Setup for MSL Regression Testing
Proposal

Revisions

2014-01-13	Initial version (by Leo Gall, BAUSCH-GALL GmbH)

Contents

1 Intent of this document

This document describes a setup for regression tests of the Modelica Standard Library (MSL). The goal is to compare simulation results from different Modelica tools against reference results.

MSL testing has been discussed before[1] and restarted at the 80th and 81st Modelica Design Meeting. Two recent developments facilitated the development of a regression test setup:

- CSV-comparison tool[2], developed by ITI, financed by Modelica Association and
- FMI Cross Check[3], developed by MAP-FMI.

2 Test setup

2.1 Process

A rough overview of the planned process, as discussed at the 81st Modelica Design Meeting:

- Modelica Association (MAP-LIB) provides reference results for
  - all examples of the Modelica Library and
  - all test cases of the ModelicaTest Library.
- Participating tool vendors
  - perform test runs in their specific tool,
  - use the public reference results in order to solve tool issues and
  - upload result files in order to support the MSL development.
- Modelica Association (MAP-LIB)
  - runs the CSV-comparison tool on submitted result files,
  - generates an overview table, similar to FMI Cross Check[4] (showing all participating tools and test cases) and
  - arbitrates between tool vendors and library developers.

To begin with, this process is intended for beta tests and release tests of MSL. If the process works nicely, it could be used for testing on a regular basis (weekly, nightly). Then, a continuous integration tool, like Hudson or Jenkins might be useful.

[1] For instance: email "Proposal for MSL additions and conformance testing" to Modelica-design list from Hubertus Tummescheit, 2012-05-01

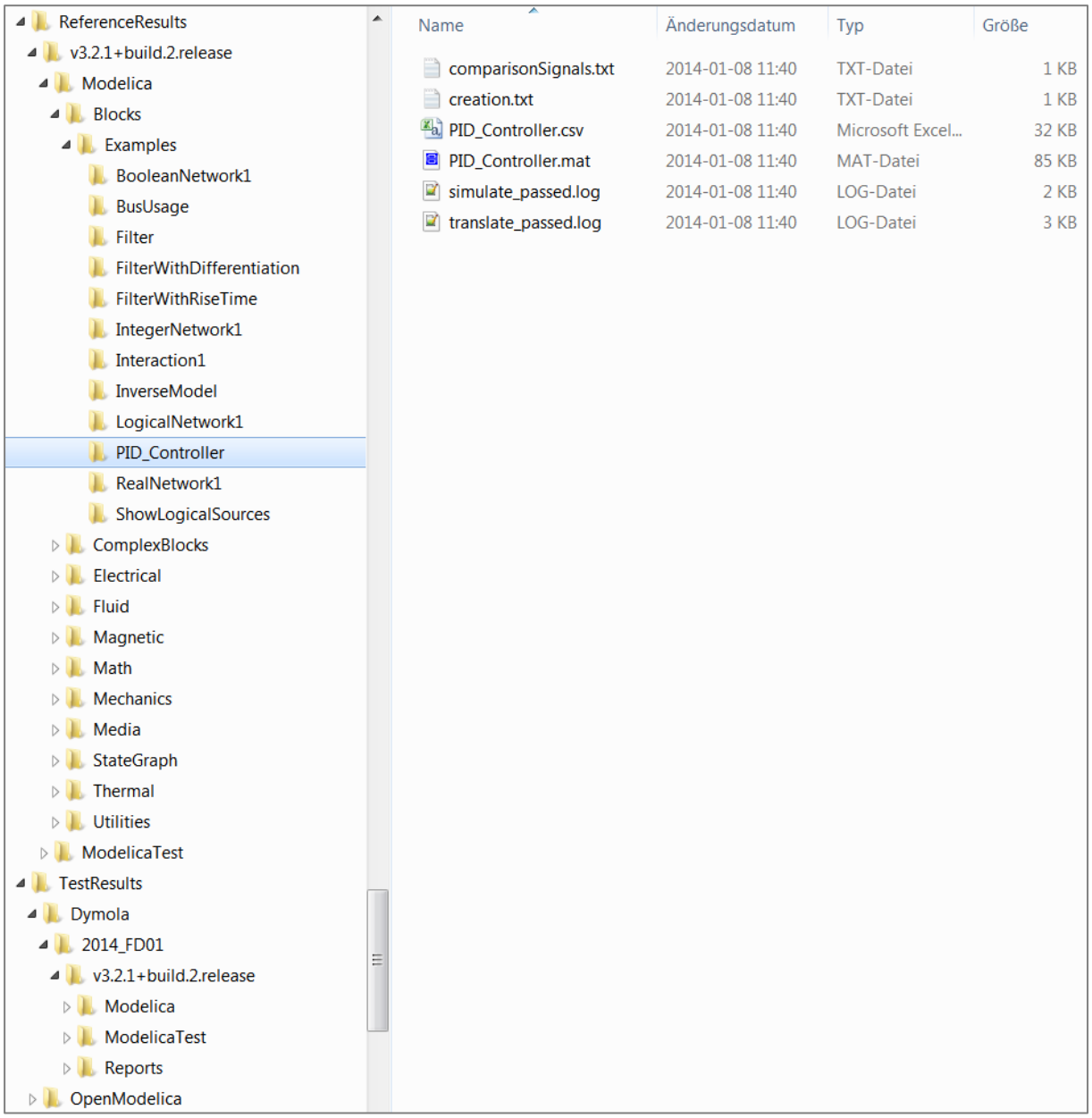
[2] See https://svn.modelica.org/projects/Modelica/branches/tools/csv-compare

[3] See FMI Cross Check Rules, https://svn.fmi-standard.org/fmi/branches/public/CrossCheck_Results/FMI_Cross_Check_Rules_v2_2013_06_26.pdf

[4] Example of a Cross Check table: https://fmi-standard.org/results_FMI_1.0_ModelExchange_win32

2.2 Folder structure

The following screenshot shows a draft folder structure for reference results and test results. The content of these folders is explained in the next section of this document.



On top level, two folders should be stored on MA web:

Folder Name	Description
`ReferenceResults`	Folder containing reference result files ("base"), to be provided by MAP-LIB (probably Library Officers). As an initial step, Leo Gall is going to generate reference result files for MSL 3.2.1 using Dymola (see section3.1). For the future, MAP-LIB has to decide what information is required and who is going to maintain the reference results.
`TestResults`	Contains new result files, to be provided by Tool Vendors. Additionally, it contains comparison flags and comparison reports, generated by CSV-compare-tool on MA server. As an initial step, Leo Gall is going to perform a comparison on his local PC (see section 4).

Inside these folders, we have several options for structuring files. For example:

1. Flat: Modelica.Blocks.Examples.PID_Controller.csv
2. Semi-Flat: Modelica.Blocks.Examples.PID_Controller/PID_Controller.csv
3. Hierarchical: Modelica/Blocks/Examples/PID_Controller/PID_Controller.csv

Option 3 (hierarchical) seems to be most appropriate:

- The Folder structure looks like a Modelica package. We are used to walk through this.
- It's easy and safe to commit specific folders (if a library officer has to update specific examples via SVN)
- If test cases are moved in the Modelica package, it's easy to move the reference results, accordingly.

Notes on this folder structure:

- The reference results and test results should have the same structure in order to allow tree compare using Compare.exe.
- The current version of Compare.exe (csv-compare-1.1.0.7078-win32) assumes unique file names. This would lead to option 1 (flat). But according to Sven Rütz (ITI) the problems with duplicated file names are going to be solved.
- New reference results can be stored by copying a full set or a subset of test results

2.3 Files to be stored for each Test Case

What needs to be stored by a library officer in order to generate new reference results? What needs to be provided by a tool vendor in order to submit test results? The following table gives an overview of a set of files per Test Case.

File name	Description	Do we need this file?
`ModelName.csv`	Result file, containing interesting signals for comparison (e.g. states)	Required
`ModelName.mat`	Result file in tool specific format, including more variables than CSV-file (useful for debugging)	Recommended
`creation.txt`	Documentation of test setup, in order to be able to reproduce the simulation results (possible file contents: operating system, compiler version, used solver, solver settings, special tool settings)	Required
`comparisonSignals.txt`	List of signal names to be compared (these signals are to be included in the CSV-file).	Optional
`translate_passed.log`	Translation log in tool specific text format (with passed/failed as a file name flag)	Recommended
`simulate_passed.log`	Simulation log in tool specific text format (with passed/failed as a file name flag)	Required
`compare_passed.log`	Comparison log, generated (with passed/failed as a file name flag)	Required for TestResults

Currently, CSV is the chosen result format, because the CSV-comparison tool is available. We re-use the CSV-format rules of FMI Cross-Check[5]. The CSV files and (optional) MAT files could be replaced by MTSF (HDF5) files, later.

Notes on log files:

- The file names of the log files are used for generating the "traffic lights" of the overview table (passed/failed/na/error), see section 5.
- Translation and simulation logs: the file content is tool specific and is only used for understanding comparison issues.
- Comparison log: the file content is planned to indicate the worst signal for the overview table. The content of this file needs to be defined together with ITI.
- If we use a more sophisticated test setup in the future, the log files could contain XML tags for JUnit instead of plain text.

[5] FMI Cross Check Rules, https://svn.fmi-standard.org/fmi/branches/public/CrossCheck_Results/FMI_Cross_Check_Rules_v2_2013_06_26.pdf, Appendix B

Meaning of log-file names:

File name	Meaning
`translate_passed.log`	Model has been successfully translated/compiled
`translate_failed.log`	Model could not be translated/compiled.
`translate_na.log`	The tool doesn't give translation diagnostics (only simulation diagnostics)
`simulate_passed.log`	Model has been successfully simulated (reached stop time).
`simulate_failed.log`	Model failed during initialization or simulation.
-	If translation failed, the simulate flag doesn't exist
`compare_passed.log`	All comparison signals are valid (Note: compare.exe accepts differing stop times. So, even for failed simulations, comparison would work and could give a positive flag)
`compare_failed.log`	At least one comparison signal was outside range of validity
`compare_error.log`	Comparison was not possible (e.g. unable to parse one of the two result files)
-	If a test result (CSV) is missing, the compare flag won't be generated

3 Generating and handling results

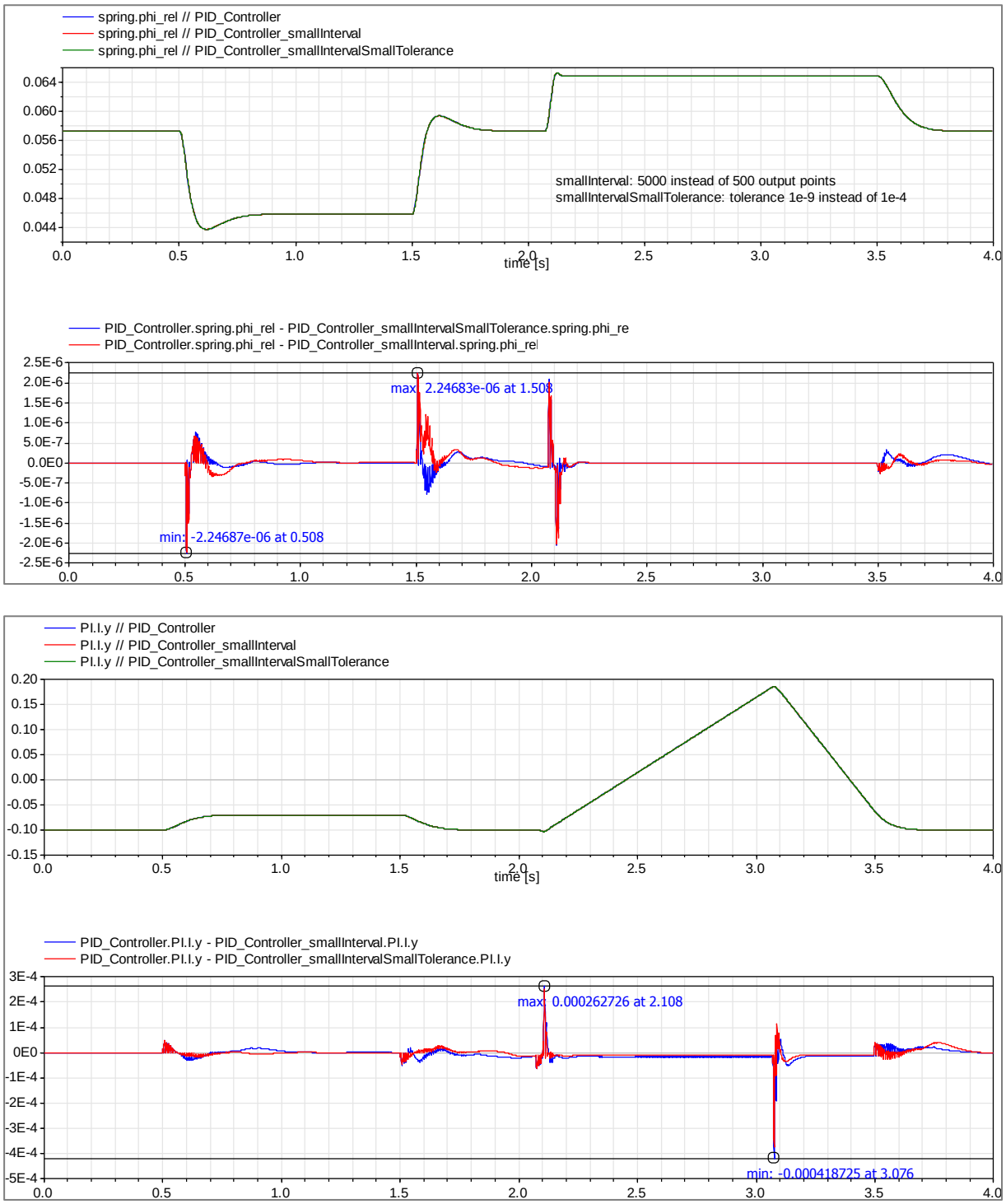
3.1 Creating a first set of reference results in Dymola

Dymola Model Management will be used for generating a first set of reference results.

Needed decisions:

- Define tool version: Dymola 2014 or Dymola 2014 FD01?
  - Dymola 2014 has been used for regression test of MSL 3.2.1
  - Dymola 2014 FD01 is currently used for development and uses new initialization of pre()
- Define experiment settings:
  - Choose a tighter output interval than Dymola default?
  - Choose a stricter solver tolerance than Dymola default?
  - Store double or single precision?

Choosing output interval and solver tolerance is crucial. The problem with smaller output interval is that it leads to larger result files (see section 0). In order to get a feeling for the influence of output intervals and solver tolerance, the PID_Controller example has been simulated with various settings. Two states have been compared:



The errors due to a large output interval (with 500 instead of 5000 output points) are up to 0.2%. This value of 0.2% is the default width of the comparison tube at discontinuity in x-direction. So, the differences illustrate, that it makes sense to reduce the output interval. For this example, the smaller solver tolerance doesn't influence results much.

Proposed way of setting output interval, automatically:

- If the model has an Interval set in the experiment annotation, divide this in halves
- If the model has no Interval set in the experiment annotation, aim for 5000 output points and calculate the interval based on stop time.

This method has to be checked after first comparisons between Dymola and OpenModelica.

3.2 Creation info

The creation log file is intended to document the test setup. By reading the creation log, it should be possible to reproduce reference or test results. Proposed content of the file `creation.txt`:

```
--
[TestCase]
modelName=Modelica.Blocks.Examples.PID_Controller

// Test info
generationTool=Dymola Version 2014 FD01 (64-bit), 2013-10-17
generationDateAndTime=2014-01-08T12:25:32Z
testedPackageURL=https://svn.modelica.org/projects/Modelica/tags/v3.2.1+build.2.rel
ease/Modelica
testedPackageRevision=7178
testDescription=Run all MSL Examples in order to create CSV references
testPC=D2660LG
testOS=Microsoft Windows 7 64bit
testUser=Leo Gall

// Experiment settings (standardized annotation)
StartTime=0
StopTime=4
Interval=0.008

// Experiment settings (tool specific)
Algorithm=Dassl
Compiler_version=Microsoft Visual C++ 2010 Express Edition (10.0)
Switch_Advanced.CompileWith64=0
Switch_Advanced.PedanticModelica=false
Switch_Evaluate=false
Output_textual=false
Output_doublePrecision=false
Output_states=true
Output_derivatives=true
Output_inputs=true
Output_auxiliaries=true
Output_equidistant=true
Output_events=true
Output_debug=false
Output_protectedVariables=false
--
```

This file could be plain text or XML. Content and format of this file has been derived from

- tool-info file for FMI-Cross-Check[6],
- FMI modelDescription.xml,
- Dymola settings and vendor specific experiment annotations.

3.3 Selecting comparison signals to be stored in CSV

The number of variables which can be handled are currently limited (see next section on result file sizes). Therefore, only states should be used for comparison.

What to do, if there are no states? Current idea:

- Modelica Examples: manually specify variables in comparisonSignals.txt
- ModelicaTest: add top level outputs to the model

[6] FMI Cross Check Implementation Notes, https://trac.fmi-standard.org/browser/branches/public/CrossCheck_Results/FMI_Cross_Check_Implementation_Notes_v2_2013_06_26.pdf, page 2

3.4 Result file sizes

The size of result files has influence on:

- speed of CSV-comparison tool
- size of HTML reports
- upload times after test runs
- required storage space on server

A first test run showed the following file sizes:
Result files for 270 examples from Modelica Library (simulation with default experiment settings of Dymola 2014 FD01, i.e. default of 500 output intervals, events are stored, no protected variables are stored):

- Mat-Files: about 320 MB
- CSV-Files: about 30 MB (only dynamic and discrete states included in CSV, therefore some files are empty)

Should the result files be stored under version control?

- Reference results: probably yes. It is very important to log which reference results have been changed.
- Test results: Unsure. Maybe that's too much data.

4 Comparing Results

Compare.exe can be used in tree compare mode.
Example call for a first test run:

```
compare.exe --mode csvTreeCompare --reportdir
"C:\Work\SimulationResults\Dymola\2014_FD01\v3.2.1+build.2.release\Reports\Modelica
" --tolerance 1e-3 --delimiter ";" --verbosity 2 --logfile
C:\Work\SimulationResults\Dymola\2014_FD01\v3.2.1+build.2.release\Reports\Modelica
\log.txt
"C:\Work\SimulationResults\Dymola\2014_FD01\v3.2.1+build.2.release\Modelica"
"C:\Work\ReferenceResults\v3.2.1+build.2.release\Modelica"
```

The resulting report directory contains one HTML file per model, summing up to about 350 MB.
The largest reports are about 20MB, empty reports (because of CSV without result signals) are about 260 KB.

The current version (csv-compare-1.1.0.7078-win32) still has problems in generating the tube. We have to wait for a fixed version.

5 Generating Overview Table

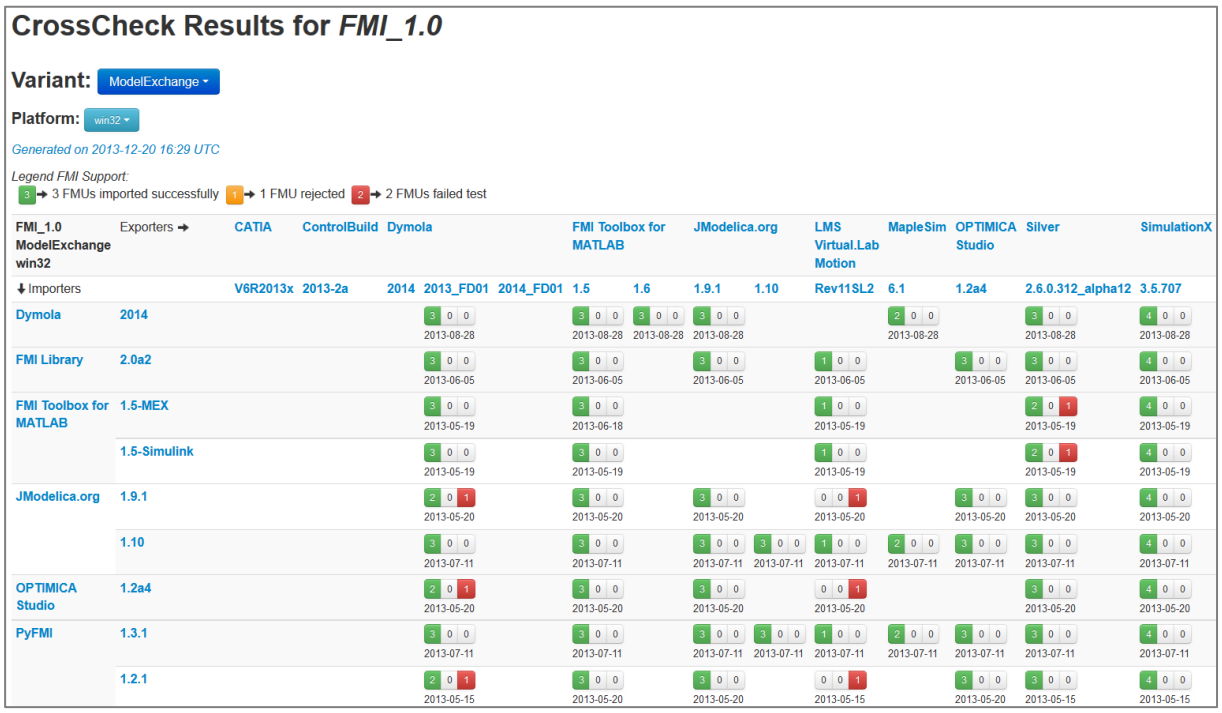
For FMI Cross Check, a Python function is used to generate the overview table[7]. This code should be adapted for generating regression test overviews.

Generating and overview table for MSL regression test should be less complex than FMI Cross Check, because we don't do real cross check, but instead compare to one defined set of reference results.

Proposal: there should be one overview table per Library. So there would be two tables, one for Modelica and one for ModelicaTest.

[7] gen_fmi_web.py from https://svn.fmi-standard.org/fmi/branches/FMISite/dev/templates

Example for a table of FMI Cross Check[8]:



CrossCheck Results for *FMI_1.0*

Variant: ModelExchange

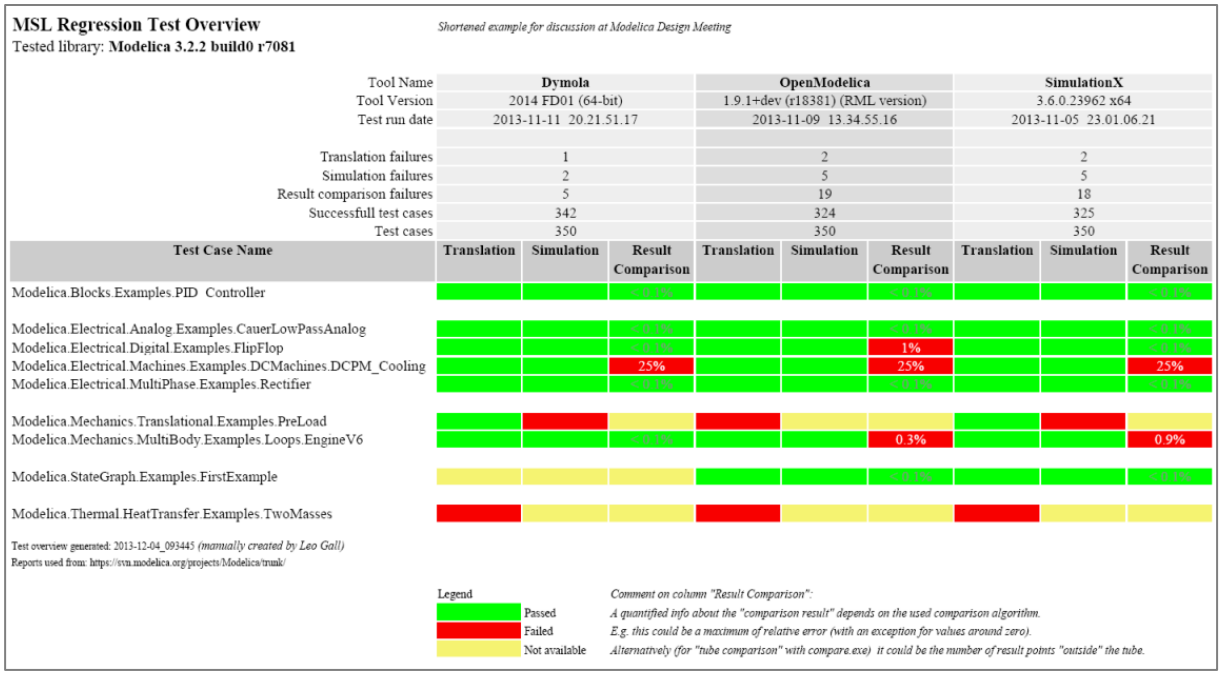
Platform: win32

Generated on 2013-12-20 16:29 UTC

Legend FMI Support:
3 → 3 FMUs imported successfully; 1 → 1 FMU rejected; 2 → 2 FMUs failed test

FMI_1.0 ModelExchange win32	Exporters →	CATIA	ControlBuild	Dymola			FMI Toolbox for MATLAB		JModelica.org		LMS Virtual.Lab Motion	MapleSim	OPTIMICA Studio	Silver	SimulationX
	↓ Importers	V6R2013x	2013-2a	2014	2013_FD01	2014_FD01	1.5	1.6	1.9.1	1.10	Rev11SL2	6.1	1.2a4	2.6.0.312_alpha12	3.5.707
Dymola	2014				3 0 0 2013-08-28		3 0 0 2013-08-28	3 0 0 2013-08-28	3 0 0 2013-08-28			2 0 0 2013-08-28		3 0 0 2013-08-28	4 0 0 2013-08-28
FMI Library	2.0a2				3 0 0 2013-06-05		3 0 0 2013-06-05		3 0 0 2013-06-05		1 0 0 2013-06-05		3 0 0 2013-06-05	3 0 0 2013-06-05	4 0 0 2013-06-05
FMI Toolbox for MATLAB	1.5-MEX				3 0 0 2013-05-19		3 0 0 2013-06-18				1 0 0 2013-05-19			2 0 1 2013-05-19	4 0 0 2013-05-19
	1.5-Simulink				3 0 0 2013-05-19		3 0 0 2013-05-19				1 0 0 2013-05-19			2 0 1 2013-05-19	4 0 0 2013-05-19
JModelica.org	1.9.1				2 0 1 2013-05-20		3 0 0 2013-05-20		3 0 0 2013-05-20		0 0 1 2013-05-20		3 0 0 2013-05-20	3 0 0 2013-05-20	4 0 0 2013-05-20
	1.10				3 0 0 2013-07-11		3 0 0 2013-07-11		3 0 0 2013-07-11	3 0 0 2013-07-11	1 0 0 2013-07-11	2 0 0 2013-07-11	3 0 0 2013-07-11	3 0 0 2013-07-11	4 0 0 2013-07-11
OPTIMICA Studio	1.2a4				2 0 1 2013-05-20		3 0 0 2013-05-20		3 0 0 2013-05-20		0 0 1 2013-05-20			3 0 0 2013-05-20	4 0 0 2013-05-20
PyFMI	1.3.1				3 0 0 2013-07-11		3 0 0 2013-07-11		3 0 0 2013-07-11	3 0 0 2013-07-11	1 0 0 2013-07-11	2 0 0 2013-07-11	3 0 0 2013-07-11	3 0 0 2013-07-11	4 0 0 2013-07-11
	1.2.1				2 0 1 2013-05-15		3 0 0 2013-05-20		3 0 0 2013-05-20		0 0 1 2013-05-15		3 0 0 2013-05-20	3 0 0 2013-05-15	4 0 0 2013-05-15

Overview table for MSL regression test (draft)[9]:



MSL Regression Test Overview
Tested library: **Modelica 3.2.2 build0 r7081**

Shortened example for discussion at Modelica Design Meeting

	Dymola	OpenModelica	SimulationX
Tool Name			
Tool Version	2014 FD01 (64-bit)	1.9.1+dev (r18381) (RML version)	3.6.0.23962 x64
Test run date	2013-11-11 20.21.51.17	2013-11-09 13.34.55.16	2013-11-05 23.01.06.21
Translation failures	1	2	2
Simulation failures	2	5	5
Result comparison failures	5	19	18
Successfull test cases	342	324	325
Test cases	350	350	350

Test Case Name	Translation	Simulation	Result Comparison	Translation	Simulation	Result Comparison	Translation	Simulation	Result Comparison
Modelica.Blocks.Examples.PID_Controller			<0.1%			<0.1%			<0.1%
Modelica.Electrical.Analog.Examples.CauerLowPassAnalog			<0.1%			<0.1%			<0.1%
Modelica.Electrical.Digital.Examples.FlipFlop			<0.1%			1%			<0.1%
Modelica.Electrical.Machines.Examples.DCMachines.DCPM_Cooling			25%			25%			25%
Modelica.Electrical.MultiPhase.Examples.Rectifier			<0.1%			<0.1%			<0.1%
Modelica.Mechanics.Translational.Examples.PreLoad									
Modelica.Mechanics.MultiBody.Examples.Loops.EngineV6			<0.1%			0.3%			0.9%
Modelica.StateGraph.Examples.FirstExample						<0.1%			<0.1%
Modelica.Thermal.HeatTransfer.Examples.TwoMasses									

Test overview generated: 2013-12-04_093445 *(manually created by Leo Gall)*
Reports used from: https://svn.modelica.org/projects/Modelica/trunk/

Legend: Passed; Failed; Not available

Comment on column "Result Comparison":
A quantified info about the "comparison result" depends on the used comparison algorithm.
E.g. this could be a maximum of relative error (with an exception for values around zero).
Alternatively (for "tube comparison" with compare.exe) it could be the number of result points "outside" the tube.

A similar table is available for OpenModelica tests:
https://test.openmodelica.org/libraries/MSL_3.2.1/BuildModelRecursive.html

[8] Screen shot from https://fmi-standard.org/results_FMI_1.0_ModelExchange_win32

[9] https://svn.modelica.org/projects/ModelicaDesign/trunk/MeetingMinutesMaterial/min81_Oberpfaffenhofen/Slides-and-Documents/MSL release test/RegressionTestOverview.pdf