PyModSimA

PySimulator and Modelica Based Collaborative System Design Simulation Analysis Environment for Energy System Applications

State of the art – Background

The Systems for Green Operations ITD of Clean Sky aims to demonstrate substantial environmental and economic benefits of more electric aircraft systems technologies.

However, the design and validation of such highly integrated systems needs more co-operative and collaborative development processes involving aircraft, engine, and equipment manufacturers.

At the current state-of-the-art it is often the case that each application area uses different design and modeling tools, which cannot easily communicate and connect to each other for total system design and validation.

A related problem is the current dependence on specific tool vendors which hampers total system design when tools from several vendors need to be used.

Total system validation is usually difficult since unit and regression testing can normally only be performed within one tool at a time, instead of for the whole system design.

Connecting models different suppliers and simulation tools for a unified system simulation is often difficult and costly, if at all possible.

Objectives

An important goal is to enable a more co-operative and collaborative development processes involving aircraft, engine, and equipment manufacturers. The design process has to be supported through advanced modeling and simulation capabilities.

Therefore the goal of the Green Operations ITD of Clean Sky consortium is to define standardized modeling methods and tools in each phase of the energy system design process.

This is examined for instance in the Use Case: "Development of a modular energy system simulation tool-chain" as part of SGO WP2. In this example, collaborators from the aircraft industry use the FMI Standard in order to integrate their models in a total system simulation.

Whereas this work has largely been conducted successfully, it also revealed problems and a need for more light-weighted and freely accessible simulation tools for collaborative modeling and design processes of total complex systems.

In particular there is a need for innovations in the following areas:

- The dependence on one specific tool vendor in the design process should be greatly reduced to not hamper total system design.
- Unit and regression testing should be applied over different tools in order that models can be safely utilized in different tools.
- Connecting models from different suppliers and simulation tools for a unified system simulation should be possible in a freely accessible tool for widespread use and dissemination.

The basis to reach these goals within this project is PySimulator (https://github.com/ PySimulator/ PySimulator/), a Python based, open source environment for simulating FMU's and running other simulation engines such as Dymola or OpenModelica (www.openmodelica.org).

The central idea of PySimulator is to provide a generic framework to perform simulations with different simulation engines in a convenient way, to organize the persistent storage of small and huge result data sets, to provide plotting and other post-processing feature such as signal processing or linear system analysis, and to export simulation and analysis results to other environments such as Matlab.

The major innovation of PySimulator is its plugin system: Nearly all operations are defined as plugins with defined interfaces. Several useful plugins are already provided, but anyone can extend this environment by his/her own plugins and there is no formal difference to plugins already provided. Thus, the PySimulator open source tool can be used by each partner independently of any commercial simulation environment used for FMU generation.

The work in this project aimed at significantly improving the capabilities of PySimulator by:

- Adding an additional simulation plugin as well as improving the existing PySimulator framework and OpenModelica plugin.

• Developing capabilities in PySimulator to perform unit and regression testing over models from multiple tools.
• Developing capabilities in PySimulator for simulating connected FMUs (i.e., compiled and exported models) from several tools. This is needed in total system model simulation.

The developed framework is the first open-source project that supports simulation of connected FMUs for both Model Exchange and Co-Simulation. Also, no other commercial framework exists that supports direct simulation of connected Model Exchange FMUs.

Description of work

Develop a Simulator plugin for one additional commercial simulation engine for Modelica models - the Wolfram SystemModeler. There is an existing plugin interface for Simulator plugins in PySimulator. Wolfram SystemModeler will be interfaced according to this PySimulator plugin interface.

Develop a plugin to perform unit and regression testing over two and more models in PySimulator (especially different versions of the same model, FMUs generated from the same model by different tools, simulating a Modelica model with different simulation engines). The test definition shall be defined with a GUI and textually by scripting. Output of a typical test is a report including a measure of the difference of the simulation results, detailed information about the differences and result plots. Develop parallelization of the simulation and analysis runs for unit and regression testing on a multi-core machine.

Functionality for connecting FMUs (FMI standard version 2.0) textually and using at least a connection GUI. Handling of many input/output signals that need to be connected shall be possible. The connection schema has to be saved for reloading it in a following session. Simulation of connected FMUs from both for Model Exchange and Co-Simulation including handling of algebraic loops for Model Exchange. This means especially to implement a Simulator plugin for the Co-Simulation Master algorithm that controls the FMUs for Co-Simulation.

Results

The main results for this project are:
- Support for an additional simulation plugin in PySimulator as well as improving the existing PySimulator framework and OpenModelica plugin.
- New capabilities in PySimulator to perform unit and regression testing over models from multiple tools and different vendors.
- New capabilities in PySimulator for simulating connected FMUs (i.e., compiled and exported models) from several tools for both Model Exchange and Co-Simulation. This enables total system model simulation.

This project provides the first open-source solution for regression testing and simulation of connected FMUs.

The first existing solution to simulate connected FMUs for Model Exchange was developed during this project.

a) Timeline & main milestones

Year 1 – proof of concept prototypes

WP2 - Simulation Environment with Plugin, Unit and Regression Testing
- First plugin for Wolfram Simulator
- First prototype for regression testing

WP3 - Simulation of Connected FMUs
- First plugin prototype for handling connected FMUs for Co-Simulation

Year 2 – improved and final delivered software

WP2 - Simulation Environment with Plugin, Unit and Regression Testing
- Improved plugin for Wolfram Simulator
- Improved plugin for regression testing

WP3 - Simulation of Connected FMUs
- Improved plugin for connected FMUs for both Model-Exchange and Co-Simulation

WP4 - Tuned Software and Final Report
- Final plugin for Wolfram Simulator
- Final plugin for regression testing
- Final plugin for connected FMUs for both Model-Exchange and Co-Simulation
- Final report and documentation

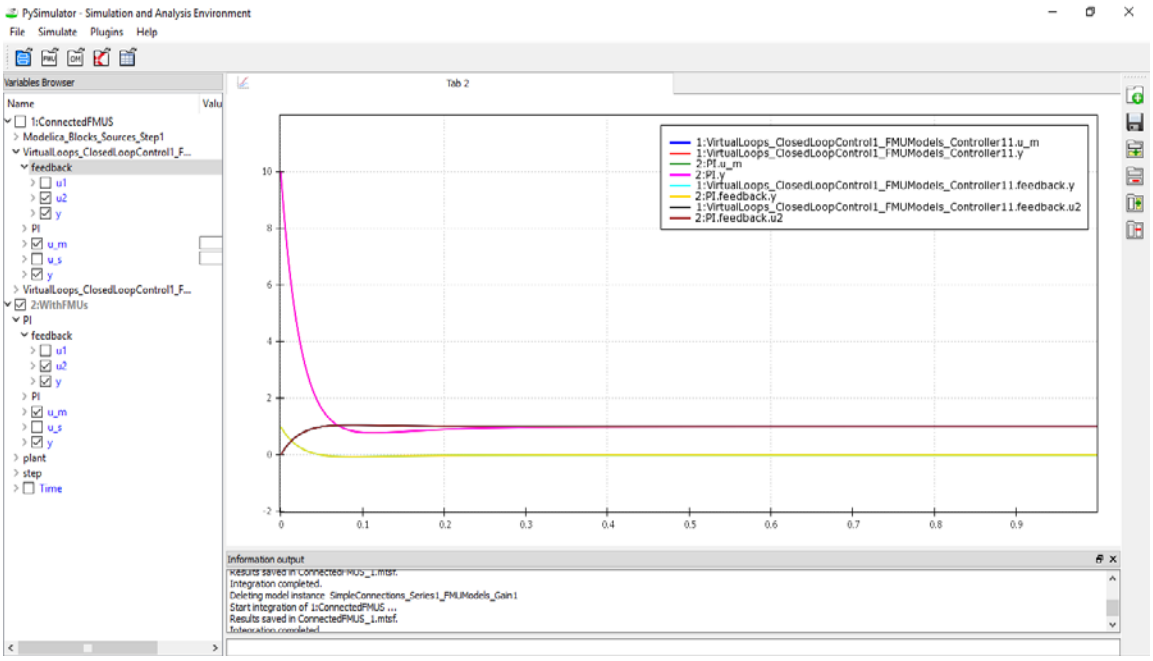
b) Environmental benefits

Total system simulation via simulation of connected components in FMU form and validation of simulation results via regression testing will allow system designers to better understand and optimize their systems. This will result in better products that will perform more efficiently and in the agreed environmental boundaries specified by European law.

c) Maturity of works performed

The developed open-source software was tested and validated by Topic Manager and their feedback was used to further improve it towards the final versions. The software proved to be robust enough to run on medium sized examples and further testing was planned to include industrial relevant benchmarks.

PyModSimA - Connected Model Exchange FMUs in PySimulator



Project Summary

Acronym : PyModSimA

Name of proposal: PySimulator and Modelica Based Collaborative System Design Simulation Analysis Environment for Energy System Applications

Technical domain:	Mathematics and computer science, civil engineering, other sciences
Involved ITD	Systems for Green Operations (SGO)
Grant Agreement:	632413
Instrument:	Clean Sky JU
Total Cost:	249,990
Clean Sky contribution:	186,381
Call:	FP7 Clean Sky Joint Undertaking SP1-JTI-CS-2013-02-SGO-02-064
Starting date:	2014-04-01
Ending date:	2016-09-30
Duration:	2 years 5 months
Coordinator contact details:	Prof. Peter Fritzson LINKOPINGS UNIVERSITET +46 708 281484 peter.fritzson@liu.se
Project Officer:	Antonio Vecchio Antonio.Vecchio@cleansky.eu

Participating members:

Linköping University - LIU

SICS East Swedish ICT AB - SICS East