Export of Modelica models to the ProMoVis environment

Jesper Moberg

April 13, 2012

Contents

Chapter 1

Templates

There is text in first chapter

1.1 About Template

This template provides a sample layout of a Standard LaTeX Report.

The front matter has a number of sample entries that you should replace with your own.

1.2 Document Class Options

The typesetting specification selected by this document template uses the default class options. There are a number of class options supported by this document class. The available options include setting the paper size, the point size of the font used in the document body and others.

1.2.1 Customizing Class Options

Select ‘Insert’, ‘Document Properties ...’, the ‘Generic’ tab and then modify desired class options in appeared dialog. Changes will be applied after pressing the ’OK’ button.

Chapter 2

Description of JModelica

2.1 Background

JModelica is an open-source Modelica environment, written in python, for compilation and simulation of Modelica models. Through its python front-end it provides an easy to use, still powerful way to perform complex tasks on compiled models.

2.2 Example model

When explaining we sometimes reference this model to clarify stuff, fixme

2.3 How the export tool uses JModelica

JModelica is used to compile the Modelica models to JModelicas JMU representation [1]. This representation is internally represented as a, possibly, nonlinear DAE. This model, can through the JModelica environemnt be linearized and a model with the following represenation can be extracted:

$$E * dx = A * x + B * u + F * w + g \tag{2.1}$$

The current version of JModelica[2] does not separate between internal states and outputs. Naturally, the x and dx vectors represents the states and outputs of the linearized system. The u vector represents declared inputs, w is modeled disturbances (FIXME are they modeled in modelica, or attached at simulation time cant find anything regarding this in modelica specification) and finally g is a constant bias.

The linearization also outputs some useful information that we later use in the generation of ProMoVis scenarios:

- *State names*, corresponding to the declared variable names from the original Modelica file.

- *Input names*, corresponding to the declared input names from the original Modelica file.
- *Working points* for the linear model *dx0*, *u0* and *x0.* Which is used to provide feedback for the user regarding the (FIXME)sanity of the linearized model.

Bibliography

[1] JModelica.org python api docs

[2] 1.7b2