

## AN OPEN FRAMEWORK FOR INTEGRATED BIM-BASED BUILDING PERFORMANCE SIMULATION USING MODELICA

P. Remmen<sup>1a</sup>, J. Cao<sup>1b</sup>, S. Ebertshäuser<sup>2</sup>, J. Frisch<sup>1b</sup>, M. Lauster<sup>1a</sup>, T. Maile<sup>1b</sup>,  
J. O'Donnell<sup>3</sup>, S. Pinheiro<sup>3</sup>, J. Rädler<sup>4</sup>, R. Streblow<sup>1a</sup>, M. Thorade<sup>4</sup>, R. Wimmer<sup>1b</sup>,  
D. Müller<sup>1a</sup>, C. Nytsch-Geusen<sup>4</sup>, C. van Treeck<sup>1b</sup>

<sup>1</sup>RWTH Aachen University, <sup>1a</sup>E.ON ERC EBC and <sup>1b</sup>E3D, Aachen, Germany

<sup>2</sup>Karlsruhe Institute of Technology, Department BLM, Karlsruhe, Germany

<sup>3</sup>School of Mech. & Materials Eng. and Electricity Research Centre, UCD, Dublin, Ireland

<sup>4</sup>Berlin University of the Arts, Institute for Architecture and Urban Planning, Berlin, Germany

### ABSTRACT

Building Performance Simulation (BPS) is a key element in the design of energy efficient buildings. The Modelica modelling language is becoming increasingly important in the BPS field. The International Energy Agency's (IEA) Annex 60 project, "New Computational Tools for Building Performance Simulation", coordinates the development of BPS in Modelica. However, data collection and model generation are time-consuming processes, often leading to uncertainty. One subtask of the Annex 60 project focuses on the use of Building Information Modelling (BIM) as basis for building performance simulation. This paper presents an open framework for automated Modelica model generation from a BIM data source. The project outcomes include open-source software tools and a Model View Definition (MVD) for IFC to BPS information exchange with Modelica.

### INTRODUCTION

Higher energy standards and efficiency requirements impose greater complexity on the building design process. Consequently, the design process of new and retrofit buildings has become increasingly complex and requires advanced digital planning tools. A major issue involves the linking of commonly used software, and improving the information exchange between the different experts involved in the design process.

Building Performance Simulation (BPS) is a key element in the design of energy efficient buildings. Recently the non-proprietary, object-oriented and equation based modelling language Modelica has become increasingly relevant in the field of BPS (Wetter et al. 2014).

The International Energy Agency's (IEA) Annex 60 project "New Computational Tools for Building Performance Simulation" is part of the Energy in Buildings and Communities Programme (IEA-EBC), and coordinates the development of BPS in Modelica (Wetter and van Treeck 2015). One subtask focuses on harmonizing and unifying model development in Modelica. Such development is currently fragmented and several model libraries from different institutions exist. Each library has its own focus and thus

contains models for diverse application. Two of the libraries, AixLib from RWTH Aachen University (Constantin, Streblow, and Müller 2014) and BuildingSystems from UdK Berlin (Huber, Nytsch-Geusen, and Schünemann 2012), serve as target libraries for the workflow shown in this paper. This process is applicable to further libraries that take part in Annex 60.

In addition to thermal simulation tools, architects and engineers use BIM-based CAD tools to design buildings and energy systems. Building Information Modelling (BIM) is a technology used to model and manage the digital representation of a building over its entire lifecycle (Eastman 2011).

However, data collection and model generation for BPS is time consuming and often leads to uncertainty and error (Bazjanac et al. 2011), and there is limited integration of BPS with BIM based tools found in the wider design area. These issues restrict the more widespread adoption of BPS amongst practitioners during the planning and design stages.

No open, standardized and automated or semi-automated processes for generation of Modelica models from BIM exist today. In order to simplify model generation, one subtask of the Annex 60 project is concerned with using BIM systems as a direct data source for Modelica simulation. Previous research focuses on geometry and single specific Modelica libraries (Yan et al. 2013; Jeong et al. 2015), while in this IEA Annex 60 project the focus is also on building HVAC systems and multiple libraries.

Two major challenges must be considered when developing a semi-automated data exchange process. First, mapping data from BIM to BPS requires expert knowledge of both BIM and BPS, which are quite different domains, in order to define consistent mapping rules. This mapping needs to consider different aspects including building geometry and heating and ventilation components (Wimmer et al. 2015).

Second, model preparation for BPS is not only a matter of data conversion but also requires sufficient initial boundary conditions, consistent system models and reliable parameter sets at the system level.

These aspects, which include simulation knowledge, are challenging to encapsulate within a semi-automated process. They require flexible interfaces that allow the user's knowledge and additional information to be added in an easy-to-use manner.

It is necessary to keep the exchange process from BIM to Modelica flexible to allow the integration of further model libraries and BIM-tools. The open framework should feature suitable interfaces that allow developers to link their models and tools to the framework as well as enable users to add information in all process steps easily.

This paper provides an overview of an open source framework to connect BIM-based architecture and engineering software with building energy performance simulation in Modelica. The first sections describe the basic assumptions, overall application domain and software foundation. The methodology section outlines the different steps in our tool chain from BIM to Modelica and leads to the workflow demonstration. The paper concludes with the current limitations, a summary of the framework and potential further developments.

## PROCESS OVERVIEW & ITS ACTORS

An obligatory requirement for the framework is a valid and well-formed BIM model. We distinguish between geometry, building physics and HVAC components as semantic model parts within a BIM. All three semantic parts require a high model quality. The following section describes the different parts of the information model and prerequisites for appropriate model quality. A well-defined BIM is the foundation of the tool chain (Wimmer et al. 2015; Cao et al. 2015). A well-defined BIM in the context of BPS contains all walls and windows with respect to different constructions as well as the HVAC

system and space boundaries. As model quality is highly dependent on decisions during model generation, this section also gives an overview about the process from a user's view. An overview of the process and its actors can be taken from Figure 1 and is described in the following section. This includes model generation, model enrichment and compliance checking for the information model, as well as the application to our framework.

### Model generation

The BIM model commonly begins with the architect designing a designated space and usage structures (left side Figure 1). Thereby all basic model semantics related to the building components (geometry and building physics) are defined within the main morphology of the building. The model is created with BIM-based CAD software. The use of suitable software allows the integration of further involved technical actors, in our case the HVAC engineer. The architect benefits by using the IFC (Industrial Foundation Classes) format, as IFC is a widely used, non-proprietary and established BIM format (International Organization for Standardization 2013). The HVAC engineer adds information about the HVAC SYSTEMS to the BIM model. This includes geometry (e.g. the structure of the piping system) and semantic information (e.g. the pipes material). The architect or a specialist on data exchange handling then coordinates outgoing model data. He is also responsible for merging enriched incoming partial models or other data to the BIM. This is crucial to the overall consistency of the BIM model in these collaborative activities.

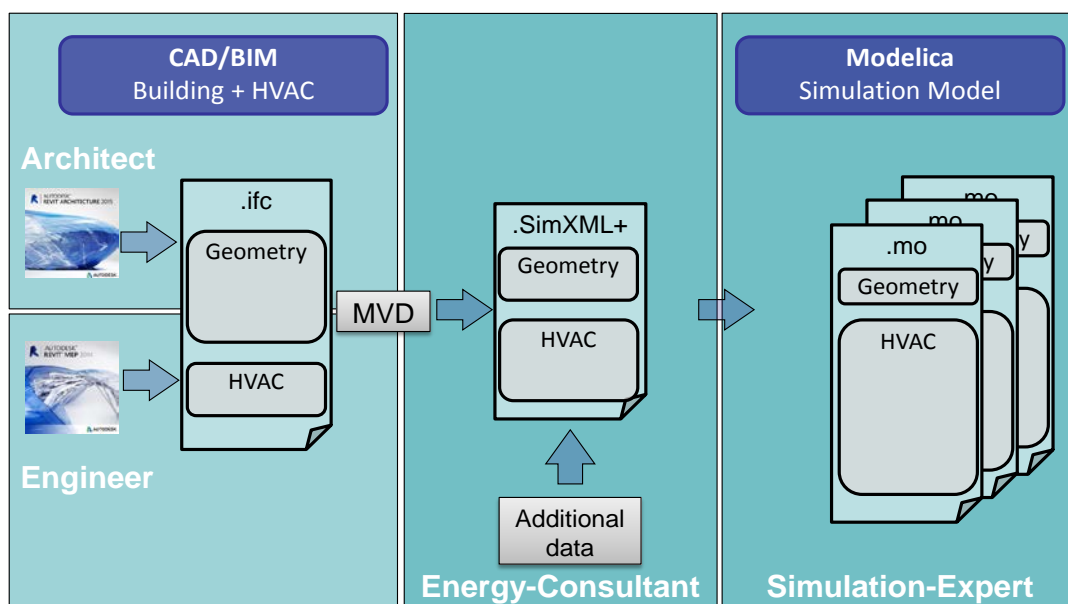


Figure 1: Actors involved in the framework from BIM to Modelica Simulation

### *Model enrichment*

The energy consultant uses a specific Model View Definition (MVD), which we develop within the Annex 60 project, to read the relevant data for Modelica simulation (middle part in Figure 1) (Cao et al. 2015; Wimmer et al. 2015). A MVD is a subset of the IFC-schema that defines discipline-specific exchange requirements (Eastman 2011). The different disciplines are thus concerned with own specific modelling details. For example, the user needs spatial geometry and building information defined in the MVD to define space boundaries and thermal zones.

Building technology systems need information on the spatial hierarchy and need to align the overall building model structure. The MVD defines the required data from the IFC file. We defined the exchange requirements with a focus on building simulation in Modelica. The energy consultant collects this data from the IFC file and migrates it into an intermediate file format, called SimXML. The information model behind SimXML provides the possibility to add additional entities and information, especially related to the building HVAC systems semantics to SimXML. For example the SimXML file can be extended with specific information for Modelica simulation. The methodology part of this paper describes technical details of this transformation.

### *Checking the BIM*

During the tool chain based framework, the information model passes through several checks, to ensure model quality. After generation of the IFC file, an IFC model checker performs a schema check. This ensures that the IFC file conforms to the IFC schema. The next check validates the information covered by the IFC file. This comprises different validation rules, especially concerning the setup of the building technology system.

For example, one validation rule checks that all pipes are connected to a circuit. This check helps to reveal inconsistencies in the model before the conversion to SimXML. The energy consultant then can add the missing information. A third check examines the SimXML file for compliance with the SimModel schema and generates error messages indicating where modifications are needed. Besides detecting errors, the tool also recognizes missing information and other possible conflicts, and informs the user automatically with respective warnings or errors. Typical errors are unconnected pipes in a hydraulic circuit.

Once all model issues are resolved, the energy consultant passes the model to the code generation part of the framework, to automatically generate valid Modelica models.

## SOFTWARE BASIS

Within the framework presented here, we use different software languages and data formats. The following section presents different software languages, tools and data models for this application.

### *Industrial Foundation Classes (IFC)*

IFC4 is the latest version of ISO standard 16739 for describing computer models of buildings (International Organization for Standardization 2013). It contains several improvements compared to its predecessor IFC 2x3. Besides the model standard itself, concepts for further specification of model content quality and richness of details have been provided. In context of building physics, the new version of IFC extends the possibilities for describing different space boundaries. Furthermore, the extended HVAC semantics were also significantly improved, together with the corresponding hierarchy and topology. Many new component entities needed for HVAC design, like *IfcValve* or *IfcBurner* have been included.

Another main modification affects the Model View Definition in IFC. In contrast to the superseded 'singular' Coordination View (CV) in IFC 2x3 the concept in IFC 4 aims at splitting it up into a Reference View (RV) and into a Design Transfer View (DTV). While the RV's main purpose is to define the subset of the IFC4 schema, the DTV is developed to edit specific elements.

### *SimModel*

SimModel is a domain data model for whole building energy simulation, related to EnergyPlus (O'Donnell et al. 2011). This XML-based data model focuses on the data needs for building energy simulation and is called SimXML. It closely aligns with the IFC data model (e.g., an *IfcFlowMover* corresponds to a *SimFlowMover*) removing some of the redundancies and simplifying relationships between objects. Common elements in SimModel retain their mapped IFC entity ID. The current version supports translation from IDD (EnergyPlus Input Data Dictionary), gbXML (Shafee and Dickinson 2011), IFC and SDD (Standards Data Dictionary) (Maile, Haves, and See 2015).

### *Python*

Python is a high-level programming language with object-oriented, functional and imperative character. Python has advantages for use in back end application of the tool chain. For instance, the code is highly readable and the syntax is easy to learn, enabling users without extensive programming experience to work with the framework. Moreover, Python is widely accepted for pre- and post-processing by Modelica developers.

## C++

C++ is a generic object-oriented programming language, with advanced programming features, including objects, inheritance, and polymorphism, while also providing the facilities for low-level memory manipulation. C++ is very popular, particularly for the development of applications that require speed and/or access to some low-level system features. We use C++ as the development language for libSimModel, enabling SimModel parsing and data translation with a bias toward efficiency and flexibility of use.

## Modelica

Modelica is a free, open-source and non-proprietary modelling language for complex technical systems. Models are composed of as equation-based, acausal, and object oriented blocks. The different blocks are connected by interfaces and a combination of these blocks form a full model. The well-formed models are translated into objects that can be processed by a simulation engine.

## METHODOLOGY

Figure 2 illustrates the overall methodology. Various software applications can export building design data in the IFC format. Thereby the two key domains are architecture and the HVAC system. The focus of the presented framework is on the conversion of the HVAC system, however we provide basic functions to process geometric data. Once data from both domains is in SimModel, the next step is to map objects and properties to a specific Modelica library. Via the API to libSimModel, the mapped objects and properties are passed to a Python based tool, called CoTeTo (Code-Templating-Tool). CoTeTo then performs the last set of data manipulations to result in a Modelica file that is ready for simulation.

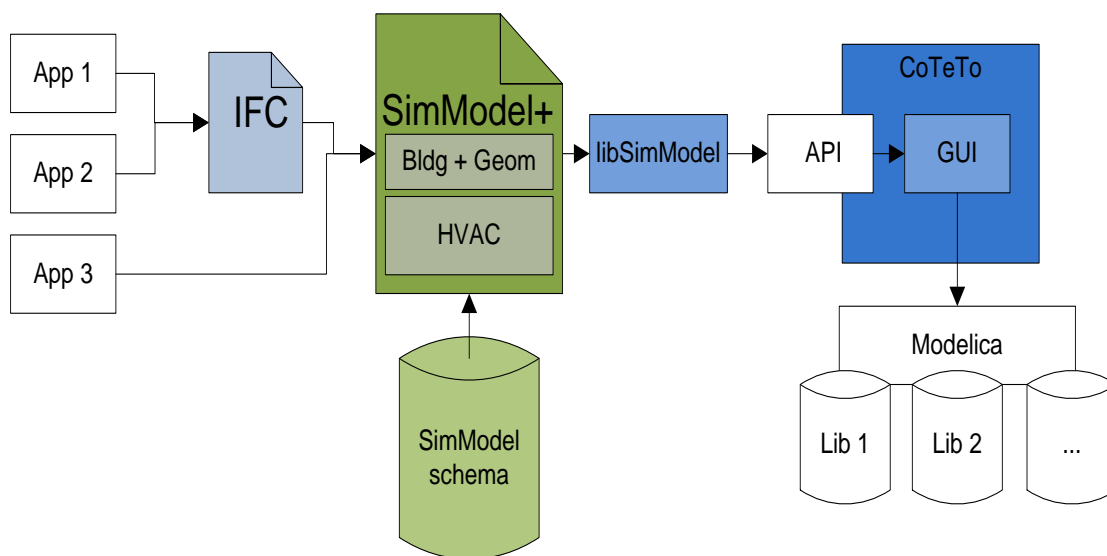


Figure 2: Methodology overview, from the information basis to Modelica

## IFC to SimModel

Figure 3 presents the detailed process of populating information into SimModel. We provide two alternatives to populate a SimXML file. The first option is to use detailed information provided by an architect in a CAD software. Many CAD applications have the functionality to write IFC4 files. For building energy performance simulation, we are not only interested in architectural but also HVAC data.

The conversion of the checked and valid IFC model to SimModel takes place in two steps. The first step is to convert building geometry and HVAC data. For building geometry data we are using existing tools such as the Space Boundary Tool (SBT) and Simergy (Rose and Bazjanac 2013). This is rather straight forward as the semantic and structural overlapping allows preservation of information in IFC and SimModel.

The conversion of HVAC data has been developed for this project. It first checks the HVAC representation with a predefined set of validation rules in the IFC data format and then converts it into SimXML. The second step is to combine the separately converted parts of the IFC model (building physics and HVAC) into one well-formed and schema valid SimModel XML file. The conversion of HVAC data is indicated as KIT Tool in figure 3.

SimModel may also be populated with statistical data. In cases where existing buildings are considered, information is often sparse. To create archetype buildings with minimal information input (building type, year of construction, net-leased area, number and height of storeys) we developed a tool called TEASER.

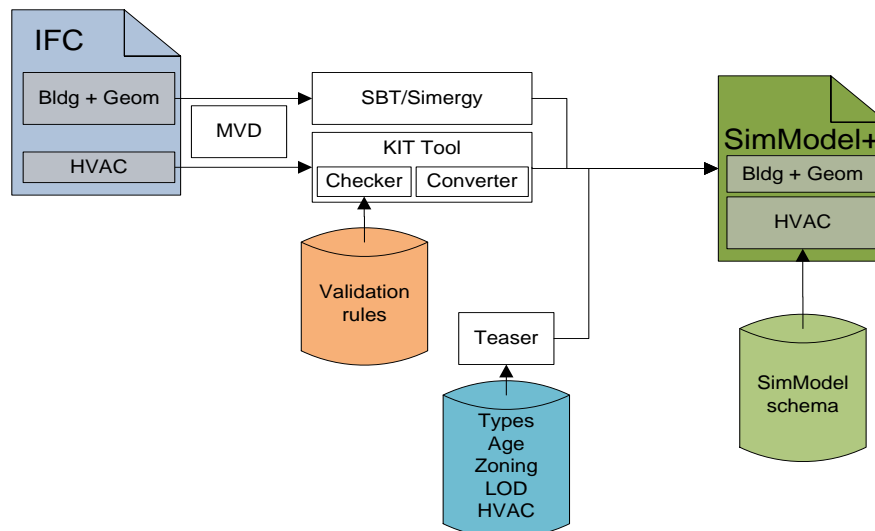


Figure 3: Transformation from IFC to SimModel within the framework

### SimModel to Modelica

To accomplish the link between the rigid data structure in SimModel with the flexible data representation on the Modelica side, mapping rules are needed (Wimmer et al. 2014; Wimmer et al. 2015). Such mapping rules enable reproducible translation of SimModel content into a Modelica representation.

The rules connect the objects and parameters with the relevant input data in the specific Modelica library. For this step, different mapping rules are developed.

In total, six mapping rules exist:

1. One to One
2. Many to One
3. One to Many
4. Gap
5. Transformation
6. Combination

The first rule describes a mapping of identical objects/ parameters, such as the parameter describing the efficiency of a boiler. A combination of several objects to one object is provided for with the second rule. For example, the ground reflection coefficient is represented in SimModel for each month and in Modelica as a single value. The third rule describes the opposite case. The fourth rule deals with missing objects/ parameters and extends the data in SimModel by the required data. A specific example is an expansion vessel. The expansion vessel is not represented in SimModel. Rule number five maps identical parameters, but with different representations in SimModel and Modelica, such as a unit conversion. The last rule represents a possible combination of the rules. The user has to define the algorithm between each object/parameter between SimModel and Modelica. Defining these mapping rules is a critical step, because the user needs to understand the definition of all the relevant objects/parameters on both sides. Thus, the user

requires high-level domain knowledge to understand the structure of SimModel and the Modelica library.

### Code Generation

The goal of the project is to enable the instantiation of Modelica models from different model libraries using a common data source. Each library has to be handled separately because of different modelling approaches. These libraries are currently under development and are likely to change in the future as well. A flexible data conversion framework is required to allow for future changes. Thus, the tool chain should allow flexible output components for different libraries in multiple versions as well as flexible input components. These should be easy to maintain, even for non-programmers.

To connect and control different parts of the framework we developed a Python based tool called Code-Templating-Tool (CoTeTo). The workflow of CoTeTo and the coupling to other tools within the tool chain is shown in Figure 4. We designed CoTeTo to be accessed by graphical, command line and library level interfaces. The multiple access points open the framework to a huge community.

The following section will give an overview of the components and their functionality. We divide the CoTeTo into input and output components.

The input component is referred to as the *Data API*. It defines a prescribed way to fetch data sets from a data source. Each output component depends on a specific data API. Although we use the Python language to write the CoTeTo, data API functions and filters can interface to other languages. In the case of our project the Building Information Model is saved in SimXML format [see SimModel description]. We access the data contained in the SimXML-file by an API to a C++ skeleton class representation of the SimModel Information Model (libSimModel) (Cao et al. 2014). The skeleton class also provides the previously described mapping

functionality. The API to libSimModel allows the linking of different mapping rules, corresponding to different Modelica libraries, to the SimXML file. The API returns the mapped information in a tree structure. The representation of components in tree structures is intuitive and reflects the structure of BIM. The data flow through the libSimModel API is not limited to a single direction. It is also possible to manipulate parameter values and send them back to the corresponding C++ class, thus they can be saved in the SimXML format.

Once all relevant data has been loaded into CoTeTo, it is passed to the output component, called *Generator*. We designed the Generator to contain all items needed to generate the code for a specific Modelica library. This includes the mapping rules files, filter functions, additional information and text templates. We can easily exchange Generators between different installations, as they are zip-files with a defined structure.

In our experience, some data needs manipulation. One example of data manipulation relates to geometric information. The Modelica libraries we considered contain different types of building models and thus need different pre-processing steps. Another example relates to control elements for HVAC components. The design of the control strategy is dependent on the modelling approach of the HVAC element. For this purpose, Generators can include filter functions (Python code) that we call between the data API and the templates. The filters are custom-built to the used library. As mentioned above, they may include simplification of geometric relationships and calculation of model specific parameters.

One major challenge in the automated generation of Modelica models is the flexibility of Modelica. Generally said, setting up useful models needs the knowledge of an experienced user. We are following the approach to encapsulate this knowledge in library specific meta-models and templates.

One essential task here is the appropriate connection of components. The API returns the connection information corresponding to the SimModel ontology, which differ from the one in a Modelica library. In addition Modelica provides the possibility to create own connection definitions, with arbitrary information. The Meta-Model checks if the connection is applicable, if not, it manipulates it or raises an error message. Once all data from the SimXML is in the correct format, CoTeTo is able to export Modelica files. Modelica models are stored in simple text files written in the Modelica language. There are two general concepts for the generation of text within a computer program. One approach is to embed print()-statements for text strings and data in the structure of a program. This is useful for almost static, well-defined structures of the data set and of the textual output. The other approach is template-based, where placeholders for the content are embedded in a text file (a template for the output). Besides placeholders, templates also offer control structures. Thus, template-based model generation allows compliance with fixed Modelica language syntax and addition of flexible model content in the same file. With this approach, we are for example able to embed pre-defined control strategies for HVAC components in the output file. Another advantage is the flexibility for the end user, who does not necessarily needs to dive into the programs internal structure, but can just enrich the template file

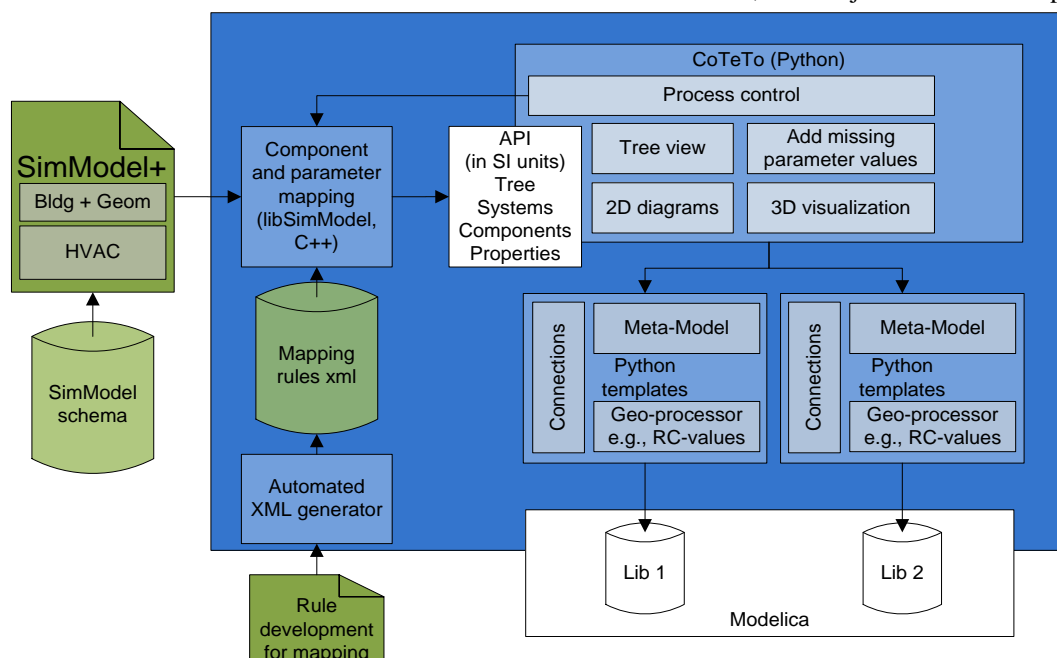


Figure 4: Detailed transformation from SimModel to Modelica models using the Code-Templating-Tool (CoTeTo)



with placeholders and simple programming constructs, whenever the used Modelica models change. This workflow is much like the form letter function in office software, which fills some variable address fields in a text document from a database.

The template approach fits very well into the flexible structure of CoTeTo, as it is independently usable for different information sources. From the list of available template engines, we chose the Mako software (Bayer 2015).

## WORKFLOW DEMONSTRATION

The following section provides an overview of a typical workflow using the open framework from BIM to Modelica. We applied the workflow to a simple Use Case using AixLib as target library. The general workflow is independent of the library used. The Use Case consists of one thermal zone, which suits the requirements of the validation example of German Guideline VDI 6007 (German Association of Engineers 2012). A simple HVAC example is provided. The hot water loop comprises:

1. Boiler;
2. Pump;
3. Controlling valve;
4. Radiator;
5. Flow and return pipes;
6. Expansion tank.

The first step of the process is the creation of a BIM representation in a suitable CAD-Software. This BIM model generates an IFC4 file that contains geometric and semantic information about building physics as well as HVAC installations. Although IFC4 includes HVAC installation, not all necessary parameters for the simulation are added in this step.

We convert the IFC file into a SimXML file. Using libSimModel and CoTeTo, the SimXML file is loaded and the mapping rules are applied. The library developer can adjust the mapping rules whenever needed. Besides the mapping of the parameter names, the following two examples show typical applications of the mapping rules:

- a) The Use Case contains no class to define an expansion vessel; this class with the corresponding parameters has to be added.
- b) The definition of the efficiency curve of the boiler is different in AixLib and SimModel, this parameter needs to be transformed.

Once the mapped data is in the Python environment, the user gets a feedback of warnings and errors to check if all parameter values are available and consistent.

The libSimModel API allows the addition of missing values into the information model. Meta-Model and Templates process the data automatically to generate a valid Modelica file. The user loads the file into the Modelica environment and checks the model for its consistency.

## LIMITATIONS

The authors are aware that the presented framework has some limitations. The conversion from BIM to a Modelica model is not fully automatic, nor a single-click solution. Different users need to work closely together to form a consistent tool chain. Some components of this tool chain need to be changed whenever the corresponding Modelica library changes. As all participating libraries in the Annex 60 project are under development, changes are not unusual and can take place regularly. The mapping rules and the templates are two examples of adaptable components. To adjust these parts the user needs a good knowledge of both the information model structure and the Modelica model. Further, the framework only currently supports a subset of Modelica models. The graphical representation and layout in the Modelica simulation environment is out of the scope of this project. The tool chain uses closed source and commercial products. For example Dymola is currently used for Modelica simulations.

## CONCLUSION

This paper presents an overview of an open framework of BIM-based Modelica BPS simulation. The work considers different aspects of the conversion from BIM to a valid simulation model. Because of the flexible approach of the framework, it is not limited to a specific Modelica library but adaptable to different Modelica libraries for Building Performance Simulation. To fulfil these requirements we developed several tools, which we connect to an integral tool chain. Starting from the well-established IFC data model the tool chain is able to transform a BIM to a Modelica model. To enhance the data model with HVAC systems and simulation related data we use an intermediate data model, called SimModel and the corresponding format SimXML. Through a generic API a Python tool can access the mapped data and create the Modelica model. This opens the framework to a large group of users. All tools developed in this project are or will be made open-source. We tested and validated the tool chain using a pre-defined Use Case. In addition to the tools, we specify a Model View Definition for BPS that can be used to specify the exchange requirements for dynamic building simulation in Modelica.

## ACKNOWLEDGEMENT

This work emerged from the Annex 60 project, an international project conducted under the umbrella of the International Energy Agency (IEA) within the Energy in Buildings and Communities (EBC) Programme. Annex 60 will develop and demonstrate new generation computational tools for building and community energy systems based on Modelica, Functional Mockup Interface and BIM standards.

Grateful acknowledgement is made for financial support by BMWi (German Federal Ministry of

Economic Affairs and Energy), promotional references 03ET1177A, 03ET1177D, 03ET1177E.

Other parts (no duplications) of the preliminary research work in Ireland were supported by a Marie Curie FP7 Integration Grant within the 7th European Union Framework Programme project title SuPerB, project number 631617.

## REFERENCES

- Bayer, Michael. 2015. "Mako Templates for Python." Accessed April 07, 2015. <http://www.makotemplates.org/>.
- Bazjanac, Vladimir, Tobias Maile, James O'Donnell, and Cody M. Rose. 2011. "Data Environments and Processing in Semi-Automated Simulation with EnergyPlus." In *CIB W078 - W102*. Vol. 2011. Accessed April 27, 2015.
- Cao, Jun, Tobias Maile, James O'Donnell, Reinhard Wimmer, and Christoph van Treeck. 2014. "Model Transformation From Simmodel to Modelica for Building Energy Performance Simulation." In IBPSA Germany, *BauSIM 2014*, 242–49.
- Cao, Jun, Reinhard Wimmer, Matthias Thorade, Tobias Maile, James O'Donnell, Jörg Rädler, Jérôme Frisch, and Christoph van Treeck. 2015. "A Flexible Model Transformation to Link BIM with Different Modelica Libraries for Building Energy Performance Simulation." *14th Conference of International Building Performance Simulation Association, Hyderabad, India*. (submitted).
- Constantin, Ana, Rita Streblov, and Dirk Müller. 2014. "The Modelica HouseModels Library: Presentation and Evaluation of a Room Model with the ASHRAE Standard 140." In *10th International MODELICA Conference, Lund, Sweden*, 293–99. doi:10.3384/ecp14096293.
- Eastman, Charles M. 2011. *BIM handbook: A guide to building information modeling for owners, managers, designers, engineers and contractors*. 2nd ed. Hoboken, NJ: Wiley.
- German Association of Engineers. 2012. "Calculation of transient thermal response of rooms and buildings - Modelling of rooms: VDI 6007-1." 91.120.10, 91.140.10 6007 - 1.
- Huber, Jörg, Christoph Nytsch-Geusen, and Tim Schünemann. 2012. "Automatische Generierung einfacher und detaillierter Gebäudemodelle aus 3D-Gebäudebeschreibungen für die Verwendung in Modelica-Systemmodellen." In *BauSIM 2012: Fourth German-Austrian IBPSA Conference*, edited by IBPSA Germany, 125–32.
- International Organization for Standardization. 2013. *Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries*, no. 16739:2013.
- Jeong, WoonSeong, Jong B. Kim, Mark J. Clayton, Jeff S. Haberl, and Wei Yan. 2015. "A framework to integrate object-oriented physical modelling with building information modelling for building thermal simulation." *Journal of Building Performance Simulation*, 1–20. doi:10.1080/19401493.2014.993709.
- Maile, Tobias, Phil Haves, and Richard See. 2015. "Integrating a Rule Based Code Compliance Software Platform Into a User-Oriented Simulation Interface." *14th Conference of International Building Performance Simulation Association, Hyderabad, India*. (submitted).
- O'Donnell, James, Richard See, Cody Rose, Tobias Maile, Vladimir Bazjanac, and Phil Haves. 2011. "SimModel: A Domain Data Model for Whole Building Energy Simulation." In *12th Conference of International Building Performance Simulation Association, Sydney, Australia*, 382–89.
- Rose, Cody M., and Vladimir Bazjanac. 2013. "An algorithm to generate space boundaries for building energy simulation." *Engineering with Computers* 31 (2): 271–80.
- Shafee, Ahamed, and John Dickinson. 2011. "Environmental Scan of BIM Tools and Standards."
- Wetter, Michael, and Christoph van Treeck. 2015. "IEA EBC Annex 60." Accessed April 07, 2015. <http://www.iea-annex60.org/>.
- Wetter, Michael, Wangda Zuo, Thierry S. Noudui, and Xiufeng Pang. 2014. "Modelica Buildings library." *Journal of Building Performance Simulation* 7 (4): 253–70.
- Wimmer, Reinhard, Jun Cao, Peter Remmen, Tobias Maile, James O'Donnell, Jérôme Frisch, Rita Streblov, Dirk Müller, and Christoph van Treeck. 2015. "Implementation of Advanced BIM-Based Mapping Rules For Automated Conversions to Modelica." *14th Conference of International Building Performance Simulation Association, Hyderabad, India*. (submitted).
- Wimmer, Reinhard, Tobias Maile, James O'Donnell, Jun Cao, and Christoph van Treeck. 2014. "Data-Requirements Specification to Support BIM-based HVAC-Definitions in Modelica." In IBPSA Germany, *BauSIM 2014*, 99–107.
- Yan, Wei, Mark J. Clayton, Jeff S. Haberl, WoonSeong Jeong, Jong B. Kim, Sandeep Kota, Alcocer, Jose Luis Bermudez, and Manish Dixit. 2013. "Interfacing BIM with Building Thermal and Daylighting Modeling." In *13th Conference of International Building Performance Simulation Association, Chambéry, France*, 3521–28.