# D4.4	# Test-based verification
Access[1]:	**PU**
Type[2]:	**Prototype**
Version:	**1.2**
Due Dates[3]:	**M24, M36**

Open Cyber-Physical System Model-Driven Certified Development

Executive summary[4]:

This deliverable provides a framework for testing for example the Master Simulation tool developed in WP2.

1 Access classification as per definitions in PCA; PU = Public, CO = Confidential. Access classification per deliverable stated in FPP.
2 Deliverable type according to FPP, note that all non-report deliverables must be accompanied by a deliverable report.
3 Due month(s) according to FPP.
4 It is mandatory to provide an executive summary for each deliverable.

Deliverable Contributors:

	Name	Organisation	Primary role in project	Main Author(s)[5]
Deliverable Leader[6]	Martin Sjölund	Linköping University	WP4 leader	X
Contributing Author(s)[7]	Lena Buffoni	Linköping University	WP4 member	
Internal Reviewer(s)[8]	Adrian Pop	RISE SICS East	WP5 leader	

Document History:

Version	Date	Reason for change	Status[9]
1.2	2018-11-30	M36 Release (no change)	Released
1.1	2018-11-21	M36 Draft	Draft
1.0	2017-11-21	First Issue	Released
0.1	2017-11-20	First Draft Version	Draft

[5] Indicate Main Author(s) with an “X” in this column.
[6] Deliverable leader according to FPP, role definition in PCA.
[7] Person(s) from contributing partners for the deliverable, expected contributing partners stated in FPP.
[8] Typically person(s) with appropriate expertise to assess deliverable structure and quality.
[9] Status = “Draft”, “In Review”, “Released”.

Contents

Abbreviations

List of abbreviations/acronyms used in document:

Abbreviation	**Definition**
FMI	Functional Mock-up Interface
FMU	Functional Mock-up Unit
MSL	Modelica Standard Library

1 Introduction

This report accompanies the deliverable D4.4. In order to industrially exploit the results of OPENCPS, it is important to ensure that the master simulator tool provides correct simulation results. This deliverable uses the results of tasks T2.3, T3.6 and T4.3. The main goal is to provide a framework for test-based verification for the OMFMISimulator developed in task T2.3. This framework can then be used to:

- Compare the results of the FMUs simulated by OMFMISimulator to reference values, which can be for example obtained by simulating the corresponding Modelica models with the OpenModelica compiler. This will ensure that the OMSimulator implementation is correct.
- Simulate the models together with their requirement sets as defined in task T4.3 and then use custom methods for test result comparison, such as comparison of requirement specific variables to ensure that the model behaves as specified by the requirements.
- Compare the simulation results of code generated by the code generator developed in task T3.6 to the simulation results form the standard OpenModelica compiler, to ensure that the generated code behaves correctly.

2 Implementation

The framework is highly configurable, allowing the user to setup a wide range of different tool and library versions (with different settings for each library). It is specific to OpenModelica and uses OMPython to communicate with the compiler. Each model is simulated in a separate process and the master tool will kill any job that takes too long.

After a run is completed, the results are stored in an sqlite3 database and a report generator is run on this database to produce tables and graphs. If there are any changes in results or significant performance changes compared to the previously recorded run, an email is sent to the authors of affected changes to OpenModelica.

3 Continuous Testing

The Python-based framework runs batch simulations and compares results against reference values as set by the user. It can run OpenModelica Linux versions at least as old as v1.8.1 (April 2012) and is used to nightly update results of different OpenModelica configurations:

- v1.8.1...master using default settings on most open-source libraries.
- v1.9.7...master using FMI for Model Exchange export, and simulated using OMSimulator (as well as FMI for co-simulation for select libraries).
- The latest master using the new OpenModelica frontend (which will become the default once it handles almost all models the current frontend handles).
- The latest master using the DAE mode of OpenModelica (testing alternative solvers).

4 Results

The setup uses a moving target for the testing, where each OpenModelica maintenance is tracked (which means that the master branch is constantly moving and older releases may get minor updates). Each tested Modelica library is also a moving target, where for example the MSL follows its maintenance branch. For the FMI testing, the master branch follows the latest OMSimulator release as well. The reference files are updated less frequently; for the MSL, the reference files now follow the latest official release of reference files whereas previously we had our own results. There may also be changed in the testing framework, changing default flags or timeouts.

All of this makes interpretation of the results difficult and needs some knowledge of what happened with all these different tools.

4.1 Testing OpenModelica Default Settings

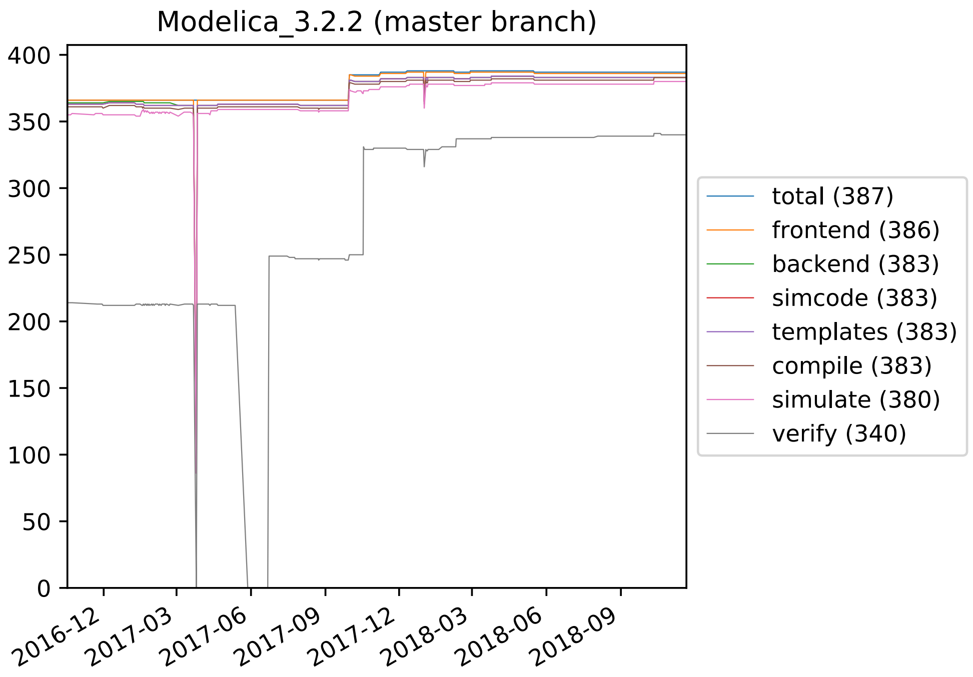

Figure 1: Baseline testing of the MSL.

Figure 1 shows the default OpenModelica settings using the well-tested dassl solver. Note that is has been tested for much longer than the following suites (using FMI and OMSimulator). The predecessor of this framework has been used since 2012 but was very hard to adapt to new settings, or comparing different versions of OpenModelica against each other.

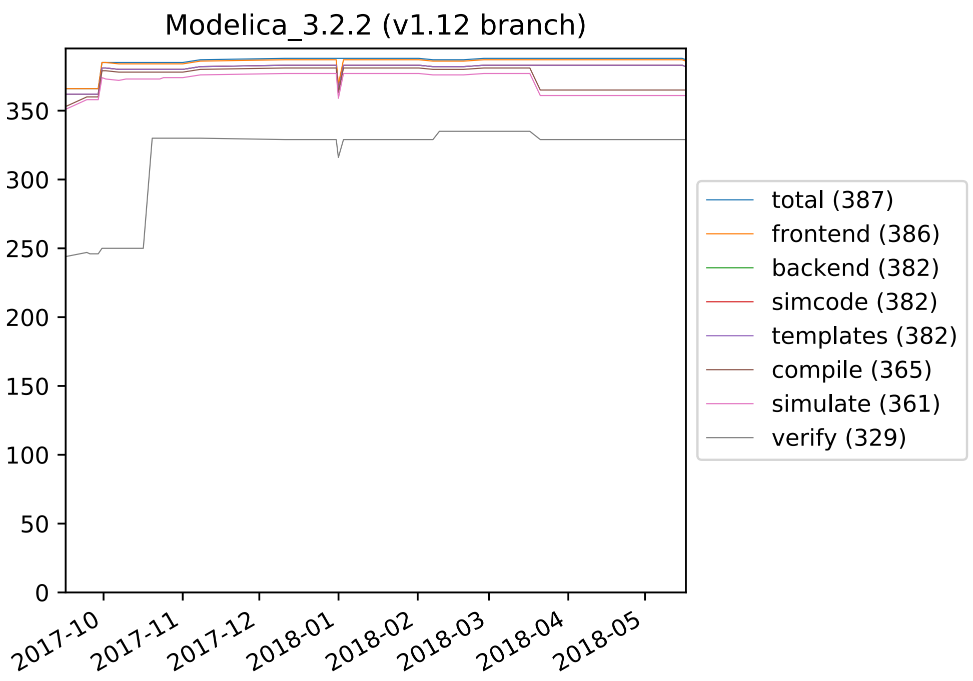

Figure 2: v1.12.0 testing of the MSL.

Figure 2 shows the default OpenModelica settings using the well-tested dassl solver, but running on the v1.12 maintenance branch of OpenModelica using the latest versions of libraries. The figure shows that at 2018-03-20, there is a regression of 16 tests. The email sent to the developers showed that there was no change in OpenModelica at this time, only a change in the MSL (3.2.2-228-gefd1137-om3…3.2.2-230-gb20d683-om3)) which indicated a change to get correct results on the OpenModelica master branch while not being compatible with v.1.12.0.

4.2 Testing FMI Export and OMSimulator

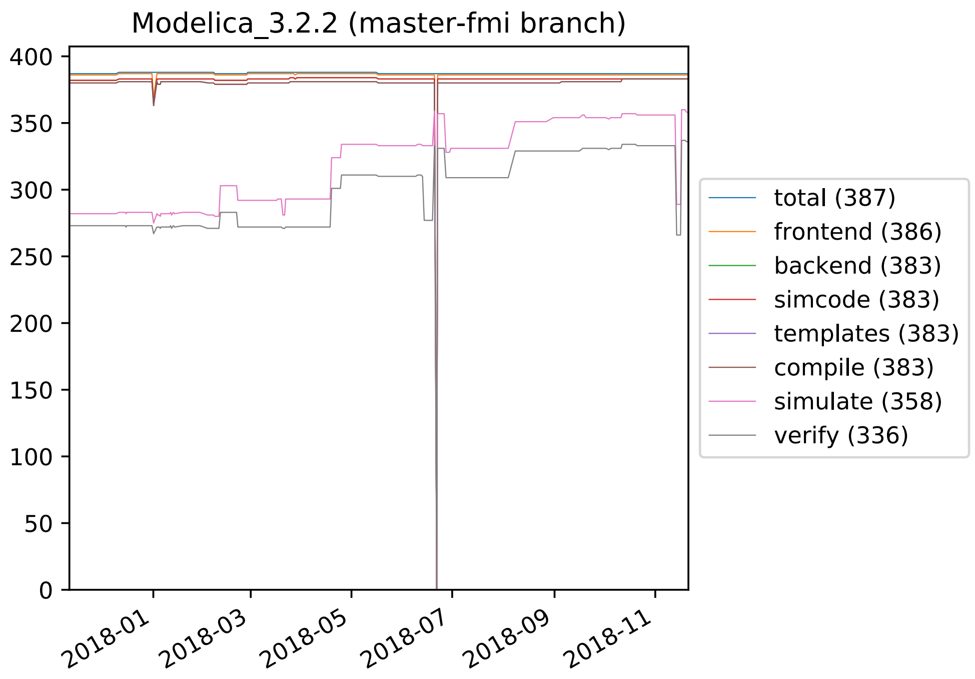

Figure 3: Testing FMI for Modelica Exchange.

In Figure 3, the recent fluctuation coincides with the change of OMSimulator changing from oms2 to oms3 being in the command-line interface. At the same time cvode changed to the default solver instead of forward Euler (the dip was due to cvode crashing if the model contained no continuous states). The results are still not on par with the default OpenModelica settings when it comes to simulations finishing, but 336 of the models verify compared to 340 using the default settings which is remarkably good especially compared to M24 (it is better than the latest OpenModelica release, v1.12.0). It is worth noting that many of the FMUs failed to simulate in older versions of OMSimulator due to race conditions; most of the improvements are due to OMSimulator and not OpenModelica generating better code.

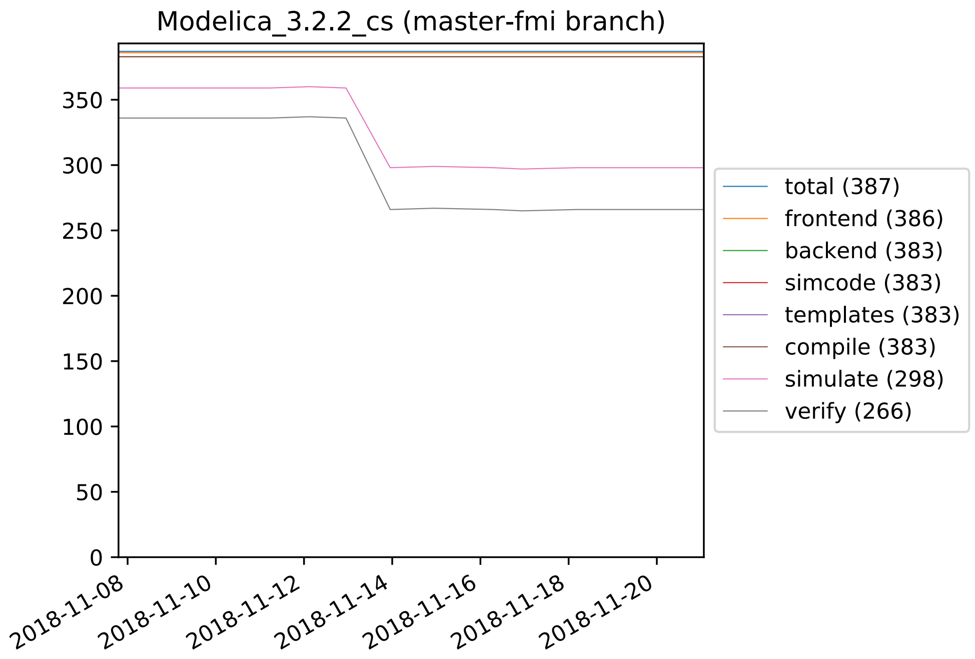

Figure 4: Recent addition to the library testing: FMI for co-simulation.

OpenModelica only exports the forward Euler solver for co-simulation FMUs, making it unable to handle all models. The dip in Figure 4 results coincides with the change of OMSimulator changing from oms2 to oms3 being used internally in the command-line interface.

5 Future Work

In the future, we would like to move the framework into the cloud-based cluster using for example Kubernetes and possibly changing the sqlite3 database to PostgreSQL, MariaDB or similar.

Using a cluster has the advantage that the testing can scale much better (current runs take several hours), but the disadvantage that performance (run-time) of compilation and simulation can no longer be compared since the different nodes in the cluster may have different CPUs.

6 Availability of the Prototype

The framework is available at:

https://github.com/OpenModelica/OpenModelicaLibraryTesting

OpenModelica releases can be downloaded through https://openmodelica.org.