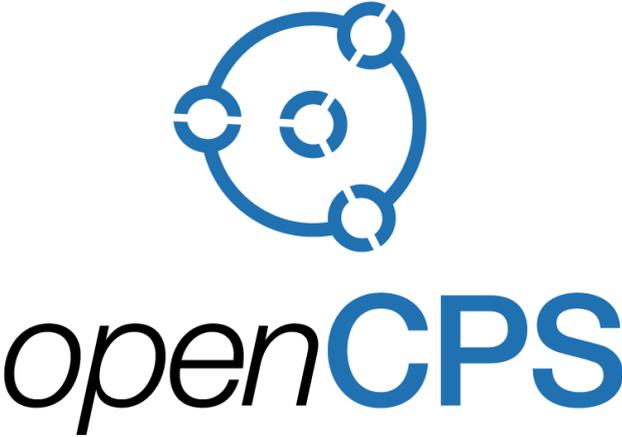


<b>D2.3</b>	<b>FMI Master Simulation Tool</b>
Access <sup>1</sup> :	PU
Type <sup>2</sup> :	Prototype
Version:	2.0
Due Dates <sup>3</sup> :	M24, M36
 <p><i>Open Cyber-Physical System Model-Driven Certified Development</i></p>	
<b>Executive summary<sup>4</sup>:</b>	
<p>This document comes with the software release of the D2.3 FMI Master Simulation Tool OMSimulator – a main collaborative development effort of OpenCPS.</p> <p>Its goal is to give an overview of the work done for the development of this simulation tool, as well as additional extensions or integrations in third party tools such as OMEdit or Papyrus.</p> <p>In summary, we can mention that most of the expectations, driven by industry needs, have been met. OMSimulator has been tested and verified using several complex industrial demonstrators. It provides now an industry exploitable open source solution for complex system co-simulation, and is a solid base for further developments.</p>	

<sup>1</sup> Access classification as per definitions in PCA; PU = Public, CO = Confidential. Access classification per deliverable stated in FPP.

<sup>2</sup> Deliverable type according to FPP, note that all non-report deliverables must be accompanied by a deliverable report.

<sup>3</sup> Due month(s) according to FPP.

<sup>4</sup> It is mandatory to provide an executive summary for each deliverable.

**Deliverable Contributors:**

	Name	Organisation	Primary role in project	Main Author(s) <sup>5</sup>
Deliverable Leader <sup>6</sup>	Sebastien Revol	CEA	T2.3 leader	X
Contributing Author(s) <sup>7</sup>	Lennart Ochel	SICSEast	T2.3 Member	X
	Robert Braun	SKF	T2.3 Member	X
	Adeel Asghar	LiU	T2.3 Member	X
	Robert Hällqvist	Saab	T2.3 Member	X
	Akos Horvath	IQL	T2.3 Member	
	Rebeka Farkas	IQL	T2.3 Member	
	Zoltán Gera	ELTE-Soft	T2.3 Member	
Internal Reviewer(s) <sup>8</sup>	Eek Magnus	Saab	OpenCPS Project leader	
	Peter Fritzson	LiU	Scientific Coordinator	

**Document History:**

Version	Date	Reason for Change	Status <sup>9</sup>
1.0	29/11/2018	First Draft Version	Draft
2.0	30/11/2018	Final Version	Final

<sup>5</sup> Indicate Main Author(s) with an “X” in this column.

<sup>6</sup> Deliverable leader according to FPP, role definition in PCA.

<sup>7</sup> Person(s) from contributing partners for the deliverable, expected contributing partners stated in FPP.

<sup>8</sup> Typically person(s) with appropriate expertise to assess deliverable structure and quality.

<sup>9</sup> Status = “Draft”, “In Review”, “Released”.

## CONTENTS

ABBREVIATIONS.....	3
1 INTRODUCTION .....	4
2 OPENCPS MASTER SIMULATION TOOL SOFTWARE.....	4
3 OMSIMULATOR RELATED ACTIVITIES.....	5
3.1 Requirement Validation .....	5
3.2 Master Simulation Tool tests .....	5
3.3 Simulation performance improvements .....	5
3.4 Papyrus integration .....	6
4 CONCLUSION.....	6
5 REFERENCES .....	7

## ABBREVIATIONS

List of abbreviations/acronyms used in document:

<b>Abbreviation</b>	<b>Definition</b>
FMI	Functional Mock-up Interface
FMU	Functional Mock-up Unit
M&S	Modelling and Simulation
N/A	Not Applicable
SotA	State of the Art
TBD	To Be Defined

## 1 INTRODUCTION

OpenCPS Full Project Proposal (FPP) document is defining D2.3 as the software prototype deliverable of the OpenCPS Master Simulation Tool (MST). This tool plays a central role in the project and many work has been done around it, in terms of user interface integration, tests, performance optimization etc.

In this document, we provide pointers to either external website or annex documents allowing to assess and to access to those developments.

## 2 OPENCPS MASTER SIMULATION TOOL SOFTWARE

The OpenCPS MST software is called OMSimulator, and is released as part of the Open Modelica tool suite. Being an Open Source Software, its source code, documentation, as well as pre-built binaries and integration tests can accessed in a public github repository located at the following address:

<https://github.com/OpenModelica/OMSimulator>

In addition to this repository, the annex document called [D2.3-Annex1-OMSimulator-Presentation.pdf](#) provides a very complete overview of the functionalities provided by the tool, as well as its integration in OMEdit Graphical User Interface. We can consider this document as the main documentation part of this D2.3 deliverable. The remaining paragraphs below provide complementary information on additional work related to T2.3 but not addressed by this annex.

Notice that this annex has been submitted to the Modelica Conference 2019<sup>10</sup>, **and is therefore temporarily considered as confidential** until the paper is accepted and presented.

As a summary, it has been shown that OMSimulator is now an **industry-grade** and **standard-based** master simulation tool, combining the benefits of **FMI** [1], **Transmission Line Modeling (TLM)** [2] and **SSP** [3]. The following tool characteristics can be noted:

- It is a highly flexible tool supporting various combinations of composite model setups, including **mixing** FMUs for co-simulation (**FMU-CS**) and clusters of FMUs for model exchange (**FMU-ME**) with several integration methods.
- It supports TLM-based **distributed simulations**, and can be directly **coupled to external simulation tools**.
- It provides **C, Lua and Python programming interfaces**, allowing **external tool integration** such as Graphical User Interfaces like **OMEdit** and **Papyrus**.
- Last, its **early native SSP standard support** allows **tool interoperability** thanks to a standardized import and export of composite models, including FMU instantiation, parameterization, connexion and graphical layout information.

---

<sup>10</sup> <https://modelica.org/events/modelica2019>

### 3 OMSIMULATOR RELATED ACTIVITIES

In the following sections, we provide additional information regarding complementary works done in the context of Task 2.3 and pointing to other deliverables or other annexes of this deliverable.

#### 3.1 Requirement Validation

The objective of the task T2.1 was to – based on real industry needs – produce a set of requirements for the development of the MST. Those requirements have been aggregated in the publicly available D2.1 deliverable, released at M12. This deliverable has been then carefully analysed, and taken into account during the development steps.

At the end of the project, the annex excel sheet called [D2.3-Annex2-OMSimulator-Requirement-Validation-M36.xlsx](#) proposes a new version of this requirement table with an additional “Compliant” column. This column can take the following values:

- *Yes*: the requirement is fully addressed by OMSimulator or a complementary development realized in OpenCPS project (such as OMEdit/Papyrus) GUI integration.
- *Partial*: the requirement is partially addressed, but remaining developments may be needed.
- *No*: the requirement is not yet addressed.
- *N/A*: either the requirement is not directly concerning OMSimulator or directly connected tools, either it could not be verified.

This coverage analysis can be summarized as below:

Number of requirements	Number of “Yes”	Number of “Partial”	Number of “No”	Number of “N/A”
<b>96</b>	<b>64</b>	<b>16</b>	<b>13</b>	<b>3</b>

We can see that a big majority of requirements is fully or partially addressed. More particularly, all priority 1 requirements are met. The remaining ones, generally considered as lower priority, are still valid and will later be taken into account.

#### 3.2 Master Simulation Tool tests

OMSimulator github repository contains continuous integration tests allowing to ensure industry readiness of the produced software. Moreover, the demonstrators defined in Work Package 6 also highly contributed to test and provide feedbacks to OMSimulator developers. In addition to that, some partners contributed to dedicated validation experiments. More particularly, ELTE Soft summarize in the document [D2.3-Annex3-ELTE-Soft-OMSimulator-Test-Report.pdf](#) a report on tests developed to help to improve the tool maturity.

#### 3.3 Simulation performance improvements

Several works have been realized in the project in order to improve the simulation performances. Task 5.6 was for instance defined in order to investigate on speed improvement thanks to parallelization. This work led to the implementation of several alternative master algorithms, described in deliverable D5.6 called “*Implementation of multi-level*”

*scheduler for executing FMUs in parallel with suitable data transfer”.*

In addition to that, IncQuery Labs worked on an alternative master algorithm focusing on a better simulation step size control, relying on meta information associated to each FMU. The result of this work is summarized in the annex [D2.3-Annex4-OMSimulator-Step-Size-Control.pdf](#). This paper has been submitted to the Modelica Conference 2019, **and is therefore temporarily considered as confidential** until it is accepted and presented.

Last we can also mention that additional variable step size algorithms experiments are also currently realized in a Saab/LiU master thesis work.

### 3.4 Papyrus integration

One objective of the project was to improve the connection between the high level modeling languages such as UML and SysML, and system simulation domain as addressed Modelica, FMI standard and OMSimulator.

One way to achieve this objective was to integrate OMSimulator into Papyrus UML tool. This integration has been realized following two complementary axes:

- The “*System Structure and Parameterization*” (SSP) soon-to-be standard import/export support into Papyrus, allowing to use SysML Internal Block Diagram as a graphical editor for FMU instantiation and interconnection. This work is summarized in annex [D2.3-Annex5-SSP-Support-for-Papyrus.pdf](#)
- The integration of scripting capabilities into Papyrus, allowing end-users to run python scripts interacting with OMSimulator directly from Papyrus, as well as easily creating interactive dashboard thanks to Jupyter Notebook<sup>11</sup> integration into Papyrus. This work is summarized in annex [D2.3-Annex6-Scripting-Support-for-Papyrus.pdf](#)

## 4 CONCLUSION

The high number of requirements defined in deliverable D2.1 was a very representative figure of the high expectations the different partners had on the OpenCPS Master Simulation Tool. At the end of the project, we can consider that most of those expectations have been met, so that OMSimulator provides now a strong base on which industrial companies can build their system simulation activities.

Being part of a complete environment supported and used by an active community, we are confident that this work will continue to be improved and will become a reference in the domain of the co-simulation.

---

<sup>11</sup> <http://jupyter.org/>

## 5 REFERENCES

- [1] FMI development group. Functional Mock-up Interface for Model Exchange and Co-Simulation v2.0. Modelica Association Project “FMI”, October 2014. URL <https://www.fmi-standard.org/>. Standard Specification.
- [2] Petter Krus. Robust modelling using bi-lateral delay lines for high speed simulation of complex systems. In DINAME 2011: 14th International Symposium on Dynamic Problems in Mechanics, 2011. URL <http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-67897>. Invited conference contribution.
- [3] <https://www.modelica.org/legal/projects/MA-Project-Application-SSP-2014-06-13.pdf>