23

Requirement definition :

*"OSMK shall support specification of host environment for hosted subsystems (supported host processors, operating systems, etc.)"*

**Comment :** Facilitate efficient and correct integration of subsystems.

Satisfaction traceability :

This requirement is satisfied in ST Model by :

* GBSInteractionModel::Signature::Classes::GBS (Class)

**Rational :** Requirement 23 is satisfied by the GBS model thanks to OSMK annotation profile. We used this generic annotation mechanismb to associate to the GBS class required information such as hosting operating system and processor.

### Requirement 42

Requirement definition :

*"The OSMK shall allow to associate hardware specific information to datatypes, such as the number of bits required to store a given variable. "*

**Comment :** In low level software, bit optimization can require to now very early in the flow the number of bits required to encode a given variable (can be any value, from 1 to at least 64..)

Satisfaction traceability :

This requirement is satisfied in ST Model by :

* SimplifiedGBSAPI::Structs::stmlld\_gbs\_RectangleParam\_t::OriginX (Property)

**Rational :** The SimplifiedGBS API model is satisfying requirement 42 thanks to ST API profile. This profile allows to annotate typed elements like struct member (for instance Rectangle\_t::OriginX) with the Dynamic tagged value which specifies maximum number of bits needed to encode the values held by this variable.

### Requirement 45

Requirement definition :

*"The OSMK shall allow to define system level synchronization mechanisms, including SW/SW, SW/HW and HW/HW communications"*

Satisfaction traceability :

This requirement is satisfied in ST Model by :

* synchro\_dma\_cpu (Model)

**Rational :** The synchro\_dma\_cpu model is partially satisfying the requirement 45. It contains an fUML executable description showing synchronizations between a CPU and DMA, including HW/HW synchronizations.

This requirement is also satisfied in ST Model by :

* GBSInteractionModel::Interactions (Package)

**Rational :** The interactions in ST GBS model are showing interactions between the the subsystem firmware and internal hardware blocks. This description of SW/HW synchronizations is partially satisfying the requirement 45.

### Requirement 46

Requirement definition :

*"The OSMK shall allow to model Hardware/Software APIs. OSMK shall enable to select interfaces between applicative part and platform parts of the system through a set of reference APIs"*

**Comment :** Thess APIs define the set of functionnalities offered by an HW block and useable for SW or System Level scenarios (also known as Low Level Driver or Hardware Abstraction Layer). For example, API used for creating and activating tasks on subsystem and control their operation.

Satisfaction traceability :

This requirement is satisfied in ST Model by :

* SimplifiedGBSAPI::SimplifiedGBSAPI (Class)

**Rational :** The requirement 46 is satisfied by the SimplifiedGBSAPI UML model. This model provides a simplified description of Multimedia IP Low Level Driver. This programming API, which has to be used by OS specific drivers is providing a SW level. description of the services offered by the IP.

### Requirement 49

Requirement definition :

*"OSMK shall enable to reference APIs for SDR use case. It shall be compliant with a SDR Architecture such as JTRS Devices API or ESSOR Architecture RD/RS/RSS APIs."*

**Comment :** One essential aspect in SDR is that the standard SDR specifications are setting the boundary between the radio application (="waveform application") and the SDR platform.

Satisfaction traceability :

This requirement is satisfied in TCS Model by :

* PHY\_function\_overview::WF application SYS (Interface)

**Rational :** The requirement 49 is satisfied by TCS case study because it is defined in the capella tool and is in progress to be exported to UML

### Requirement 101

Requirement definition :

*"For inter-subsystem SW-to-SW communication a SW API is adopted which abstracts from low-level implementation aspects such as system address maps, interrupt configurations and bus protocols."*

**Comment :** Potential examples are the MCA MCapi protocols or the POSIX mmap or sockets interfaces.

Satisfaction traceability :

This requirement is satisfied in ST Model by :

* SimplifiedGBSAPI::SimplifiedGBSAPI (Class)

**Rational :** Requirement 101 is satisfied by the SimplifiedGBSAPI which provides an abstraction of low level implementation aspects.

## Not satisfied requirements

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)

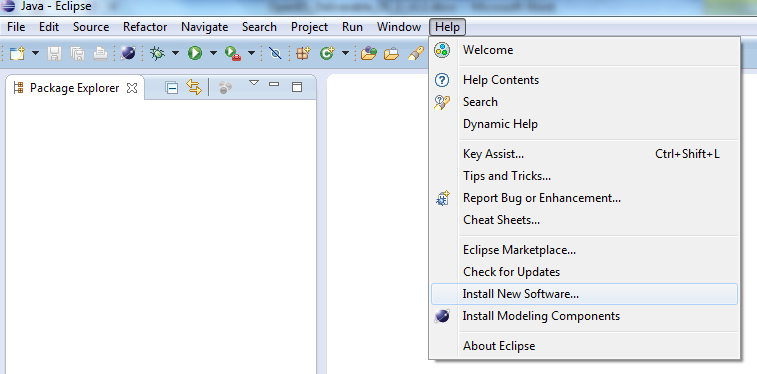


Figure 1 : Install new Software menu

* Step2 : Select Kepler main repository

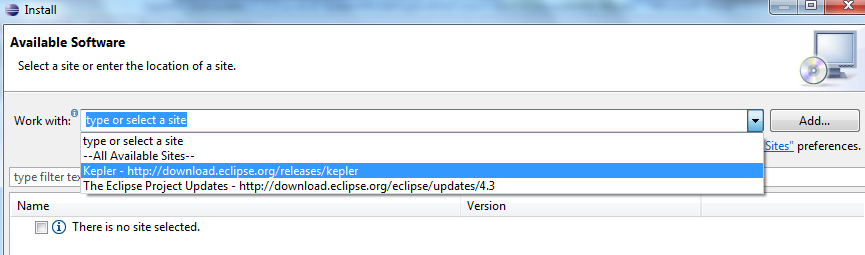


Figure 2 Select Luna Repository

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

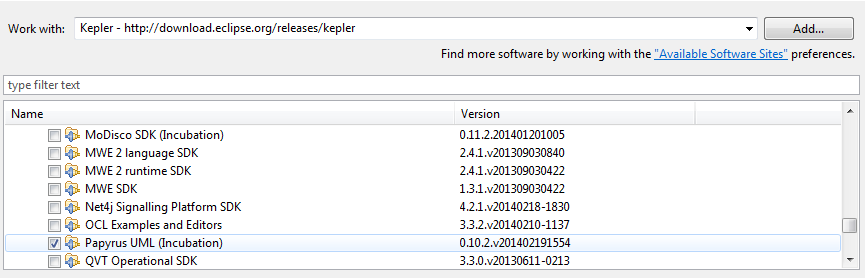


Figure 3 Select Papyrus in Modelling subproject

* 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.

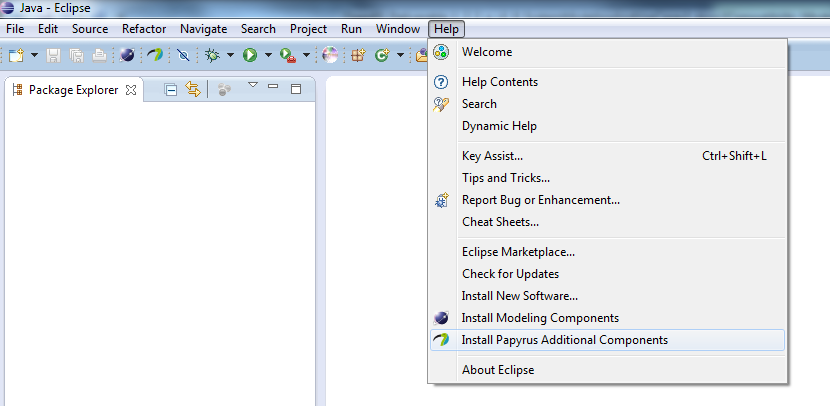


Figure 4 Install Papyrus Additional Components

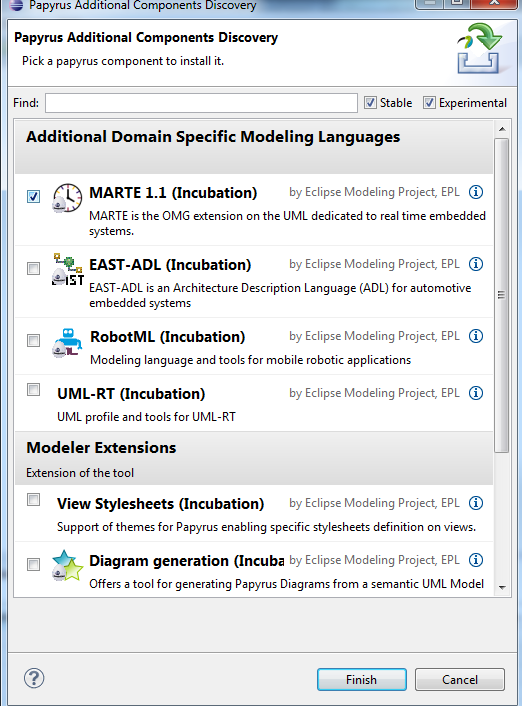


Figure 5 Select MARTE and click on finish

## 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

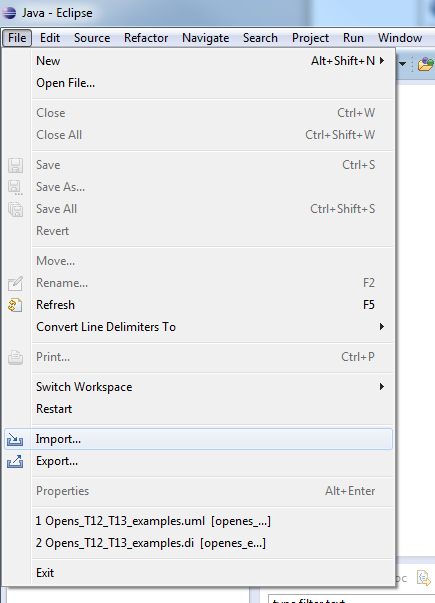


Figure 6 Click on File -> Import

* Select General/Existing Project into workspace

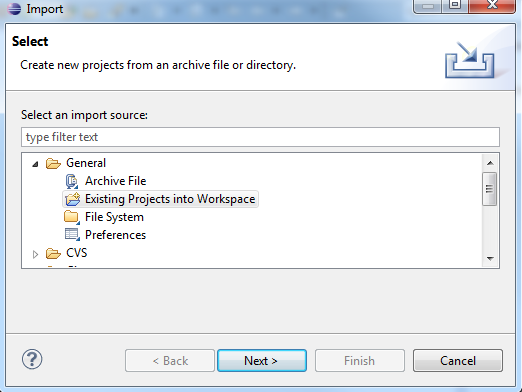


Figure 7 Select General/Existing projects into Workspace

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

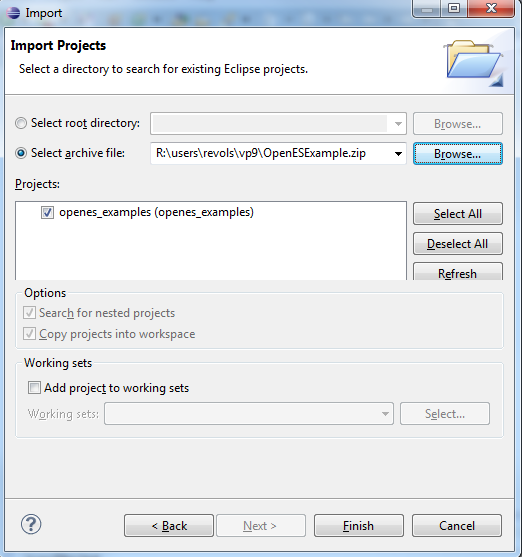


Figure 8 Zipped project selection

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

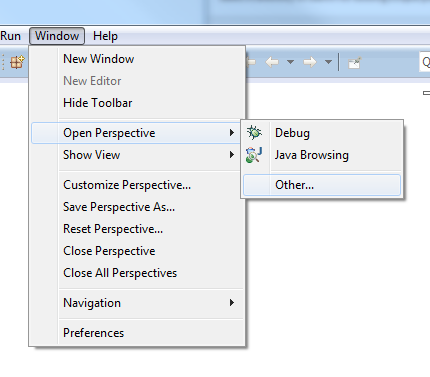


Figure 9 Path to activate Papyrus perspective

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