





















## 6 Summary

In this paper, an implementation of the Functional Mock-up Interface for Model Exchange in the JModelica.org platform has been presented. The export functionality enables users to generate FMI compliant models, FMUs from Modelica models and to use them in different FMI compliant tools.

The FMU import is based on Python. Models can be imported into the Python environment, where FMUs are represented by objects of a Python class. The FMU model class provides the user with a one to one mapping of the FMI functions, as well as convenient high-level methods for setting parameter values and simulating models.

This paper also shows that simulation of Functional Mock-up Units using JModelica.org produce results comparable to those produced by a state of the art commercial tool.

Future extensions include support for sparse Jacobians in the FMI specification.

This work was partially funded by the ITEA2 project OPENPROD.

## References

- [1] Johan Åkesson. Optimica—an extension of modelica supporting dynamic optimization. In *In 6th International Modelica Conference 2008*. Modelica Association, March 2008.
- [2] Johan Åkesson, Karl-Erik Årzén, Magnus Gäfvert, Tove Bergdahl, and Hubertus Tummescheit. Modeling and optimization with Optimica and JModelica.org—languages and tools for solving large-scale dynamic optimization problem. *Computers and Chemical Engineering*, 34(11):1737–1749, November 2010. Doi:10.1016/j.compchemeng.2009.11.011.
- [3] Christian Andersson. A new Python-based class for simulation of complex hybrid daes and its integration in jmodelica.org. Master’s thesis, Department of Mathematics, Lund University, Sweden, 2010.
- [4] Willi Braun, Bernhard Bachmann, and Sabina Pross. Synchronous events in the OpenModelica compiler with a petri net library application. In *3rd International Workshop on Equation-Based Object-Oriented Modeling Languages and Tools*, 2010.
- [5] K.E. Brenan, S.L. Campbell, and L.R. Petzold. *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*. Society for Industrial and Applied Mathematics, 1996.
- [6] Dassault Systemes. Dymola web page, 2010. <http://www.3ds.com/products/catia/portfolio/dymola>.
- [7] Hilding Elmqvist. *A Structured Model Language for Large Continuous Systems*. PhD thesis, Department of Automatic Control, Lund University, Sweden, May 1978.
- [8] Jan Eriksson. A note on the decomposition of systems of sparse non-linear equations. *Bit Numerical Mathematics*, 16(4):462–465, 1976. DOI: 10.1007/BF01932730.
- [9] Python Software Foundation. ctypes: A foreign function library for Python, 2009. <http://docs.python.org/library/ctypes.html>.
- [10] J. Hunter, D. Dale, and M. Droettboom. Matplotlib: Python plotting, 2010. <http://matplotlib.sourceforge.net/>.
- [11] Center for Applied Scientific Computing Lawrence Livermore National Laboratory. SUNDIALS (SUite of Nonlinear and Differential/ALgebraic equation Solvers), 2009. <https://computation.llnl.gov/casc/sundials/main.html>.
- [12] Sven Erik Mattsson and Gustaf Söderlind. Index reduction in differential-algebraic equations using dummy derivatives. *SIAM J. Sci. Comput*, 14(3):677–692, May 1993.
- [13] T. Oliphant. Numpy Home Page, 2009. <http://numpy.scipy.org/>.
- [14] R.E Tarjan. Depth-first search and linear graph algorithms. *SIAM J. Computing*, 1(2):146–160, 1972.
- [15] The Modelica Association. The Modelica Association Home Page, 2010. <http://www.modelica.org>.
- [16] Johan Ylikiiskilä, Johan Åkesson, and Claus Führer. Improving Newton’s method for initialization of Modelica models. In *8th International Modelica Conference*, 2011.