Visualizing Simulation Results from Modelica Fluid Models Using Graph Drawing in Python

Marcus Fuchs   Rita Streblow   Dirk Müller

RWTH Aachen University, E.ON Energy Research Center, Institute for Energy Efficient Buildings and Indoor Climate, Aachen, Germany, mfuchs@eonerc.rwth-aachen.de

Models of large thermo-fluid networks can be useful to better understand the dynamic behavior of complex systems. Yet, numerical outputs and line plots of individual variables may not be sufficient ways of processing the simulation results for the user. Thus, the aim of this paper is to present a visualization approach by means of graph drawing. To demonstrate the approach, we use an example from the Modelica Standard Library and the use case of a district heating system model. We parse the Modelica model code to generate a `System` graph that represents the model structure and its graphical layout. The graph drawing subsequently visualizes the results for every time-step. In the examples, we vary line thickness to visualize mass flow rates between two nodes and line color to show temperatures of the medium. We argue, that this approach can be a useful tool for modeling and analysis.

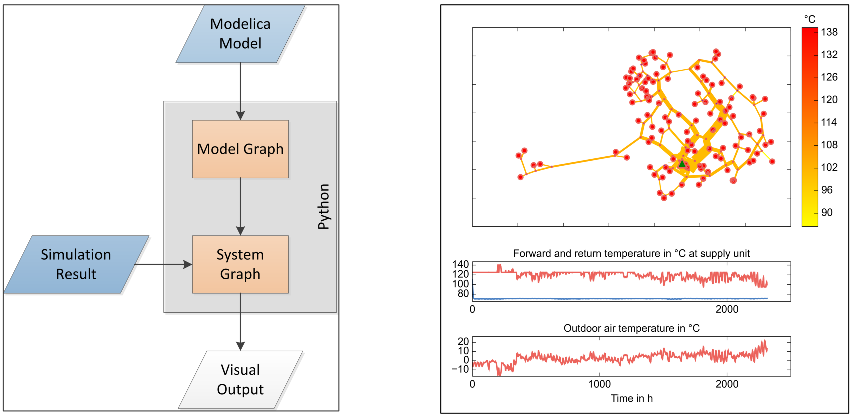

Figure 1. Process scheme and example of visual output for a district heating network simulation

The left of Fig. 1 shows a process scheme, illustrating how a Modelica model is represented in the form of a `Model` graph in Python and subsequently converted to a `System` graph. The `System` graph is designed to give a clear representation of the system layout, add data from simulation results, and provide a visual output according to the specifications of the user. One example of such visual output is given on the right of Fig. 1, generated from a district heating network model and its simulation result file. In a next step, such visualization output images are be combined into a video animation in order to present the dynamic behavior of the system. In addition to the visualization functionalities, we plan to use the Python graph structure as a foundation for automated generation of Modelica models, thus providing an integrated workflow to handle the complexity of input and output data for large thermo-fluid network modeling.