An open toolchain for generating Modelica code from Building Information Models

Matthis Thorade[1] Jörg Rädler[1] Peter Remmen[2] Tobias Maile[3] Reinhard Wimmer[3] Jun Cao[3] Moritz Lauster[2] Christoph Nytsch-Geusen[1] Dirk Müller[2] Christoph van Treeck[3]

[1]Berlin University of the Arts (UdK),
{m.thorade,jraedler,nytsch}@udk-berlin.de
[2]RWTH Aachen University, E.ON Energy Research Center, Institute for Energy Efficient Buildings and Indoor Climate, Aachen, Germany
{premmen,mlauster,dmueller}@eonerc.rwth-aachen.de
[3]RWTH Aachen University, Institute of Energy Efficient Building, Aachen, Germany
{maile,wimmer,cao,treeck}@e3d.rwth-aachen.de

Buildings become increasingly integrated to reduce energy and peak power and to increase occupant health and productivity, leading to complex building design. Building Performance Simulation (BPS) is a key element in the design of energy efficient buildings, and there is increasing interest in using the Modelica modelling language for BPS. The International Energy Agency's Energy in Buildings and Communities Programme (IEA-EBC) coordinates development of BPS in Modelica in the project "Computational Tools for Building and Community Energy Systems", also known as Annex 60.

Even with advanced BPS libraries at hand, developing BPS models and collecting required input data are time-consuming and error-prone processes, preventing practitioners from using BPS more extensively in standard planning processes. Building Information Modelling (BIM) is a well established technology to model and manage the digital representation of a building over its entire lifecycle. Reusing existing Building Information Models (BIM) as basis for Building Performance Simulation (BPS) has the potential to make BPS model development and subsequent simulation easier, faster and more reliable.

Activity 1.3 of the Annex 60 project is working on an open-source toolchain that can semi-automatically generate code for BPS Modelica models from a BIM data source. In this contribution, we give an overview over the toolchain and the involved steps as well as a more detailed description of those parts of the toolchain that are used for controlling the workflow and for the actual generation of Modelica source code files.