INTEGRATING BUILDING INFORMATION MODELING WITH OBJECT-ORIENTED PHYSICAL MODELING FOR BUILDING THERMAL SIMULATION

A Dissertation

by

WOON SEONG JEONG

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Wei Yan
Committee Members,	Mark J. Clayton
	Jeff S. Haberl
	Yoonsuck Choe
Head of Department,	Ward V. Wells

December 2014

Major Subject: Architecture

ABSTRACT

This study presents a Building Information Modeling (BIM) to Building Energy Modeling (BEM) translation framework (*BIM2BEM*) through the integration of BIM with Object-Oriented Physical Modeling (OOPM) for building thermal simulation to support sustainable building design. A Model View Definition (MVD) is used as a data modeling methodology to assist *BIM2BEM*, and an application of *BIM2BEM* is demonstrated by visualizing energy performance in BIM.

The framework of *BIM2BEM* is made of a system interface between BIM and OOPM-based BEM (*ModelicaBEM*). The interface consists of the following two major phases: (1) pre-processing BIM models to add required thermal parameters into BIM and generate the building topology, and (2) translating BIM to *ModelicaBEM* automatically and running the thermal simulation. Finally, a case study was conducted to demonstrate and validate the simulation results.

The MVD enables efficient model translation consisting of a process model and a class diagram. The process model demonstrates the object-mapping process from BIM to *ModelicaBEM* and facilitates the definition of required information during the model translation process. The class diagram represents the object information and object relationships for producing the software tool for automatic model translation. In order to demonstrate and validate the approach, simulation result comparisons have been conducted for three test cases, each having two models: (1) the BIM-based Modelica model (*ModelicaBEM*) generated using the framework, and (2) the model manually

created using Lawrence Berkeley National Laboratory's Modelica Buildings library. The results show that the framework: (1) enables BIM models to be translated into *ModelicaBEM* models, (2) enables system interface development based on the MVD for thermal simulation, and (3) facilitates the reuse of the original BIM data for building energy simulation without an import/export process.

The visualization application enables visualizing building energy simulation results in BIM for designers to better understand the relationship between design decisions and the building performances. This new application lets architects use BIM as a common user interface for building design and performance visualization.

I dedicate this dissertation to my parents.

ACKNOWLEDGEMENTS

I would like to thank several people for their tremendous help and support throughout my Ph.D. study. First and foremost, I want to sincerely thank my advisor, Dr. Wei Yan. Without his feedback, support, guidance, and patience, I would not be able to begin and finish my Ph.D. study. This dissertation was initiated by the questions during the NSF PBIM project. While developing the framework described in this dissertation, there were a lot of intellectual challenges. His advice, insight, and constructive criticism regarding my research help me a lot in finding directions and possible answers. I am grateful to have had the chance to know Dr. Wei Yan and work with him.

Also, I want to thank my dissertation committee members for their service and helping me structure and further develop my dissertation. I am much honored to have very renowned scholars as my thesis committee members. Prof. Mark J. Clayton brought in his expertise in research design and was always available for discussing my research and providing feedback. Prof. Jeff S. Haberl, with his expertise in the PBIM project as well as building energy modeling, helps me by pointing out helpful resources. Prof. Yoonsuck Choe in the Department of Computer Science and Engineering broadened my view toward artificial intelligence through his lectures. His lectures on artificial intelligence inspired new ideas for the implementation of the algorithm required for my research. The support of my Ph.D. committee helped improve my research, and I feel very fortunate to have had the opportunity to work with Prof. Mark J. Clayton and Prof. Jeff S. Haberl with two projects: ASHRAE 1468 and NSF PBIM.

There are several people that I owe acknowledgements for giving me thoughtful and sincere comments to pursue my Ph.D. study. While I had been studying for my Master's degree, I had a fairly narrow research view on process and data modeling and construction management. I am very grateful to former colleagues at UIUC, Prof. Sangwon Han at the University of Seoul and Prof. JeongWook Son at the Ewha Womans University, Prof. Frank Boukamp at RMIT, and Ok Kyue Kim at the Chungbuk National University.

Many thanks also go to my colleagues and friends. I would like to thank Jang Hyun Kim, Dea Bock Kim, No Youn Park, Myung Joon Seo, Moon Jae Jeon and Won Il Boo for the professional and personal discussions and being wonderful friends. I also want to thank all of the members of the ASHRAE 1468 and PBIM research projects and the BIM SIM research group for good discussions. Especially, I want to thank Jong Bum Kim a lot in helping develop the interfacing framework.

Additionally, I would like to thank my family for their support. Without their sacrifice and support, I would not be able to complete my Ph.D. study. I am grateful to my parents, HeaIl Jung and OkRea Kim for all their endless support, love, and trust and for believing in me. I would like to express special words of thanks to my brother, Wun-Cheol Jeong, for his advice and my sister and brother-in-law, Ji-Sun Jeong and Hea-Young Lee, for their support, love, trust and everything. Also, I would like to thank all my relatives for staying in contact with me and giving their advice to me whenever I came to Korea.

Finally, thanks also go to everyone whom I have not mentioned here and who supports me through advice and/or friendship during my time as a Ph.D. student at Texas A&M University.

This research was funded by a grant from the National Science Foundation under Grant No. 0967446. NSF’s support is gratefully acknowledged. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

NOMENCLATURE

API	Application Programming Interface
CSV	Comma-Separated Value
EPI	Energy Performance Indicator
BEPV	Building Energy Performance Visualization
BEM	Building Energy Modeling
BES	Building Energy Simulation
BIM	Building Information Modeling / Model(s)
LBNL	Lawrence Berkeley National Laboratory
ModelicaBEM	Modelica-based BEM models translated from BIM models
ModelicaBIM	The developed library that contains BIM-based Modelica wrapper classes for LBNL Modelica Buildings library to support the creation of ModelicaBEM.
MVD	Model View Definition
OOP	Object-Oriented Programming
OOPM	Object-Oriented Physical Modeling
UML	Unified Modeling Language(s)
IDEF	Integrated Definition of Functional Modeling

TABLE OF CONTENTS

LIST OF FIGURES

Page

LIST OF TABLES

Page

1. INTRODUCTION

A lack of integration between building design and energy simulation prevents the efficient use of building performance analysis in the design process. The utilization of energy simulation tools to evaluate building performance can be critical at the design stage to for comparing design alternatives. The research team that I am a member of has investigated a modeling method to enable the integration of multi-domain simulations into design models using BIM and an OOPM approach: integrating physical simulation capabilities into BIM (*Physical BIM* or *PBIM*) (Yan et al. 2013; Jeong et al. 2013; Jeong et al. 2014a, 2014b; Kim et al. 2014; Kota et al. 2014).

The PBIM project is teamwork by Wei Yan, Mark J. Clayton, Jeff S. Haberl, WoonSeong Jeong, Jong Bum Kim, Sandeep Kota, Jose Luis Bermudez Alcocer, Manish Dixit, Mohammad Rahmani Asl, Mateo Aviles, and Alexander Stoupine. My Ph.D. dissertation research is one of the major components of the project and has close connection to other components of the project. To reflect the relationship between my dissertation research and other components of the project, in the dissertation, “I” and “the team” are used as task contributors. The differentiation between the use of “I” and “the team” indicates whether the described task has been conducted mainly by myself or the project team. In particular, I worked very closely with Jong Bum Kim in the PBIM project. Jong Bum has significant contributions on the development of PBIM library (Kim et al. 2014) and Modelica models for the validation of the library.

This study presents the framework of BIM to BEM (*BIM2BEM*) enabling the integration of BIM with OOPM for building thermal simulation to support sustainable building design (Jeong et al. 2014a). To assist the creation of the framework, I created a Model View Definition (MVD) using a data modeling methodology (Jeong et al. 2014b). The *BIM2BEM* framework and the MVD methodology are described below.

1.1. *BIM2BEM* Framework

This study presents the detailed *BIM2BEM* framework as a system interface for BIM-based building thermal simulation. The framework translates BIM to *ModelicaBEM*, which is a hybrid model between BIM and BEM.

The research objectives are: (1) to enhance the interoperability between BIM and BEM, (2) to enable more reliable BEM generation from BIM, (3) to inform sustainable building design with displayed simulation results in BIM, and (4) to enhance the usability of BIM as a common user interface to simulate energy performances. To achieve these objectives, the PBIM team created the *BIM2BEM* framework and I developed its prototype *Revit2Modelica* as a collection of computational tools to add additional parameters to BIM models, generate building topology, translate BIM models to *ModelicaBEM* program code, and run the simulation.

The *BIM2BEM* framework requires pre-processing BIM to include additional thermal parameters and generate building topology for *ModelicaBEM*. In order to create the *Revit2Modelica* prototype, I utilized the modeling capability of a BIM authoring tool (Revit) and the tool's API. The prototype can run thermal simulation automatically by: (1) translating Revit models to *ModelicaBEM*, (2) executing the simulation, and (3) displaying simulation results as data graphs in Revit. The prototype has utilized the

ModelicaBIM library that the PBIM research team created (Kim et al. 2014) and the Modelica Buildings library developed by LBNL (Wetter et al. 2014).

The following chapters explain details of the framework and the prototype, including research methods, implementation, results, case studies that validate the framework, and an application.

In this study, the Modelica-based BEM models translated from BIM models are called *ModelicaBEM,* which contains BIM information and can execute thermal simulation to obtain building performances such as indoor temperature, energy loads, and so on. The terms “Building Information Modeling” or “Building Information Model(s)” are used interchangeably in different contexts and are abbreviated as BIM.

1.2. Data Modeling Using MVD

The exchange of data between building design models and energy simulation models has been a major challenge in the design process, resulting in the fact that building energy performance simulation is often omitted from the process (Bazjanac 2008). The translation process is labor intensive, error-prone, and cumbersome (Bazjanac 2008, 2009; Bazjanac et al. 2011; Hand et al. 2005). Although tools have been developed to support the generation of an energy model from a design model, disconnects still exist between the various models (Bazjanac 2008; General Services Administration 2014; Rose and Bazjanac 2013; O’Donnell et al. 2011). I hypothesize that the problem can be alleviated by taking the advantage of Object-Oriented Modeling methods existing in both design and simulation models. To improve and enhance the model translation effectiveness, the PBIM project team investigated a new approach to link Building Information Modeling (BIM), which is commonly used to support architectural design, to Building Energy Modeling (BEM) that supports energy simulation. I used C# programming to directly access the Object-Oriented data representation within a BIM authoring system, and Modelica, an Object-Oriented Physical Modeling language, to generate energy models and simulate the energy performance. The hypothesis is that use of Object-Oriented constructs within both BIM and BEM will enable more efficient and reliable translation and improve maintainability.

This research has employed data modeling methods to assist *BIM2BEM* framework connecting Autodesk Revit to the LBNL Modelica Buildings library.

The research creates a Model View Definition (MVD) that consists of a process model and a class diagram and then conduct testing to assure that the translation works properly. The following four phases have been conducted for the MVD development:

(1) Develop a process model to document the mapping from BIM to BEM.

(2) Develop class diagrams to represent the required information and object relationships.

(3) Implement the translation classes to support *ModelicaBEM* creation using BIM data.

(4) Conduct tests to demonstrate and validate the *BIM2BEM* framework.

The objectives of using *MVD* are to: (1) enable BIM models to be translated into *ModelicaBEM* models, (2) enable system interface development based on the MVD for thermal simulation, and (3) facilitate the reuse of the original BIM data in building energy simulation without an import/export process.

The research scope is confined to translating the building envelope information of BIM, including geometry, material, and topology of a building model.

2. BACKGROUND

The U.S. DOE. (2014) has been providing information on over four hundred software tools for building energy simulation. Among the tools, a few are widely utilized in education and industry (Maile, Fischer, and Bazjanac 2007; Aksamija 2012; Attia 2011; Crawley et al. 2008). Ecotect, Energy-10, Radiance, and eQUEST, among the tools listed by the U.S. DOE, are widely used in education (Haberl 2008), while DOE-2, eQUEST, Ecotect, EnergyPlus, Energy-10, Green Building Studio, HEED, and IESVE are widely utilized in industry (Attia et al. 2009).

Recently, one emerging research area is BIM-based energy performance analysis. Some existing tools have been modified for using BIM and new simulation tools have been developed. For better integration between BIM and energy simulation, the research on standard data schemas or common data structures such as the Industry Foundation Classes (IFC) and Green Building eXtensible Markup Language (gbXML) has been conducted (Bazjanac and Maile 2004; Bazjanac 2001, 2008; O'Donnell et al. 2011).

In the meanwhile, studies have presented the potentials of an Object-Oriented Physical Modeling (OOPM) approach for building energy simulation. In an effort to support building energy simulation, a series of energy simulation libraries have been created by using OOPM languages such as Modelica (Wetter 2009).

In this chapter, I reviewed literatures for (1) a brief history of the conventional energy simulation tools, (2) recent studies on BIM-based energy simulation, (3) the new approach of OOPM-based energy simulation, and (4) problems in data translation between BIM and BEM.

2.1. Existing Building Energy Simulation Tools

Existing building energy simulation tools aim to evaluate energy performance and thermal comfort during a building's life cycle. Specifically, DOE-2 and EnergyPlus have been widely used at multiple stages of a building's life cycle due to their functionality of exchanging data with other tools through standard data formats such as IFC (Maile et al. 2007). For over thirty-five years building scientists, mechanical engineers, and computer programmers have developed sophisticated simulation tools that consist of simulation engines and graphical user interfaces (GUIs) (Clarke 2001; Crawley et al. 2008). As described in the "Building Energy Software Tools Directory" (U.S. DOE 2014), the tools differ in terms of input, output, GUIs, programming languages, and computer platforms. For example, the simulation engines in the traditional building simulation programs, including: DOE-2, ESP-r and EnergyPlus (Winkelmann et al. 1993; Clarke 2001; Crawley et al. 2001) have been developed using computer languages such as FORTRAN, C and C++ to describe the physical processes that they simulate (Wetter 2009). DOE-2.1e and DOE-2.2 utilizes a Building Design Language (BDL) Processor for translation of the user input, and DesignBuilder provides a GUI to EnergyPlus (DesignBuilder Software 2014; Good et al. 2008). In case of IESVE, SunCast (SunCast 2014) and ApacheSim (ApacheSim 2014) are used for thermal analysis (IESVE 2014).

While the simulation engines and GUIs are developed separately, they reduce the amount of time needed for manual data entry and enable designers to test the building performance of alternative designs (Crawley et al. 2008). Currently, building performance simulation (BPS) professionals widely use dedicated GUIs such as eQUEST and IESVE for building performance simulations, which facilitate input generation and output analysis.

However, architects and designers are now required to spend significantly more effort acquiring simulation knowledge to better understand thermal processes to evaluate energy performance of their designed buildings (Attia 2011; Maile et al. 2007). In addition, while the GUIs are considered dedicated tools to BEM professionals, architects and designers have difficulty using BEM tools because of the incompatibility with their individual workflows as well as the complexity of using the BEM tools (Gratia and Herde 2002; Punjabi and Miranda 2005). To provide integrated building performance analyses with the design process, BIM-based analysis tools have started to emerge (Aksamija 2010). Compared to the existing dedicated GUIs, BIM-based simulation tools can directly utilize the building design data that have already been created by architects in BIM, which eliminates the need to re-enter the building data into dedicated BEM GUIs. Considering that parametric studies for optimizing building performance are fast becoming a trend in the design of sustainable buildings, this integrated BIM-based BEM will be in high demand by the design community.

In order to support a more seamless integration with a design phase, BIM-based BEM tool development would require providing an easy-to-use user interface, an efficient and effective BIM-to-BEM system interface, and more intuitive simulation result visualization.

2.2. Building Information Modeling for Building Energy Simulation

As an emerging technology in the Architecture, Engineering, Construction and Operation (AECO) industry (Eastman et al. 2008), BIM has been supported by the contemporary authoring tools including Revit Architecture (Autodesk Inc. 2014a), AECOsim Building Designer V8i (Bentley System 2014), and ArchiCAD (GRAPHISOFT 2014), among others. BIM assists a design process in the following ways:

- BIM facilitates the access of comprehensive building data including building components and their properties during a building's life cycle (Eastman et al. 2008).
- BIM's parametric design capability enables architects to update their design quickly, interactively, and in real time during the design process (Lee, Sacks, and Eastman 2006).

BIM's capability to support the design process, thus changing the building industry and architecture education is promising (Eastman et al. 2008). However, a lack of essential physical properties, the limitation of modifying the existing properties in BIM authoring tools to apply real data, and the different object semantics between BIM and energy simulation prevent BIM's direct use in building simulation. For example, in Revit 2013, some material parameters such as solar and infrared absorptivities are missing while they are required for building energy simulation.

In addition, a certain level of data translation is required for BIM data to properly represent the thermal envelope and decisions about how to zone a building is often left up to the users (Bazjanac and Kiviniemi 2007). For instance, if an interior wall is added to separate one room into two rooms, the wall should be translated into thermal boundary conditions. Also, while a window frame consists of components such as seal, gasket, rubber, etc., in a detailed BIM model, BEM only needs to retrieve the necessary frame area data and the thermal conductivity perpendicular to the glass and frame. The new system presented in this work is developed and demonstrates a process of these and other sample cases.

While a limited number of building energy simulation tools are integrated with BIM through IFC and gbXML, additional efforts such as manual model check and modification for defining thermal zones are requested to generate reliable energy models (General Services Administration 2014). For instance, DOE2.2-based Green Building Studio provides web-based energy analysis working with Revit, but its users must complete simulation settings in order to create a reliable gbXML file. Furthermore, while some existing BIM-based energy simulation tools support the energy model generation processes, incorporating simulation results into the design decision is still considered difficult (Bazjanac 2008). Typically, energy simulation tools provide a large quantity of numerical results presented in tables or charts. The results do not allow visualization of the relationships between the performance and the designed building objects, while visualization should be enabled to indicate which building objects are critical to building energy performance. Consequently, designers may not be informed about how to proceed with design changes.

The BIM-based approach relying on the use of procedural language, e.g. Fortran-based BEM tools converts building objects to non-object-oriented models to run the simulation. Compared to these existing BIM-based approaches, the present approach utilizes a new OOPM simulation engine that can be more naturally integrated with object-based BIM, which can be used as the simulation input. This way, object (or component)-level thermal performance results can be the output, which will be discussed in the following chapters.

2.3. Object-Oriented Physical Modeling for Sustainable Design and Building Energy Simulation

In order to facilitate the efficient use of multi-domain simulations, Object-Oriented Modeling approaches have been developed (Wetter 2009). OOPM is a fast-growing research area providing structured and equation-based modeling techniques (Fritzson 2010). Modelica is a relatively new, unified OOPM language that facilitates the modeling of dynamic behaviors using differential algebraic equation (DAE)-based calculations (Fritzson 2010).

Existing Modelica authoring tools include Dymola (Dassault Systèmes 2014) and OpenModelica (by PELab, Linköping University), among others. The tools facilitate the design of complex physical systems for mechanic, electric, thermal, control, and other processes. The topology of a component connection diagram in Modelica directly corresponds to the structure and decomposition of physical system topology of energy models (Fritzson and Bunus 2002; Fritzson 2010). The capability of topology creation facilitates a natural mapping from object-based architectural modeling (BIM) to OOPM-based energy modeling.

The Modelica-based OOPM approach aims to support virtual prototyping for the rapid evaluation of different concepts based on quickly adding models or extracting subsystem models in the building energy systems (Wetter 2009). Specifically, LBNL Modelica Buildings library is a Modelica-based building thermal modeling and simulation system (Wetter 2009), which is validated (Nouidui, Wetter, and Zuo 2012) and under continued development. The library consists of dynamic models and control systems for object-based building energy simulation and supports thermal simulations including heating and cooling systems, controls, and heat transfer through the building envelopes (Wetter 2009). Because of the OOPM methods employed in LBNL's Modelica Buildings library and its potential to benefit interoperability between BEM and BIM (which is also object-based modeling), the PBIM research team decided to investigate the use of this library as the thermal simulation engine in the research.

However, with the existing LBNL Modelica Buildings library, a user needs to write programming code to construct building models and run the simulation in Modelica. This will require the designers to acquire Modelica programming knowledge and experience, which are beyond the capability or interest of most designers.

BIM-based *ModelicaBEM*, which is the focus of this research, will overcome this problem by automating the energy model generation process based on complete BIM models, while the BIM model creation process often involves manual and interactive use input.

2.4. Problems in the Data Translation Process between BIM and BEM

Studies have presented the value of reusing data that has been produced by building designers when creating building energy models (Bazjanac 2008, 2009; Rose and Bazjanac 2013; O'Donnell et al. 2011; Bazjanac and Maile 2004). To increase the usability of data from designers in building energy simulation, various research prototypes (Bazjanac 2008, 2009) and commercial products have been created, such as Green Building Studio.

However, reliably generating high quality BEM using current tools remains difficult. Although much of the process has been automated, intervention by the user to simplify models, choose among representations with subtle differences, and correct errors. For example, the users of Green Building Studio, which is a web-based energy analysis tool working with Revit, must finish the model check process to create a reliable gbXML file. Current energy simulation engines have their own unique input formats consisting of non-object-oriented text files with highly specialized syntax and semantics (O'Donnell et al. 2011). The different data structures between BIM and an energy simulation engine often prevents efficient data translation or exchange. For instance, an existing translation process is required to perform data exchange through standard data schemas, which hinders the utilization of the parametric modeling capability of BIM in the design process. While a limited number of energy simulation tools support standard schema-based model translation, the absence of a standard interface in the tools also requires additional efforts and understanding of simulation processes for architects and designers to obtain building analysis results (O'Donnell et al. 2011; Attia 2011; Maile, Fischer, and Bazjanac 2007).

The efficient and effective data translation between BIM and BEM may be achieved when two domains have the same modeling method such as an Object-Oriented Modeling method. In addition, a comprehensive data exchange model can support direct mapping between them and facilitate an easy-to-use user interface implementation.

Based on the development of an interdisciplinary data exchange model and implementation of the model for direct mapping without an import/export process, *BIM2BEM* can facilitate the reuse of data from BIM in building energy simulation.

3. METHODOLOGIES

The methodologies of the research include creating MVD for model translation between BIM and *ModelicaBEM* and creating a system interface for the translation. A prototype of the *ModelicaBEM* will be created and tested for validation.

MVD usually defines the subset of IFC models for supporting data interoperability (IAI 2014). I adopted the concept of MVD to reduce the interoperability problem and support more seamless translation between BIM (Revit) and *ModelicaBEM*.

3.1. Creating the MVD for Model Translation between BIM and *ModelicaBEM*

This section describes challenges and tasks, tools and data, and methodology for the MVD development. The MVD is intended to handle the translation from a BIM to BEM represented in Modelica to facilitate executing thermal simulation.

3.1.1. Challenges and Tasks

The main challenge of developing the MVD is how to facilitate seamless model translation, requiring less manual data conversions between BIM and *ModelicaBEM*. To achieve effective and efficient model translation, the following tasks need to be completed: (1) defining an object mapping process between BIM and *ModelicaBEM* to identify required information, (2) representing the identified datasets for the object mapping process, and (3) implementing the represented datasets and object relationships in the Modelica-based simulation tool.

Define an Object Mapping Process between BIM and *ModelicaBEM*

Object semantics and relationships in architectural models are often represented differently than in the energy models. For example, the energy modeling abstracts building components as 2D surfaces in order to enhance simulation performances, while the components are presented as 3D geometry in BIM.

In addition, connectivity of building components or the model topology is an important aspect in energy modeling. To meet the requirement of creating the BEM topology, the building topology from BIM needs to be mapped into *ModelicaBEM* topology.

To facilitate consistent object semantics and relationships between BIM and *ModelicaBEM*, an object mapping process needs to be conducted. The object mapping process demands to identify what information BIM and *ModelicaBEM* should be able to exchange. I utilized a data model method to classify mismatched object semantics and behaviors for the object mapping. The data modeling enables to maintain consistent object classifications of building components from BIM to *ModelicaBEM*.

Represent Datasets for the Object Mapping Process

Different object semantics and relationships between BIM and *ModelicaBEM* demand their own data structure. For instance, data for building components such as walls, floors, roofs, etc. are represented as 3D solids in BIM, whereas the same data are considered as surfaces in *ModelicaBEM*. In addition, a room object in BIM is represented as a zone in BEM, and the topology information for boundary condition is only represented in BEM, which can be retrieved by the connectivity of building objects information from BIM. To map the mismatched objects and behaviors, a data representation process is needed regarding what datasets in BIM and *ModelicaBEM* are used.

Implement the Datasets and Object Relationships

The datasets and object relationships need to be created in Modelica using parameters and functions. Instantiated objects can present building and related energy components in *ModelicaBEM*. For example, the area parameter can represent diverse geometry instead of just rectangular shape. Building topology in BIM can be mapped into *ModelicaBEM* topology, and calculated area information from BIM can be stored in a parameter.

3.1.2. Tools

For the MVD development, I used the BIM authoring tool Revit and its API, Modelica, and the LBNL Modelica Buildings library (Wetter et al. 2014).

BIM Authoring Tool (Revit) and Its API

BIM supports 3-Dimensional, semantically rich, and parametric modeling for design and construction during a building's life cycle (Eastman et al. 2008; Lee, Sacks, and Eastman 2006). BIM tools enable such capabilities through their own data structures and implement the structures using specific database schema (Eastman et al. 2008; Lee, Sacks, and Eastman 2006; Sacks, Eastman, and Lee 2004). The BIM tools allow the databases to be represented as standard data models such as IFC, which is a standard data schema for exchanging data among different applications, through user commands or API (Autodesk Inc. 2014a; GRAPHISOFT 2014; Bentley System 2014). In Revit, software developers can access specific building component data and a comprehensive database through API using the C# language (Autodesk Inc. 2014b). In this study, instead of using standard data models such as IFC or gbXML, I utilized the Revit API capability to access the BIM data directly to (1) preserve object relationships established by parametric modeling, (2) define a model view of Revit to support bi-directional data exchange with LBNL Modelica Buildings library.

Modelica and Dymola

To support modeling and simulation, OOPM has been developed to offer a structured and equation-based modeling approach (Tiller 2001; Fritzson 2010). Modelica is an OOPM language and enables users to simulate energy systems by modeling the energy objects and the object connections. Such capabilities can facilitate an object mapping from the BIM structure to *ModelicaBEM* naturally. Modelica libraries such as LBNL Modelica Buildings library (Wetter et al. 2014) facilitate the use of Modelica in thermal simulation, offering model components and solvers. I used Dymola as an integrated simulation environment for Modelica models with LBNL Modelica Buildings library as the thermal simulation engine.

LBNL Modelica Buildings Library

The LBNL Modelica Buildings library has been developed for building energy simulations to support the simulation of heating and cooling systems, controls, heat transfer through building envelopes, and airflow (Wetter 2009). One of the major resources for building thermal analysis in the library is the HeatTransfer and Room packages, which have been validated through benchmarked simulation models (Nouidui, Wetter, and Zuo 2012; Wetter and Nouidui 2011). The validation accounts for the capability of whole building simulations (Nouidui, Wetter, and Zuo 2012).

In order to create *ModelicaBEM* that can use the LBNL Modelica Buildings library for building thermal simulation, Modelica codes must be created based on BIM data. Insufficient data exchange capability between BIM and the library results in the designers' subjective interpretations of building data and human errors in creating *ModelicaBEM*. In addition, the absence of a de facto standard interface for the data exchange causes a difficulty in translating BIM into *ModelicaBEM* incorporating with the library. *BIM2BEM* facilitates the data exchange through the model view of the library and the intermediate classes.

Method and Tasks

The method in the MVD development includes (1) developing a MVD through data modeling, (2) implementing the designed classes in the MVD using the Modelica and the C# languages, and (3) conducting test cases for validation by simulation result comparisons for BIM models.

MVD Development

I utilized a data modeling method to develop a MVD for data exchange between BIM and *ModelicaBEM*. The MVD consists of (1) modeling the process to map objects including mismatched objects' semantics and behaviors and (2) designing classes to represent the required information and object relationships.

Process Modeling

The process in this research is the mapping from BIM to BEM. I used process modeling to identify required information and object relationships. Although some data can be easily translated, the challenges arise from recognizing mismatched object semantics and behaviors between the BIM and the BEM. Resolving these mismatches was a major task in this research.

Semantic mismatches hamper data exchange of objects and parameters because the BIM and BEM make use of fundamentally different abstractions. For the example, BIM represents a building envelop by composing building components such as walls, floors, and roofs, while BEM represents the envelope as exterior and interior surfaces. In order to map the building components into exterior surfaces, the required information can include the area of the surfaces and the summation of them through a function. To implement the function, the object relationships between related building components need to be defined to inform what kind of and how many surfaces constitute the whole exterior surface.

Behavior mismatches occur when the objects are similar or identical in BIM and BEM, but the behavior of the object is different. The required information must be derived by applying a rule that accepts the BIM information as input and produces the BEM information as output. For instance, when a user separates two rooms by using an interior wall in BIM, BEM defines the thermal boundary conditions to facilitate heat transfer between the separated rooms. A rule is: if one surface of the wall object in BIM is defined as a *surface boundary*, the other surface can be a *construction boundary* automatically to map the boundary condition into BEM. BEM defines the two boundary conditions to calculate heat transfer on interior walls between thermal zones. I can apply the rule in generating *ModelicaBEM*'s building topology. The boundary condition information can be obtained by implementing the rule using a room object and building components enclosing the room such as walls, floors, and roofs.

The process modeling method involves decomposing the process into a series of activities, connecting them into a logical sequence, and collecting the data requirements (Elmasri and Navathe 2010; WFMC 1999). I used process modeling to identify the required information and object relationships to support efficient workflows. The process model can be used in defining the scope of data modeling.

There are several graphic and non-graphic methods for process modeling, such as the Flowchart, Unified Modeling Language(s) (UML), and IDEF0 (Lee, Sacks, and Eastman 2007). IDEF0 (Integrated Definition of Functional Modeling) is most commonly used in product data modeling (ISO TC 184/SC 4 1994). I used IDEF0 to describe how activities for the mapping are connected, ordered, and structured. The unique feature of IDEF0 is its ICOM codes (Input, Control, Output, and Mechanism presented by arrows): Input and output arrows represent the data and object flows into and out of a function; Control arrows indicate the required conditions for a function; and Mechanism arrows denote the means to performance a function (NIST 1993). IDEF0 models are especially useful in understanding a data flow (Lee, Eastman, and Sacks

2007). I created an IDEF0 diagram for the process model, and then define what information is needed to map data between BIM and *ModelicaBEM*. The information will be represented through a class diagram including attributes and class relationships. Chapter 4 Developments explains how requirements for object mapping can be represented using IDEF0 specifications.

Class Design

I developed a class diagram to represent specific data types and object relationships as objects and relationships. Based on the investigation of the mapping process for the required information and object relationships, I created two model views to define datasets: Revit Model View and Modelica Model View. Based on the two model views, I created an intermediate class package consisting of wrapper classes and interface classes as an Exchange Model View.

The class diagram enables *ModelicaBEM* not only to follow the data structure and semantics of BIM but also to represent related information for thermal simulation. The next chapter describes how to create the class diagram using UML.

Implementation

I used the C# language to implement the functions in the interface classes, which facilitate data translation such as building topology translation.

Wrapper classes are created in Modelica to bridge between the Revit BIM classes and the energy model classes in the LBNL Modelica Buildings library. The wrapper classes enable *ModelicaBEM* to populate instantiated objects. Consequently, a *ModelicaBEM* is able to represent mismatched semantics and behaviors by composing related instances and parameters that store the values from BIM. *ModelicaBEM* relies on a system interface that can pre-process BIM to prepare the required information and assemble the instantiated objects before the *ModelicaBEM* reaches the LBNL Modelica Buildings library. The interface classes enable the prototype *Revit2Modelica* to pre-processing BIM.

Conducting Test Cases and Simulation Result Comparisons

Three test cases have been studied to demonstrate and validate the MVD development. For demonstration, a prototype *Revit2Modelica* is created to shows how BIM models can be automatically translated into *ModelicaBEM*. In case of validation, the simulation result comparisons are conducted between two BEM models in each test case: one is automatically generated *ModelicaBEM* and the other is the manually created model following LBNL's BEM structure.

3.2. Creating a System Interface for the *BIM2BEM* Framework

This section describes challenges, objectives, tools, data, and methods in the system interface development.

3.2.1. Challenges and Tasks

Below I discuss the challenges and tasks of mapping objects, adding new parameters, creating *ModelicaBEM*, running simulations, and displaying simulation results.

Define Additional Parameters in BIM

BIM contains general material properties including some thermal parameters. However, BIM misses some important thermal parameters such as solar and infrared absorptivities and the frame ratio of a glazing system that are used in the LBNL Modelica Buildings library. This type of new parameters needs to be defined in BIM.

Create *ModelicaBEM*

The distinct object semantics between BIM and the LBNL Modelica Buildings library require me to establish a set of steps in order to automatically create *ModelicaBEM* as follows.

- Building components in BIM such as roofs, walls, and floors will be transferred into BEM objects.
- Room objects in BIM can be transferred into thermal zones in *ModelicaBEM*.
- The topology of BIM will be obtained and translated into object connectivity in *ModelicaBEM*.

These steps enable *BIM2BEM* to (1) translate each BIM object into *ModelicaBEM* objects and (2) translate BIM topology into *ModelicaBEM* topology.

Run Simulations

ModelicaBEM is made of Modelica programming code. The Integrated Development Environment (IDE) - Dymola is used to execute the code for thermal simulation. The developed prototype (*Revit2Modelica*) enables *BIM2BEM* to call Dymola from the BIM interface in order to execute the thermal simulation automatically.

Once the thermal simulation is complete, *BIM2BEM* calls Dymola to produce object-based simulation results, e.g. heat flow of each building element, room temperature, and annual heating and cooling loads.

Display Simulation Results in BIM

The object-based simulation results are presented as a series of data graphs in Dymola. To support designers' decision-making, displaying the simulation results in BIM is needed and implemented in *Revit2Modelica*.

3.2.2. Data

***ModelicaBIM* Library**

The research team of PBIM created a *ModelicaBIM* library for mapping BIM to the LBNL Modelica Buildings library that can run in Dymola (Kim et al. 2014).

ModelicaBIM library consists of a *ModelicaBIM* package and a *ModelicaBIM* structure. The *ModelicaBIM* package includes Modelica wrapper classes of building component classes in the LBNL Modelica Buildings library. For example, the research team utilized the *MixedAir* class in the LBNL Modelica Buildings library, which can model thermal zones with completely mixed air, to create the wrapper class - Room. The Modelica wrapper classes use similar object semantics and structures as in BIM to facilitate the object mapping. The *ModelicaBIM* structure composes of five example building models referencing the *ModelicaBIM* package. The example models provide a series of templates to construct *ModelicaBEM*. The *Revit2Modelica* prototype demonstrates the process of automatic creation of *ModelicaBEM* that follows the

ModelicaBIM structure and calls the LBNL Modelica Buildings library for thermal simulation.

3.2.3. Methods

The methods in the *BIM2BEM* interface development include: (1) pre-processing BIM models to create extended BIM, (2) developing the *Revit2Modelica* prototype by creating system and user interfaces, and (3) conducting a case study with a benchmark energy model for validation.

Pre-processing BIM

I designed a series of pre-processes: adding, translation, and calculation of parameters with two specific commands including *AddingParameter* and *GenerateBuildingTopology*. The adding process is to add required information for object mapping and physical modeling that is not defined in existing BIM or need to be modified. The translation process is to translate information between BIM and LBNL Modelica Buildings library, for example, the room-to-thermal zone translation. The translated information will be embedded into the *ModelicaBEM* topology generated by the *GenerateBuildingTopology* command. The calculation process is to compute new values using existing or added values in BIM, for example, the window-frame ratio.

Developing *Revit2Modelica* Prototype

The *Revit2Modelica* prototype can demonstrate (1) the BIM to *ModelicaBEM* mapping, (2) execution of thermal simulation, and (3) displaying simulation results in BIM. The prototype consists of a system and user interface developed using Revit's API commands: *AddingParameter* and *GenerateBuildingTopology*. The system interface implements the process of automatic translating and simulating *ModelicaBEM*, and the user interface enables designers to check and save simulation results, which allows users to compare a series of simulation results easily. As an example, a sample building consists of four exterior walls, a floor, a roof, and a room. The walls attached to the room are considered as exterior walls in BIM, but they need to be recognized as construction boundaries in *ModelicaBEM*. The prototype will automatically map the walls properties into the energy properties with relevant boundary conditions to create the *ModelicaBEM*.

Conducting a Case Study

A case study to demonstrate and validate *BIM2BEM* has been conducted by using a benchmark model (BESTEST Case 600) and comparing simulation results for heating and cooling loads.

4. DEVELOPMENTS

This chapter describes the developments of MVD for model translation and the system interface for the *BIM2BEM* framework.

4.1. MVD for Model Translation

BIM and LBNL's BEM (manually created using Modelica) follow Object-Oriented Modeling concepts; however, they have different object semantics and behaviors, which are challenging for model translation. In this study, I developed a MVD to define data exchange requirements for Revit and the LBNL Modelica Buildings library.

MVD usually defines the subset of IFC models for supporting data interoperability (IAI 2014). We adopted the concept of MVD to reduce the interoperability problem and support more seamless translation between BIM (Revit) and *ModelicaBEM*. The MVD development follows a Modeling-Diagramming-Implementing approach: (1) developing a process model to identify building objects and their relationships, (2) creating a class diagram based on application model views and the exchange model view, and (3) implementing the intermediate class package.

4.1.1. Develop a Process Model for Translation

To identify mismatched objects semantics and behaviors, the research team studied the translation process between BIM and *ModelicaBEM*. Figure 1 shows the overall process of how BEM is incorporated with BIM: data from BIM are translated into *ModelicaBEM*, *ModelicaBEM* produces object-based results after completing the simulation, and the results are able to be displayed in BIM finally (Jeong et al. 2013).

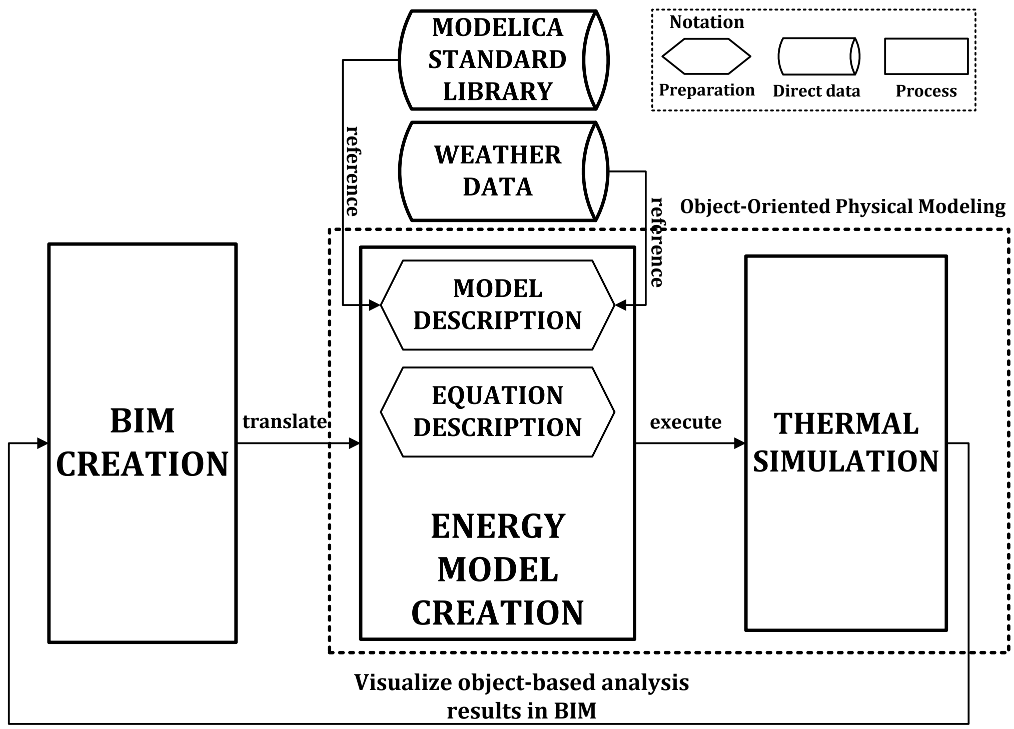


Figure 1. The overall translation process between BIM and *ModelicaBEM*.

The data translation process mainly occurs between BIM Creation and Energy Model Creation shown in Figure 1. The model description preparation in the Energy Model Creation activity is to populate required information from the Modelica standard library and the weather data. The major classes and parameters in LBNL Modelica Buildings library are shown in Table 1. The boundary condition definition in the library has five types: exterior opaque surfaces (datConExt), exterior opaque surfaces with windows (datConExtWin), interior walls between thermal zones (datConBou or surBou), and interior partitions in a thermal zone (datConPar). I applied the translation rules (in Section 4.1.3) for translating the building topology into the boundary conditions.

Table 1. Class and parameters adapted from LBNL Modelica Buildings library (Wetter et al. 2014).

Classes	Object Properties	
	Name	**Description**
MixedAir models a room with completely mixed air for heat transfer through a building envelop. The room consists of any number of construction types and surfaces for heat exchange through convection, conduction, and infrared radiation and solar radiation.	medium	The medium information of a room air such as gas, most air, dry air, and so on.
	aFlo	The floor area attached to a room.
	hRoo	The roof area attached to a room.
	datConExt	Opaque surfaces.
	nConExt	Number of datConExt.
	datConExtWin	Opaque surfaces with windows.
	nConExtWin	Number of datConExtWin.
	datConPar	Interior partitions in a thermal zone.
	nConPar	Number of datConPar.
	datConBou	Opaque surfaces on interior walls between thermal zones.
	nConBou	Number of datConBou.
	surBou	Opaque surfaces on the same interior walls between thermal zones.

Table 1. Continued.

Classes	Object Properties	
	Name	**Description**
	nSurBou	Number of SurBou.
	nPorts	Number of ports that constructs equations to simulate physical processes.
	latitude	Latitude information of a room.
	energyDynamics	The information of fluid types in networks of vessels, pipes, fluid machines, vales, and fittings.
	linearizaRadiation	A setting value whether to linearizes emissive power or not.
OpaqueConstructions describes material definitions for constructions with one or more layers of material.	matLayExt	Construction material for exterior walls.
GlazingSystem describes thermal properties for glazing systems.	nLay	Number of glass layers.
	haveExteriorShade	A setting value whether a window has an exterior shade or not.
	haveInteriorShade	A setting value whether a window has an interior shade or not.
	glass	Thermo physical properties for window glass.
	gas	Thermo physical properties for window gas fills.
	uFra	U-value of frame.
	absIRFra	Infrared absorptivity of window frame.
	absSolFra	Solar absorptivity of window frame.
DoorDiscretizedOpen describes the bi-directional airflow through an open door.	medium	The medium information of the room airflow through an open door.
	width	Width of opening
	height	Height of opening

The building can be represented with instances from the classes in Modelica using LBNL's library. As shown in Figure 2, a thermal zone consisting of six surfaces without any openings can be declared as a thermal zone instance (line 1) and surfaces information is given parameters of the zone instance. Six surfaces of the thermal zone are categorized as opaque surfaces (line 5), and their layer information (line 7), area (line 8), a tilt angle (line 9), and an azimuth angle (line 10) are provided.

```
1 Buildings.Rooms.MixedAir mixedAir(
2   redeclare package Medium = MediumA,
3   AFlo=32,
4   hRoo=3,
5   nConExt=6,
6   datConExt(
7     layers={matLayRoo,matLayFlo,...},
8     A={32,32,...},
9     til={Buildings.HeatTransfer.Types.Tilt.Ceiling,Buildings.HeatTransfer.Types.Tilt.Floor,...},
10    azi={Buildings.HeatTransfer.Types.Azimuth.S,Buildings.HeatTransfer.Types.Azimuth.N,...}),
11  nConExtWin=0,
12  nConPar=0,
13  nConBou=0,
14  nSurBou=0,
15  linearizeRadiation=false,
16  energyDynamics=Modelica.Fluid.Types.Dynamics.FixedInitial,
17  nPorts=1,
18  lat=0.73268921998722)
```

Figure 2. A code block for thermal zone modeling in LBNL Modelica Buildings library.

LBNL Modelica Buildings library is developed based on engineering-semantic point of view. As shown in Figure 2, the thermal zone instance does not require any wall instances to simulate a room model even though the building consists of several walls. Such mismatched object semantics (compared to BIM) require us to define an object semantics rule set to represent information in BIM for the BIM-to-*ModelicaBEM* translation. Based on the investigation and the guideline for object mapping (Clayton et al. 2013), the research team sets the rule set as follows:

- Addition: adding missing data in BIM that are required for *ModelicaBEM*, such as solar and infrared absorptivities, solar transmittance, infrared transmissivity of glass.
- Translation: data translation between BIM and *ModelicaBEM* to represent mismatched semantics, such as rooms to thermal zones.
- Calculation: calculating new values using existing values in BIM, such as the window-frame ratio and construction boundary types.

I can identify required processes applying the rule set to *ModelicaBEM* creation and represent the BIM-to-*ModelicaBEM* translation using IDEF0 as shown in Figure 3.

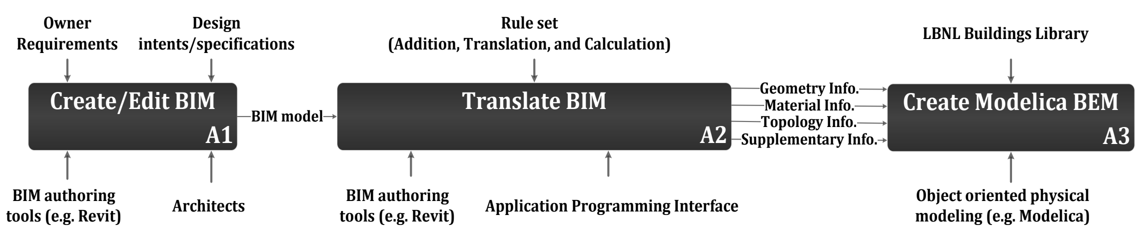


Figure 3. A process model of BIM-to-*ModelicaBEM* translation using IDEF0.

However, the Translate BIM activity only represents object semantics. As described in the Methodologies chapter, the LBNL Modelica Buildings library also represents object behaviors including building topology differently. To represent such mismatched behaviors, I identify detailed processes of the Translate BIM activity as shown in Figure 4.

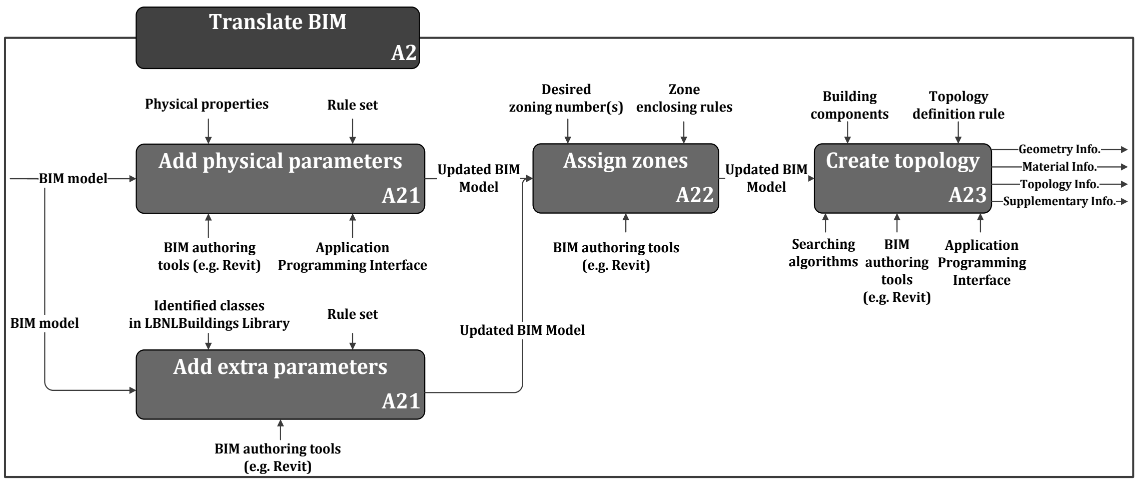


Figure 4. Detailed process model of translate-BIM activity to represent required data flow.

Figure 4 shows the data flow of the Translate BIM activity including topology creation. The information in each step is considered as the requirements for the data model and they will be made into classes. Based on the processes and a mapping guideline (Clayton et al. 2013), I defined the requirements from each step as shown in Table 2.

Table 2. Defined steps and required data.

Steps	Description	Data requirements
Create/Edit BIM	Architects can create building components such as walls, floors, roofs, doors, and windows to represent their design intents and specifications in Revit.	• Geometry information: area, tilt, azimuth, height, width • Material information: thickness • Supplementary information: project location, identification numbers for each building components
Define physical parameters and extra parameters	In order to map missed physical properties in Revit into LBNL Modelica Buildings library, I defined the step of adding the physical parameters, e.g., solar and infrared absorptivities, in existing material properties in Revit. Those physical parameters are added through Revit API. Additional information regarding the glazing system for thermal modeling needs to be prepared in Revit. The properties for glass thickness and ratio of window frame can be added via updating the window family, and parameters for material properties, e.g., solar transmittance, can be added by using Revit API.	• Additional material information: thermal conductivity, specific heat capacity, mass density, solar and infrared absorptivities • Additional thermal information for glazing system: glass thickness, the ratio of window frame • Additional material information for glass: solar transmittance, infrared transmissivity of glass, solar reflectance of surface, U value of frame, infrared and solar absorptivity of window frames

Table 2. Continued.

Steps	Description	Data requirements
Assign zones	To conduct room-to-thermal zone translation, zoning information is required in Revit. I defined thermal zones by using room components in Revit. The room components basically contain the information of height and the area attached to floors. The latitude information for the room can be retrieved from a Revit function.	• Room information: height, area of floors, latitude • Zoning information: latitude
Create topology	The thermal information for heat transfer of the building envelope can be prepared in Revit: the information of how the building envelope is constructed, e.g., boundary condition types, and the number of ports for thermal network connections in *ModelicaBEM* can be generated based on the building topology information retrieved using Revit API. The building topology provides the information of how many and what building components are connected to a room. The information will be the values of the boundary condition variables in a MixedAir instance in *ModelicaBEM*.	• Thermal information: boundary condition types, and the number of ports • Building components information

The data requirements in Table 1 and Table 2 can be represented as classes and properties in a class diagram. The class diagram will also represent object behaviors via defined functions and relationships among classes.

4.1.2. Create Class Definitions

The class diagram contains classes, including properties and functions, and class relationships in application model views and the exchange model view.

The model views represent what the specific data and datasets are for the data requirements from the process models in Revit and LBNL Modelica Buildings library (Revit Model View and Modelica Model View). The Exchange Model View consists of wrapper classes and interface classes and allows *ModelicaBEM* to hold building geometries, material properties, and topology for thermal simulation.

Application Model Views

Revit Model View

Revit models contain architectural data created by the users. Very large data sets can be represented in BIM; however, only part of the data is applicable to thermal simulation. The data for thermal simulation are represented as native Revit instances of classes and relationships shown in Figure 5. These classes can represent the data requirements categorized in Table 2. The description of the classes is adapted from Autodesk references (Autodesk Inc. 2014b). Based on the description, I created a class diagram (Figure 5) using a UML Class Diagram to show the relationships among the classes (Object Management Group 2014).

- ***Revit.Element***: the *Element* class represents geometry information of building components such as area, height, length, volume, and so on through relationships with *Revit.Parameter* class. Such inherited classes from the *Element* class, e.g., *Wall, Floor*, and *Roof (Base)* can have the geometry information.

- ***Revit.Wall***: the *Wall* class, derived from the *Element* class, has additional geometry information such as orientation and width information. Also, the type information of a wall can be defined through *WallType* class.
- ***Revit.RoofBase***: the *RoofBase* class provides additional information such as the roof types. The type property can categorize roof instances with specified purposes such as flat roof or slope roof.
- ***Revit.Floor***: the *Floor* class is inherited from the *CeilingAndFloor* class that provides support for all ceiling and floor objects including geometry information. This *Floor* class has an additional type property representing floor types.
- ***Revit.FamilyInstance***: the *FamilyInstance* represents a single object of a family type such as doors and windows. If a door is created in a wall between two rooms, the connectivity information regarding which room is connected to another can be defined via the properties, *fromRoom* and *toRoom*. The *FamilyInstance* has a relationship with *FamilySymbol*, which enables additional parameters to represent thermal properties such as the frame ratio information. For example, a window family allows creating a thickness parameter to calculate the frame ratio. The parameters are represented through the *FamilySymbol* class, and the *FamilyInstance* class represents a window object.
- ***Revit.Architecture.Room***: the *Room* class represents the basic information such as area, height, and perimeter inherited from the superclass (*SpatialElement*). The volume information of a room is only defined in the *Room* class. Such a

relationship provides access to retrieve required information after a room instance is created.

- ***Revit.Material***: the *Material* class represents material information regarding the color of the material, the name of general material types, the shininess and smoothness of the material, and so on. The category property in this class enables building components to access their material information, i.e., once a wall instance is created, the instance has parameters including a material instance.

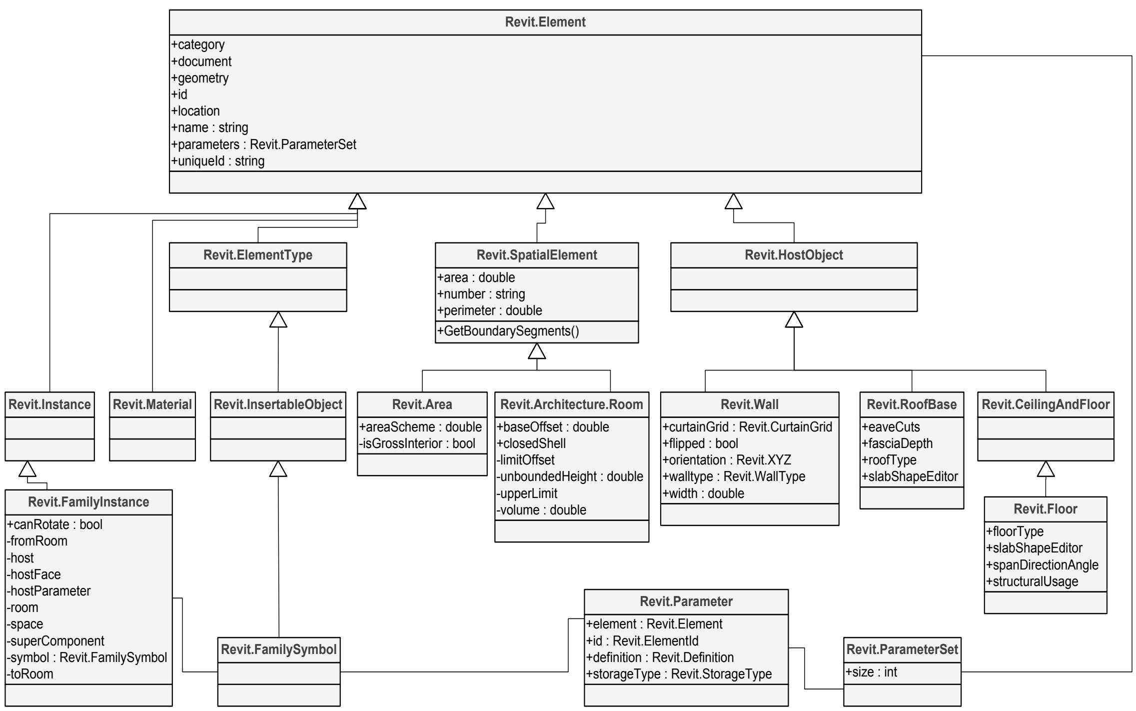


Figure 5. Class diagram of the Revit model view.

Using the objects described in this class diagram, the steps of “Create/Edit BIM”, “Add extra parameters”, and “Assign zone” in the process model can be defined. “Add physical parameters” can be composed through functions in the interface classes.

Modelica Model View

I created a Modelica model view to represent the information identified in Table 1. The model view shows what classes and relationships are defined to create a model through the LBNL Modelica Buildings library. The classes are: MixedAir, GlazingSystem, DoorDiscretizedOpen, FixedBoundary, WeatherData and additional Modelica standard classes.

Figure 6 shows the data subsets in LBNL Modelica Buildings library as classes and relationships. For example, in the model description code block (Figure 2), the mixedAir object is instantiated from the MixedAir class under the Rooms package (Rooms.MixedAir), and the properties in the object are declared in another MixedAir class under the BaseClasses (BaseClasses.MixedAir). LBNL Modelica Buildings library defines the relationship between the two classes: the BaseClasses.MixedAir object is encapsulated as a parameter object in the Rooms.MixedAir object.

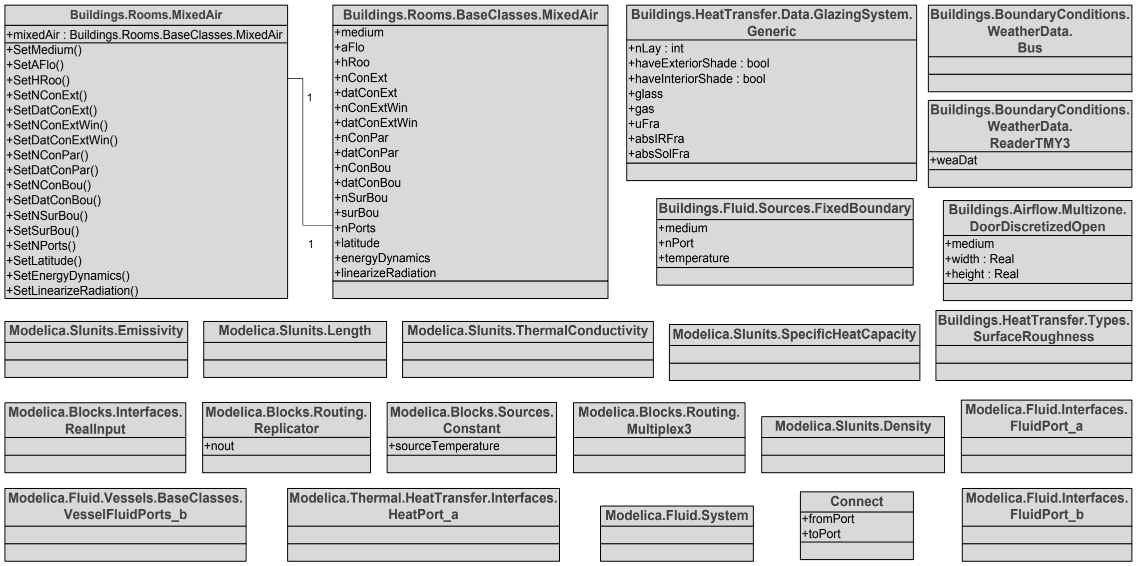


Figure 6. Class diagram of the Modelica model view.

I used the UML specifications to represent the classes in Figure 6 based on the investigation of LBNL Modelica Buildings library.

Exchange MVD

I created an Exchange MVD to define interface and wrapper classes. The Exchange Model View integrates two different semantic models, which represent not only architecture but also engineering points of view. For example, the wall class has a material instance parameter and a different material type in the Modelica model, and the property of the wall instance such as area can be passed to the Modelica model. The room class has wall instances to pass material and area information to parameters in MixedAir class.

Wrapper Classes

Wrapper classes allow *ModelicaBEM* models to follow object semantics of Revit and to utilize LBNL Modelica Buildings library. Wrapper classes adopt the following rules, which represent a common object relationship among building objects.

- A building object in Revit consists of room objects, walls, floors, and roofs enclosing the room.
- A room object in Revit is transformed into a single thermal zone object (MixedAir object in LBNL's BEM). A multi- zone model can be created through connecting multiple room objects.
- Topology for energy modeling can be translated from the connectivity of the Revit building components.

Based on the rules, wrapper classes include *Wall, Floor, Roof, Window, Door*, and *Room* classes. The instances from the classes enable the building component information to be transformed from BIM to *ModelicaBEM*.

Figure 7(a) shows the wrapper classes containing a series of properties that can transfer Revit parameters into *ModelicaBEM*. The relationships represent the rules. For example, the *BIM2BEM.Room* class has relationships with *BIM2BEM.Wall, BIM2BEM.Floor*, and *BIM2BEM.Roof* classes to represent the composition of a building object.

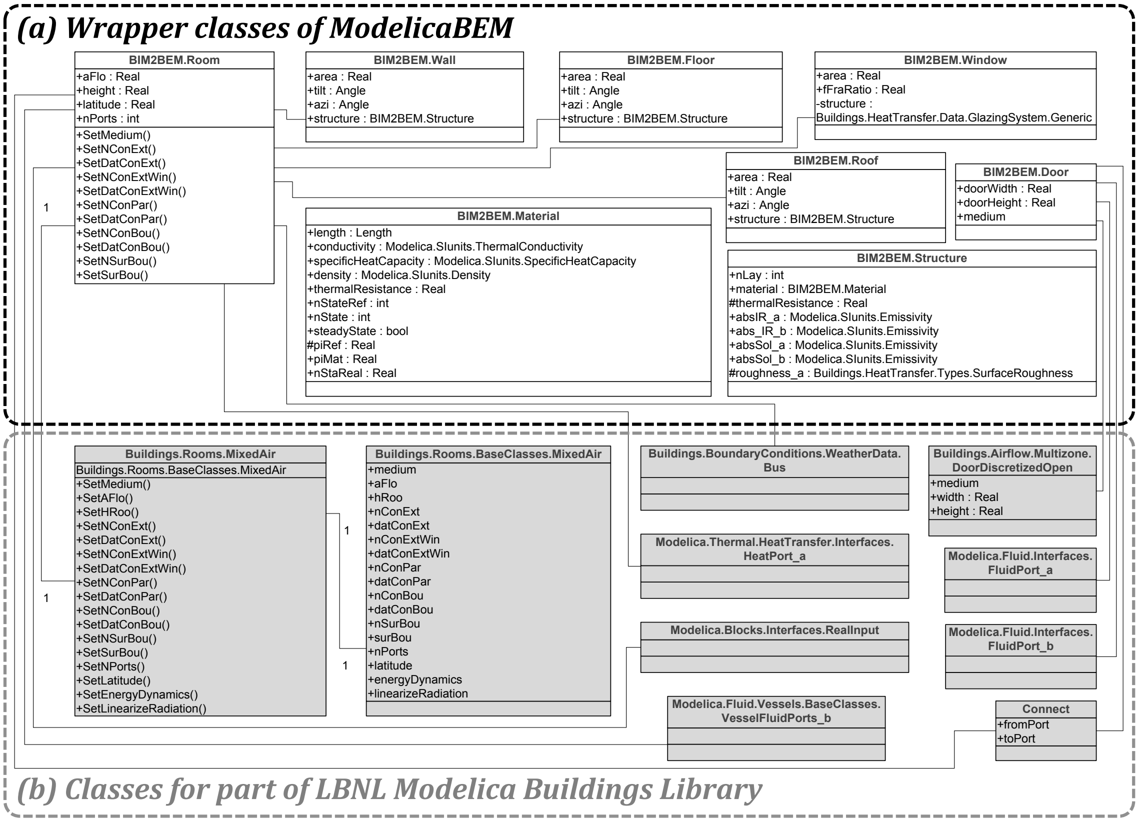


Figure 7. Class diagram of the wrapper classes.

To differentiate class names between the application model views and wrapper classes, I specified the class name by starting with domain name, e.g., the material class names in Revit and the wrapper classes are *Revit.Material* and *BIM2BEM.Material* respectively. The wrapper classes are described below.

- ***BIM2BEM.Room***: the *Room* class represents a single-zone model and wraps the *MixedAir* class of LBNL Modelica Buildings library in it. The *MixedAir* class models a room filled with mixed air (Wetter et al. 2014; Nouidui, Wetter, and Zuo 2012; Wetter and Nouidui 2011; and Wetter 2006 have validated the

MixedAir model and the library). The class properties and functions enable *ModelicaBEM* to populate required information such as building materials and components and thermal boundary conditions. For example, area, tilt, and azimuth as parameters are created in a room object.

- ***BIM2BEM.Wall***, ***BIM2BEM.Floor***, and ***BIM2BEM.Roof***: These classes store the basic building component information from BIM including area, tilt, azimuth, and material layers for energy modeling. The material layer information is represented as a parameter of a *BIM2BEM.Structure* instance. The defined relationships between the Room class and the building components classes enable each instantiated building object to be encapsulated in a room object to pass the geometry information.
- ***BIM2BEM.Window***: the *Window* class represents window geometry data as well as the data of glass panels and window frames in the Modelica model view. The *Window* class can represent material information by defining GlazingSystem.Generic instance as a parameter.
- ***BIM2BEM.Door***: the *Door* is a wrapper class of the *DoorDiscretizedOpen* class and consists of geometry properties such as width and height for calculating multi-zone airflow. Currently, I implemented a closed door to calculate heat transfer and infiltration through the door. A port property represents room-door-room connection as a parameter.
- ***BIM2BEM.Structure*** and ***BIM2BEM.Material***: the *Structure* and *Material* classes are designed to represent material properties and geometry information of

opaque constructions and how materials are assembled. The *Structure* class contains construction information with the number of layers where each layer is represented by *Material* instances. The *Material* class represents thermal information such as thermal conductivity, heat capacity, density, and so on. The additional thermal properties for inner and outer surfaces such as solar and infrared absorptivities exist in the *Structure* class.

Interface Classes

The interface classes are as shown in Figure 8. The methods in the interface classes enable *ModelicaBEM* to contain object instances following Modelica language specifications. The following classes present the functions and object relationships in *ModelicaBEM*.

- ***ModelicaBIM***: the *ModelicaBIM* class has functions to populate a *ModelicaBEM* model, which consists of three parts: building model description, energy components, and connections. The relationships in Figure 8 show the composition (e.g. the building model description is represented with composition relationships among the classes in the wrapper classes).

The functions enable the *ModelicaBEM* objects to instantiate the classes. For example, the *GetMaterialInstances()* function instantiates the *BIM2BEM.Material* class to present related materials in the building model description. As shown in Figure 8d, eight functions map building envelope data, boundary conditions, and room geometry information from Revit to wrapper classes. In addition, two functions are defined to describe energy components and connections. For example, the energy object from the

WeatherData class provides selected weather information to the *ModelicaBEM*; the information of selected geographical location is retrieved from Revit by the user's setting of the building location.

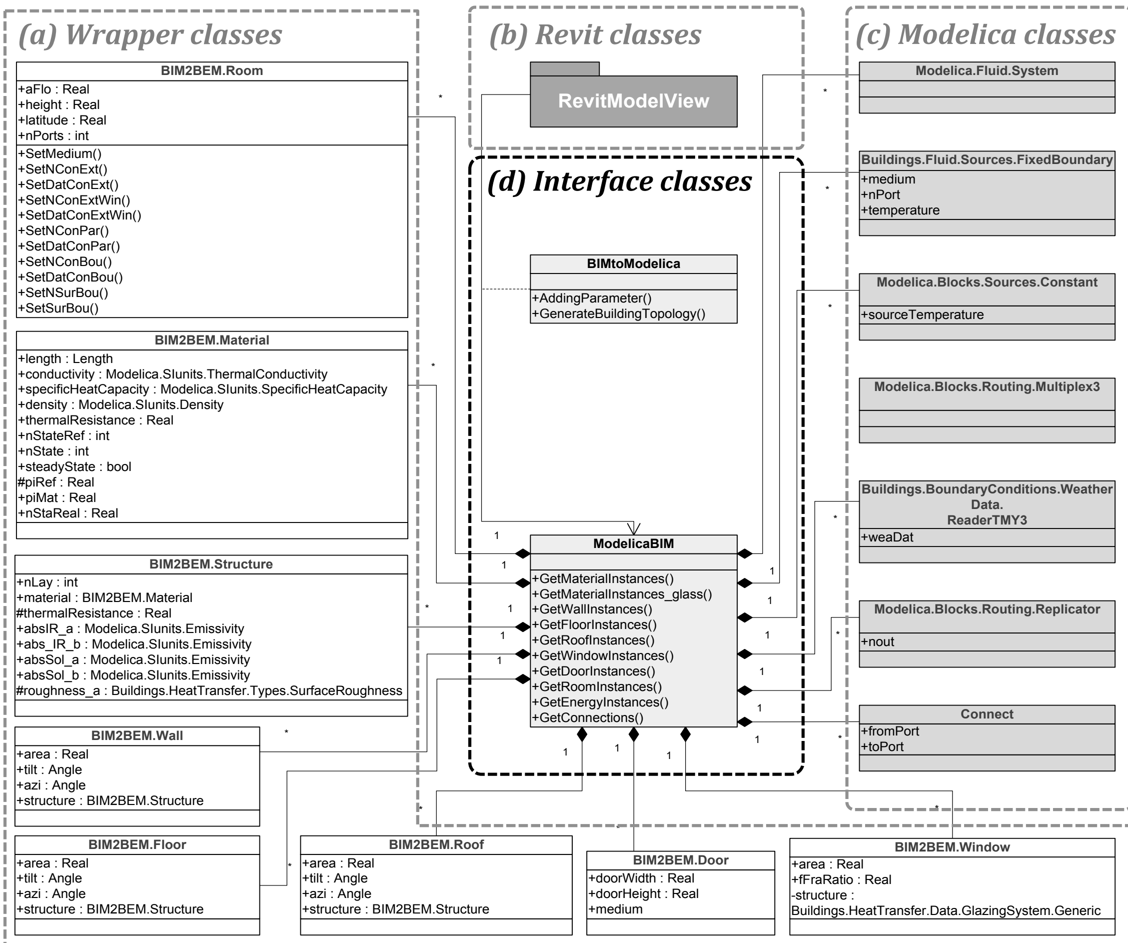


Figure 8. Class diagram of the interface classes and wrapper classes.

- ***BIMtoModelica***: the *BIMtoModelica* class has two functions to support the *ModelicaBIM* class having relationships with Revit classes as shown in Figure 8b. The two functions are implemented using Revit API. The *AddingParameter()* function allows a Revit model to include missing physical properties defined in Table 2. The specific values of boundary conditions in a Room instance can be computed through the *GenerateBuildingTopology()* function, which provides the boundary condition type information and generates building topology from an extended BIM.

A detailed example of creating *ModelicaBEM* through the interface classes will be explained in the following chapter.

4.1.3. Implement Wrapper and Interface Classes

ModelicaBEM can be created with the implementation of the classes of the wrapper and interface.

The implementation of the wrapper classes is accomplished using Modelica language specifications. Figure 9 shows the implementation consisting of four declaration sections: Property, Wrapper component, Energy component, and Connect declaration. The Property declaration demonstrates how defined properties in *BIM2BEM.Room* class can be implemented using Modelica. The Wrapper and Energy component declarations represent the object relationships between the Room class and the connected classes. The Connect declaration implements connectors that represent a physical flow between the wrapper components and the energy components. Based on the implementation, the class package can act as a template when creating new instances.

```
model Room "A room model for single zone with completely mixed air"
  extends Buildings.Rooms.BaseClasses.ConstructionRecords;
  replaceable package Medium = Medium;

  parameter Real AFlo;
  parameter Real Height;
  parameter Real Latitude;
  parameter Integer nPorts;

  Buildings.Rooms.MixedAir mixedAir(
    redeclare final package Medium = Medium,
    final AFlo=AFlo,
    final hRoo=Height,
    final nConExt=nConExt,
    final datConExt=datConExt,
    final nConExtWin=nConExtWin,
    final datConExtWin=datConExtWin,
    final nConPar=nConPar,
    final nConBou=nConBou,
    final datConBou=datConBou,
    final nSurBou=nSurBou,
    final surBou=surBou,
    nPorts=nPorts,
    energyDynamics=Modelica.Fluid.Types.Dynamics.FixedInitial,
    final lat=Latitude,
    linearizeRadiation=false)
    a;

  Modelica.Blocks.Interfaces.RealInput Room_uSha[1](each min=0, each max=1)
    a;
  Modelica.Blocks.Interfaces.RealInput Room_qGai_flow[3](unit="W/m2")
    a;
  Buildings.BoundaryConditions.WeatherData.Bus Room_weaBus
    a;
  Modelica.Fluid.Vessels.BaseClasses.VesselFluidPorts_b Room_ports[nPorts](
    redeclare each final package Medium = Medium)
    a;
  Modelica.Thermal.HeatTransfer.Interfaces.HeatPort_a Room_heaPorAir
    a;
  Modelica.Thermal.HeatTransfer.Interfaces.HeatPort_a Room_heaPorRad
    a;
  Modelica.Thermal.HeatTransfer.Interfaces.HeatPort_a Room_surf_conBou[nConBou] if haveConBou
    a;
  Modelica.Thermal.HeatTransfer.Interfaces.HeatPort_a Room_surf_surBou[nSurBou] if haveSurBou
    a;
equation
  connect(Room_uSha, mixedAir.uSha) a;
  connect(Room_qGai_flow, mixedAir.qGai_flow) a;
  connect(Room_weaBus, mixedAir.weaBus) a;
  connect(Room_ports, mixedAir.ports) a;
  connect(Room_surf_surBou, mixedAir.surf_surBou) a;
  connect(Room_surf_conBou, mixedAir.surf_conBou) a;
  connect(Room_heaPorRad, mixedAir.heaPorRad) a;
  connect(Room_heaPorAir, mixedAir.heaPorAir) a;
  a
end Room;
```

Property declaration

Wrapper component declaration

Energy component declaration

Connect declaration

Figure 9. Implementation of the Room class as a wrapper class based on Modelica language specification.

The interface classes' implementation facilities a system interface to automatically create the required instances of *ModelicaBEM*. Based on the class specification, I developed the *Revit2Modelica* prototype to transfer data from Revit into *ModelicaBEM*. The prototype enables a Revit model to contain additional materials and define building topology for *ModelicaBEM*. For example, the prototype implements the *GenerateBuildingTopology()* function to map construction boundary condition types into *ModelicaBEM*. The boundary condition types are calculated based on our predefined rules as follows:

- Exterior walls, roofs, and floors, which enclose a room and have non-sharing building components with any other rooms, should be mapped into opaque surfaces.
- Exteriors walls enclosing a thermal zone and containing a window(s) should be mapped into opaque surfaces with windows.
- Shared interior walls between rooms should be defined as opaque surfaces on the location of the interior walls as construction boundaries and surround boundaries between thermal zones.
- Interior walls inside a single room should be mapped into interior partitions in a thermal zone.

In addition, object instances can be generated through the *Revit2Modelica* prototype.

The Exchange MVD enables the object mapping between the two applications based on the Object-Oriented modeling concept. In addition, the demonstration shows that the MVD follows the Object-Oriented Programming (OOP) approach in the development, which facilitates natural object mapping between Revit and LBNL Modelica Buildings library. Once the required instances are populated from the prototype, the values of them or instances themselves are encapsulated in other instances. As an example of material objects, the material object of a wall is encapsulated as a parameter object in a wall instance to represent the material information.

4.2. System Interface for the *BIM2BEM* Framework

BIM2BEM supports an iterative Model-Simulate-Evaluate approach. The approach consists of three major steps: create extended BIM, execute *Revit2Modelica*, and report energy performance information back to BIM (Figure 10).

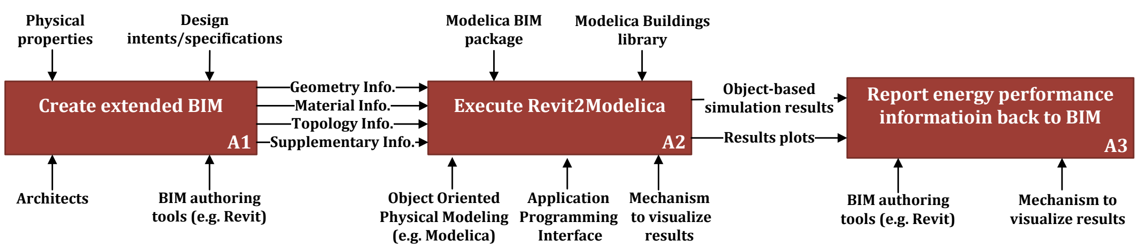


Figure 10. Overall process diagram of *BIM2BEM* using IDEF0.

Figure 10 shows the major implementation steps following the IDEF0 specification. The steps are as follows.

- Create extended BIM (A1): This pre-processing step adds required physical properties into existing BIM (Revit) models using the *AddingParameter* command. Next, designers use Revit to create a BIM model and the *GenerateBuildingTopology* command to obtain the topology of BIM.
- Execute *Revit2Modelica* (A2): Revit2Modelica translates the Revit model into a Modelica thermal model containing objects of the *ModelicaBIM* package classes, which are connected based on the BIM topology. Then *Revit2Modelica* can immediately call Dymola to execute the thermal simulation.
- Report energy performance information back to BIM (A3): *BIM2BEM* reports simulation results as data graphs in the BIM user interface.

Figure 11 shows the detail activities of the component A1 (Create extended BIM), which consists of four major steps. The following section will describe each of the steps.

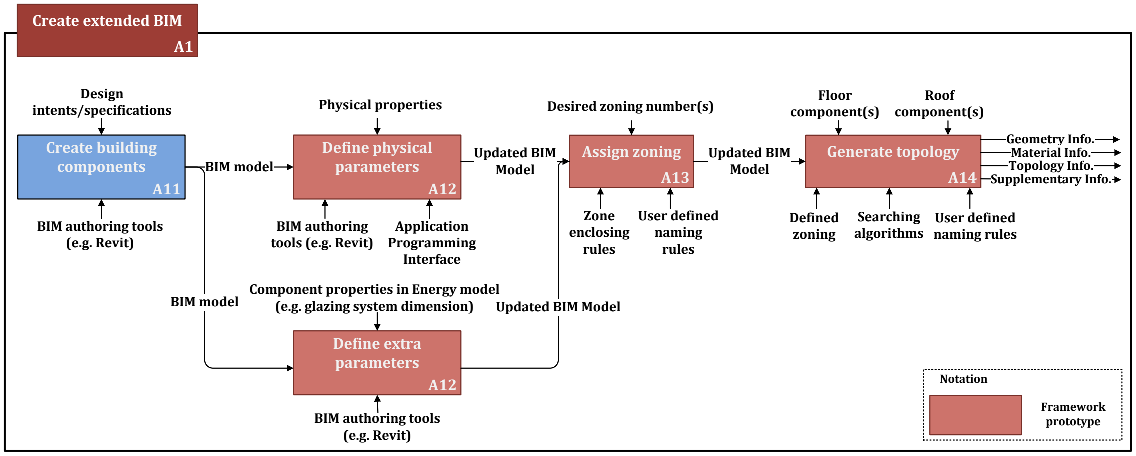


Figure 11. Overall workflow of pre-processing BIM models. The blue box represents the process of only using Revit application, whereas the light maroon boxes stand for the processes of adopting the present methods.

4.2.1. Create Extended BIM

Create BIM in Revit (A11)

Figure 12 shows a sample BIM model created using Revit. Revit enables the users to create and access comprehensive BIM model data such as building components and their properties used in design, construction, and operation. For example, users can create walls with diverse types of materials such as brick, concrete, gypsum board, and so on or define a new wall type by combining of the materials types.

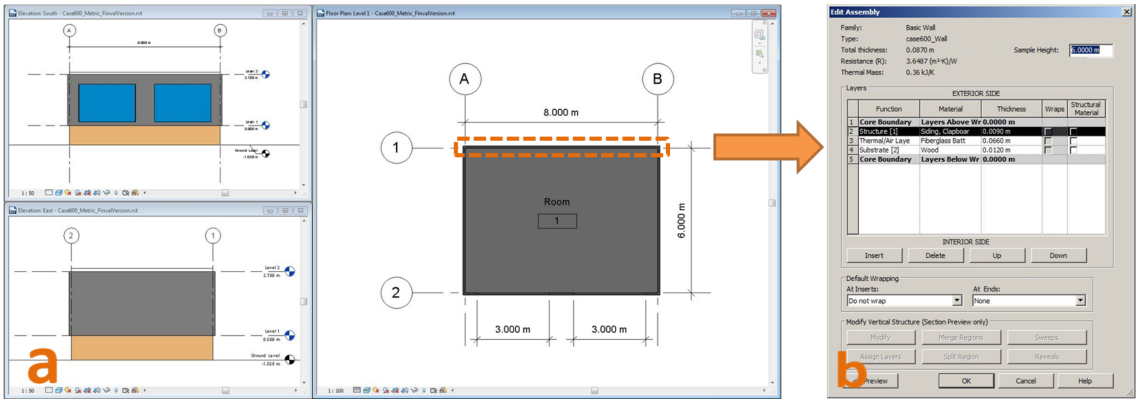


Figure 12. BIM in Autodesk Revit. (a) Elevation and floor-plan Views. (b) A wall assembly layers.

Revit models have a data schema as shown in Figure 5. The schema is different from IFC or gbXML, and can be extended with new objects and new parameters through Revit API as shown in Figure 13. In this study, new material parameters are added into the schema as shown inside the dotted-line box in Figure 13.

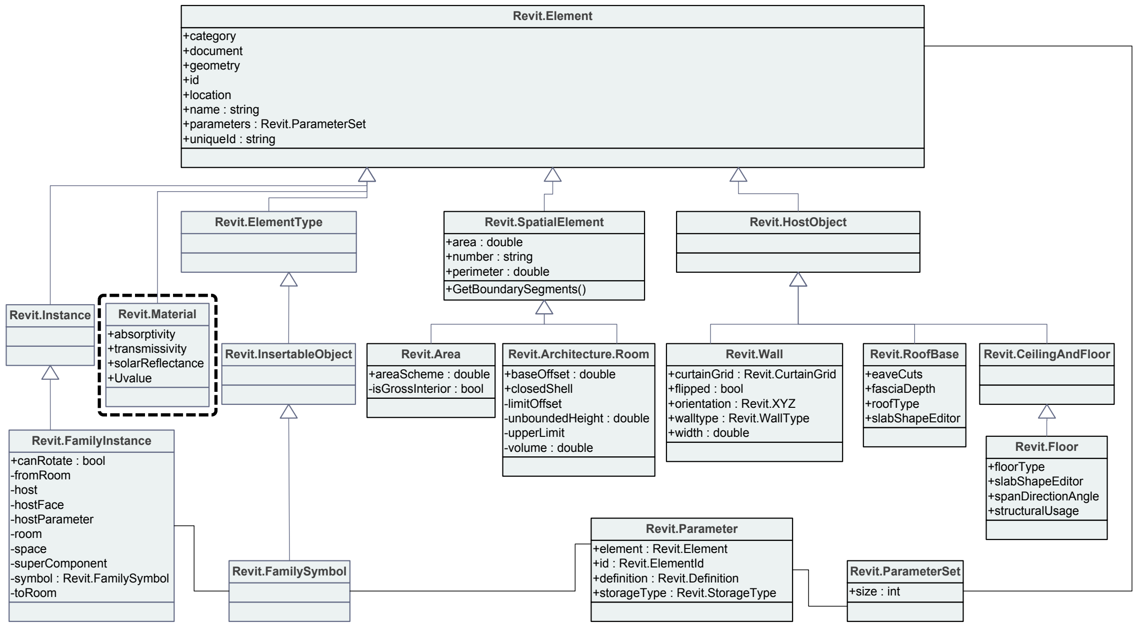


Figure 13. Extended Revit model based on Figure 5 presenting the object class hierarchy.

Add Physical Parameters (A12)

AddingParameter Command

The *AddingParameter* command can add thermal properties to the window object in Revit (Figure 14). Thermal properties including thermal conductivity, specific heat capacity, and mass density are already defined in Revit (Figure 14a). However, LBNL Modelica Buildings library requires additional properties that are missing in Revit, e.g., solar and infrared absorptivities, solar transmittance, infrared transmissivity of glass, solar reflectance of surface, and U value of the frame. The *AddingParameter* command provides a custom parameter input window as shown in Figure 14b. Once the users assign the physical parameters to a material, each building component containing the material will have the thermal properties that can be translated into *ModelicaBEM*.

In this research, the *AddingParameter* command supports the extension of material parameters only in Revit and will be read later by *Revit2Modelica*. The added parameters are not added into standard schemas such as IFC or gbXML.

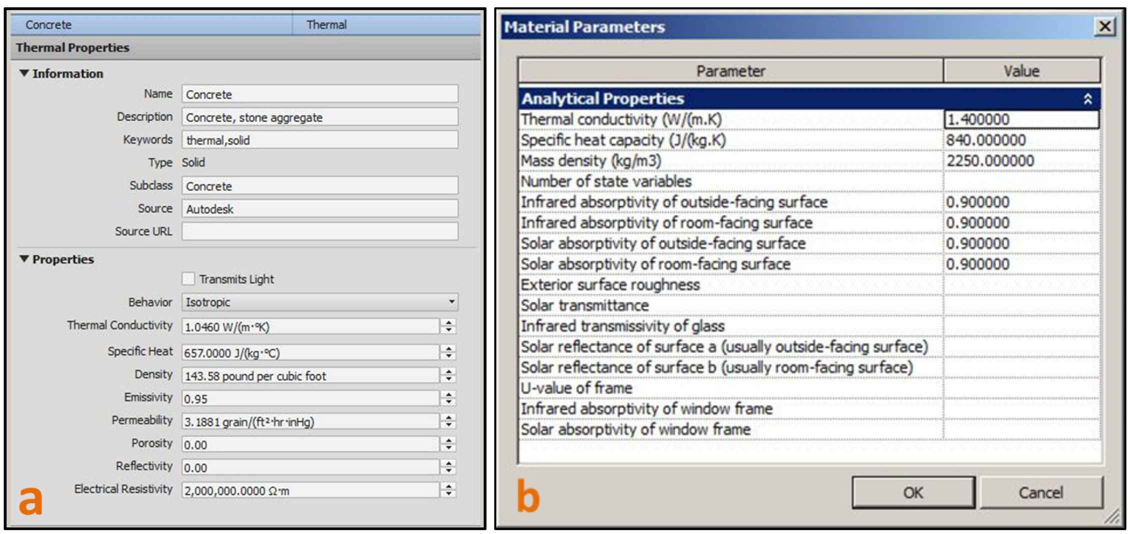


Figure 14. (a) Default material properties in concrete material in Revit. (b) The custom parameters window to add required thermal properties for simulation.

Creating Parameters for Thermal Properties in the Existing Glazing System of Revit

User can create new parameters for the Revit glazing system objects: PBIM_Glass_Thickness, PBIM_Frame_Thickness_Upper, PBIM_Frame_Thickness_Right, PBIM_Frame_Thickness_Lower, and

PBIM_Frame_Thickness_Left to meet the input requirements of ModelicaBEM on glass thickness and the ratio of window – frames (Figure 15).

The added material parameters in Revit will be translated into the Modelica Material class parameters defined in the *ModelicaBIM* Package. The detailed description of how to generate the instantiated material object will be demonstrated in the Experiments and validation chapter.

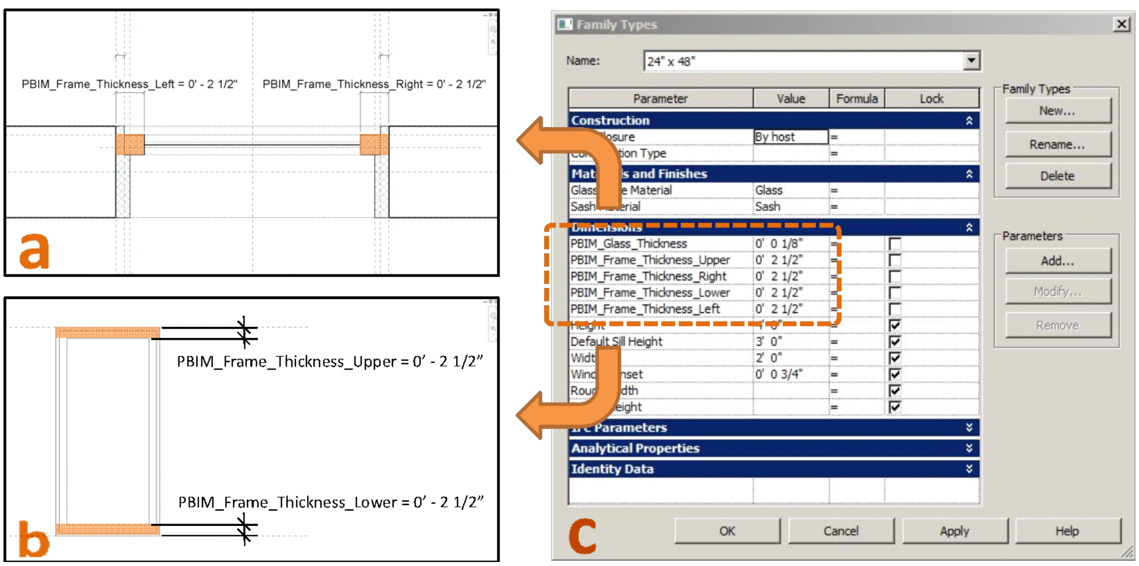


Figure 15. Creating thermal properties in existing glazing systems. (a) Creating frame thickness parameters for left and right frame component in a plan view. (b) Creating frame thickness parameters for upper and lower frame component in an elevation view. (c) The property window for the Window family.

Assign Zoning Information (A13)

The thermal zoning properties consist of geometry information and the construction boundary condition. The geometry information includes location information (latitude), height of the zone, and the area of the zone. The construction boundary condition includes the building topology information, which describes what building components consist of a zone. For example, if a single zone energy model is made of four walls, a roof, and a floor, the zone has the boundary condition as exterior construction with no windows.

A Revit model consisting of one room can be converted into a single-zone energy model. To transfer room objects to thermal zones by mapping the zoning properties, users need to create room objects containing geometry and building topology information in Revit.

The phase to assign zoning information can establish building objects' relationships. For example, based on the room-floor-roof relationship, the area of the floor and the roof can be linked to the room object and calculated as the inner-wetted area of the floor and the roof. If a room consists of multiple floor and roof objects, the room-floor-roof relationship facilitates the retrieval of the corresponding floor and roof geometry for the room.

Geometry Information

Room objects possess room height, area of floor, and volume information. By setting the location of a building (Figure 16), the latitude information can be retrieved and translated into *ModelicaBEM*.

Preparation of Building Object Connectivity

Revit database contains the information about which walls and doors enclose a room object that can be used to calculate the building topology graph horizontally (in a floor plan). To enable the calculation for building object connectivity vertically (for multiple floors), I utilize existing room properties shown in Figure 16a to link the names of the roofs and the floors defined in their object properties in advance; then, specify the corresponding names of the roofs and floors in the room object's property window. For example, after a user named a roof and a floor object as R1 and F1, respectively, the connectivity of the roof and the floor to a room object can be calculated by indicating R1 and F1 as the values of *Ceiling Finish* and *Floor Finish*.

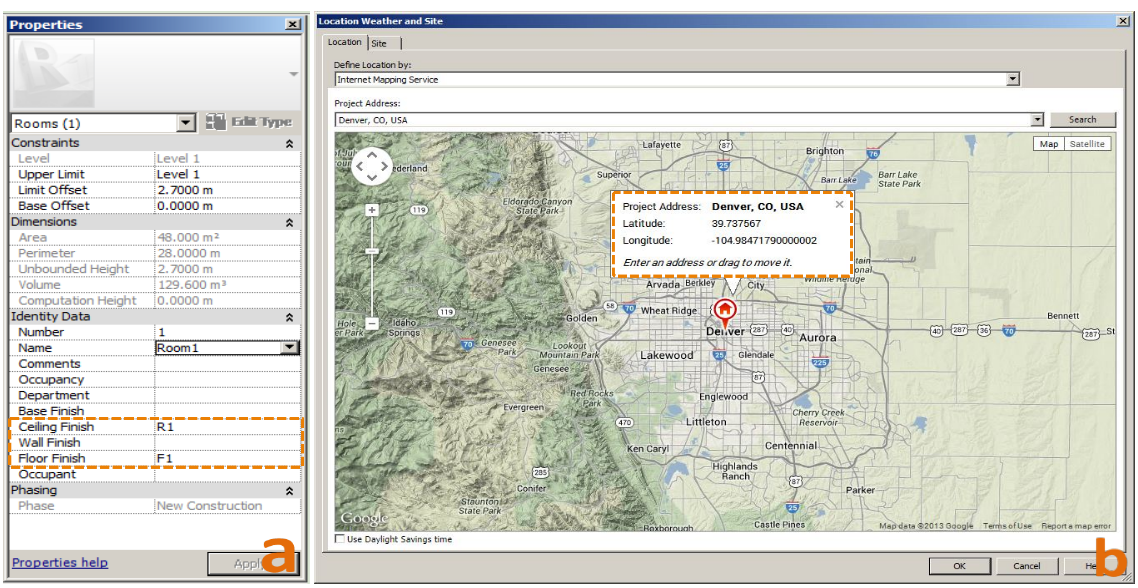


Figure 16. Defining building topology and location information. (a) Preparing topology in the vertical direction by specifying the roof, floor, and room relationship. (b) Specifying the location of the building for linking to weather data and translating the location information into *ModelicaBEM*.

Generate Building Topology (A14)

Based on the assigned zoning information and building object connectivity, building topology can be generated through the developed *GenerateBuildingTopology* command. Figure 17 shows the result diagram representing the building topology information in BIM. The BIM topology can be translated into boundary conditions to define zoning objects in *ModelicaBEM*.

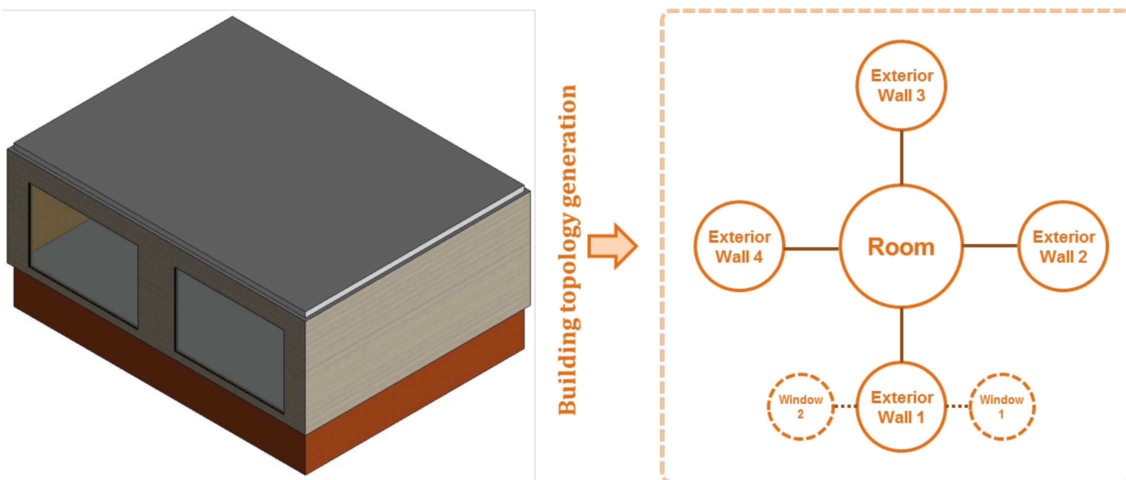


Figure 17. Generating the building topology graph using BIM.

Based on the building topology information, *BIM2BEM* provides the values of each boundary condition parameter in the zoning objects.

4.2.2. Develop Revit2Modelica Prototype

Once BIM is updated with required thermal information and zoning information, a *ModelicaBEM* can be automatically created by Revit2Modelica. As shown in Figure 18, the prototype consists of three major steps: create *ModelicaBEM*, run thermal simulation, and plot data.

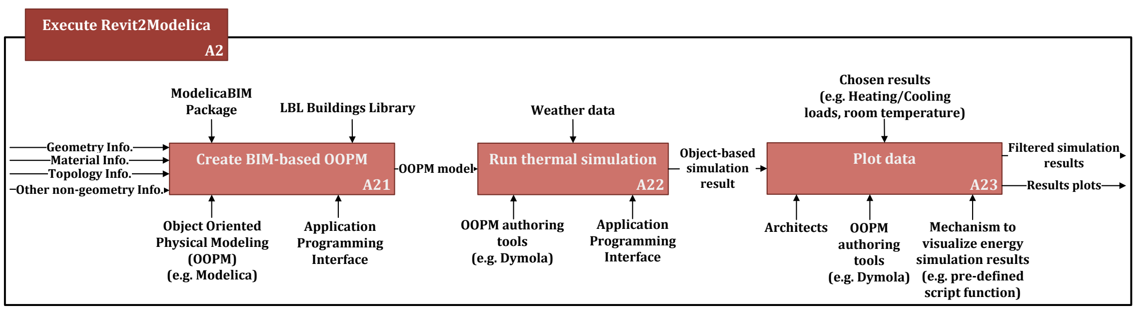


Figure 18. ***Revit2Modelica*** **workflow.**

Create *ModelicaBEM* (A21)

Revit2Modelica can create a *ModelicaBEM* using the data structure and semantics of BIM and the LBNL Modelica Buildings library through the *ModelicaBIM* package. The *ModelicaBEM* objects instantiate the classes in the *ModelicaBIM* package. The objects are categorized as three sections: (1) building model description, (2) energy components, and (3) equations. Revit2Modelica generates Modelica codes to compose the three categories of objects using BIM information.

Building Model Description

The building model description section in *ModelicaBEM* presents building envelope components. The description follows Modelica language specifications and the class definition in the *ModelicaBIM* package. To generate building components in *ModelicaBEM*, I developed a series of Revit API functions (Table 3) to map (1) building envelope information, (2) boundary conditions of the building envelope, and (3) room geometry information (Figure 19) from Revit to *ModelicaBEM*.

Table 3. Functions to generate building model descriptions

Functions	Object properties
GetMaterialInstances(): Getting material objects by retrieving material information including added physical properties from extended BIM models and translating the information into objects of the Material class in *ModelicaBIM* package.	• Thickness • Thermal conductivity • Heat capacity • Density • Thermal resistance
GetMaterialInstances_glass(): Getting glass objects by retrieving windows information from extended BIM models and translating the information into objects of the Glass class in the LBNL Modelica Buildings library.	• Thickness • Thermal conductivity • Solar transmittance • Solar reflectance • Infrared transmissivity of glass • Infrared absorptivity
GetWallInstances(), GetFloorInstances() and GetRoofInstances(): Getting building objects and mapping required properties from extended BIM models into the objects of the Wall, Floor, and Roof classes in the *ModelicaBIM* package.	• Structure of materials including the number of layers of materials, Infrared and Solar absorptivities • Area • Tilt • Azimuth

Table 3. Continued.

Functions	**Object properties**
GetWindowInstances(): Getting window objects with glass material objects as a parameter and mapping windows information from extended BIM models into the objects of the Window class in the *ModelicaBIM* package.	• Area • Frame ratio • Structure of glass materials including the number of glass material layers and glass material names; U-value of frames; Infrared and solar absorptivity of the window frames; Shading status
GetDoorInstances(): Getting door objects and mapping the doors' geometry information from extended BIM model into the objects of the Door class in the *ModelicaBIM* package.	• Width • Height
GetRoomInstances(): Getting room objects and mapping room's geometry information including latitude and height, area information of a floor attached to the room object, and boundary conditions from extended BIM models into the objects of the Room class in the *ModelicaBIM* package.	• Latitude • Height • Area of a floor • Boundary conditions

In the *ModelicaBEM* code sample (Figure 19), the building envelope material object is generated through *GetMaterialInstances()*, which instantiates the Material class (code line 1). The material object is encapsulated as a parameter object in each building component instance. For example, the floor object (line 2) is generated by *GetFloorInstances()*, using the material object instance (shown in the blue box).

The LBNL Modelica Buildings library defines the boundary condition types of building components. The *GenerateBuildingTopology()* command provides the type information of the boundary conditions and generates building topology from the extended BIM. Once the topology is retrieved, the values of it are encapsulated in thermal zoning objects. For example, if a floor, a roof, three exterior walls (without windows and doors), and an interior wall enclose a thermal zone, the value of the number of exterior boundaries (nConExt) is 5 (line 4). In addition, the floor instance generated by *GetFloorInstances()* (line 2) is encapsulated as a parameter in a thermal zoning object (Room1) as shown in line 3, 4, 5, and 6.

A room object is generated by *GetRoomInstances()* with room geometry and other envelope information (line 3). The boundary conditions (line 4 to 7) and their parameters including area, tilt angles, and azimuth (line 4, 5, 6 for exterior boundary surfaces; line 7 for an interior wall surface) are also generated.

```
  //Material description
1 PBIM.BIMPackage.Material Floors181951(x=0.2032, k=1.4, c=840, d=2240, R=0.145142857142857);
  //Building components description
2 PBIM.BIMPackage.Floor F1(structure(material={Floors181951},final nLay=1),area=15.70061376,tilt=3.14159265358979,azi=3.14);
3 PBIM.BIMPackage.Room Room1( redeclare package Medium = MediumA,Latitude=0.739294051238215, Height=3.048, AFlo=15.70061376,
4 nConExt= 5,datConExt(layers={W1_B.structure, W1_L.structure, W1_F.structure,R1.structure,F1.structure},
5 A={W1_B.area, W1_L.area, W1_F.area,R1.area,F1.area}, til={W1_B.tilt, W1_L.tilt, W1_F.tilt,R1.tilt,F1.tilt},
6 azi={W1_B.azi, W1_L.azi, W1_F.azi,R1.azi,F1.azi}),nConExtWin=0, nConBou=1, datConBou(layers={W1_R.structure},
7 A={W1_R.area}, til={W1_R.tilt}),nConPar=0,  nSurBou=0,nPorts=3);
```

Encapsulated

Figure 19. A Modelica code block representing a floor object and a room object generated by *Revit2Modelica*.

Energy Components

The energy components represent required energy information to calculate heat exchange through a building envelope in different processes: convection, conduction, infrared radiation, and solar radiation. *Revit2Modelica* generates the *ModelicaBEM* code for the heat exchange processes by providing specific values for the following components shown in Figure 20: system (line 1); boundary source components (line 2); input connectors (line 2); the number of ports (line 2); convective heat gain (line 3); radiation heat gain (line 4); latent heat gain (line 6); weather data (line 7); control signal for a shading device (line 8); and replicator (line 9).

```
  //Energy components description
1 inner Modelica.Fluid.System system
  ;
2 Buildings.Fluid.Sources.FixedBoundary boundary(redeclare package Medium = MediumA, nPorts=2,T=293.15)
  ;

3 Modelica.Blocks.Sources.Constant qConGai_flow(k=0)
  ;
4 Modelica.Blocks.Sources.Constant qRadGai_flow(k=0)
  ;
5 Modelica.Blocks.Routing.Multiplex3 multiplex3_1
  ;
6 Modelica.Blocks.Sources.Constant qLatGai_flow(k=0)
  ;
7 Buildings.BoundaryConditions.WeatherData.ReaderTMY3 weaDat(filNam="//weatherdata/USA_IL_Chicago-OHare.Intl.AP.725300_TMY3.mos")
  ;
8 Modelica.Blocks.Sources.Constant uSha(k=0)
  ;
9 Modelica.Blocks.Routing.Replicator replicator(nout=1)
  ;
```

Figure 20. A Modelica code block representing energy components for thermal simulation generated by *Revit2Modelica*.

Equation

In Modelica models, objects are connected through ports using the *connect* command, which constructs equations (such as physics law equations) to simulate physical processes. In *ModelicaBEM*, building and energy objects are connected using this method. As shown in Figure 21, *Revit2Modelica* generates the *connect* statements to establish the connections between energy components (line 1), an energy component and a building component (line 2), and buildings components (line 3, 4, and 5).

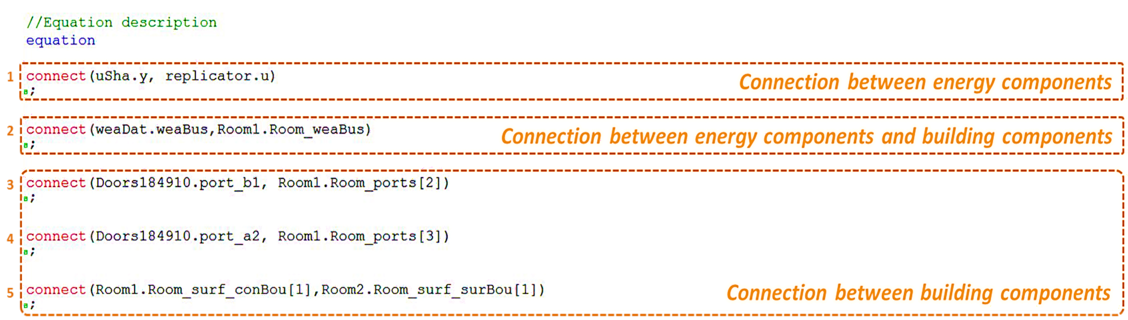


Figure 21. A Modelica code block representing equations (connections) generated by *Revit2Modelica*.

Revit2Modelica creates the building components, energy components, and the equations and save them into a *ModelicaBEM* file.

Run Thermal Simulation (A22)

Dymola provides a Modelica modeling and simulation environment. To execute the simulation from Revit, I developed functions using Revit API to Dymola commands to run the simulation. To communicate between Revit and Dymola, I adopted the NDde library, which enables .NET applications to integrate applications using Dynamic Data Exchange (DDE) (Microsoft Corporation 2014). The library helps call the following

Dymola commands with passed arguments from Revit to run the simulation (Figure 22c).

- *OpenModel* to open a Modelica model produced by *Revit2Modelica*.
- *SimulateModel* to execute energy simulation with the settings of the simulation time and the time-step.
- *IncludePlot* to generate result plots after completing the simulation.

After receiving the simulation time and the *ModelicaBEM* file path information, Dymola starts thermal simulation automatically, which is performed in the background of Revit.

Plot Data (A23)

Modelica simulations provide object-based results. Utilizing LBNL Buildings library, *ModelicaBEM* can provide designers with building object-based thermal performance results, for example, a room's temperature, the heat transfer resulted from conduction, and heating and cooling loads, etc. This feature is very helpful for designers to find individual building object performances (in addition to the overall performance of the building) and decide on how to improve the underperformed objects.

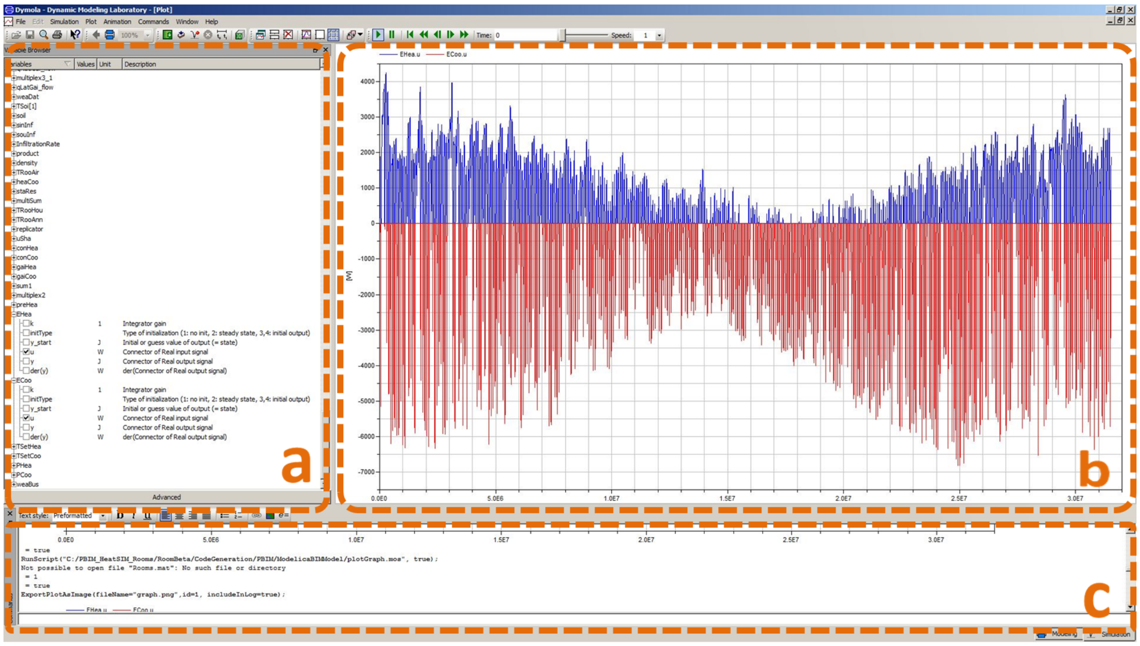


Figure 22. Thermal simulation execution and object-based simulation results in Dymola. (a) An object selection panel. (b) Simulation result for the selected object-a room: heating load (blue line) and cooling load (red line). (c) A command console to show a simulation run through scripts called from *Revit2Modelica*.

Revit2Modelica enables selecting objects in Revit to inspect the objects' thermal performances. Dymola scripts are generated by *Revit2Modelica* to automatically select any objects in *ModelicaBEM* (Figure 22a) for the designers to inspect the results (Figure 22b).

4.2.3. Report Energy Performance Information Back to BIM

As a final step to provide enriched simulation results with object-based data graphs for building designers, I created a series of steps as shown in Figure 23 to report the results in Revit.

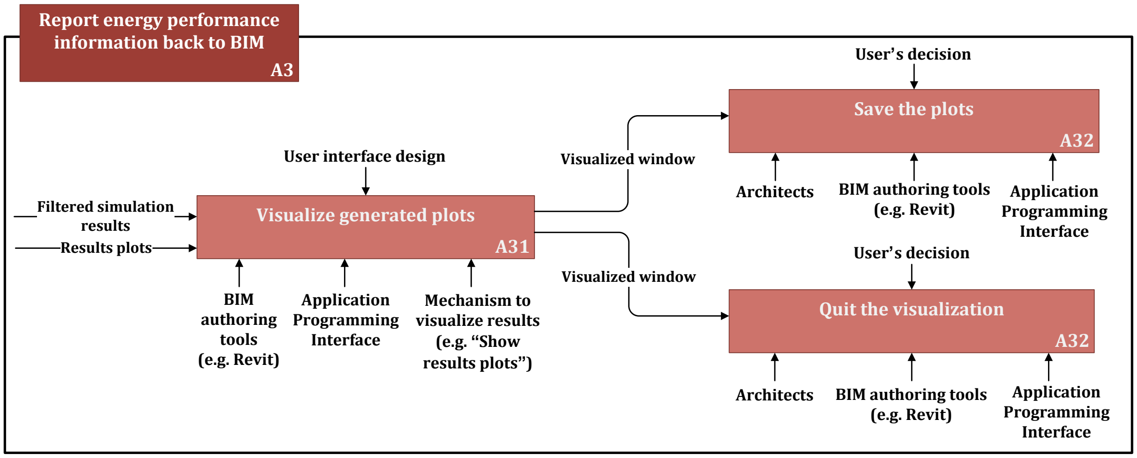


Figure 23. Workflow for reporting energy performance information to BIM.

Visualize Generated Plots (A31)

Based on the generated data graphs in Dymola (Figure 22b), I developed a user interface shown in Figure 24 enabling building designers to inspect the simulation results directly in a Revit user interface window. The benefit of this is that the result graph can be linked to the corresponding building object for future development to enhance the visualization of building object performance.

Save the Plots or Quit the Visualization (A32)

Once *Revit2Modelica* has completed the visualization, users can save the graph as a separated file using the "Save" function in the user interface (Figure 24), which can be utilized for the comparison of results from different design options.

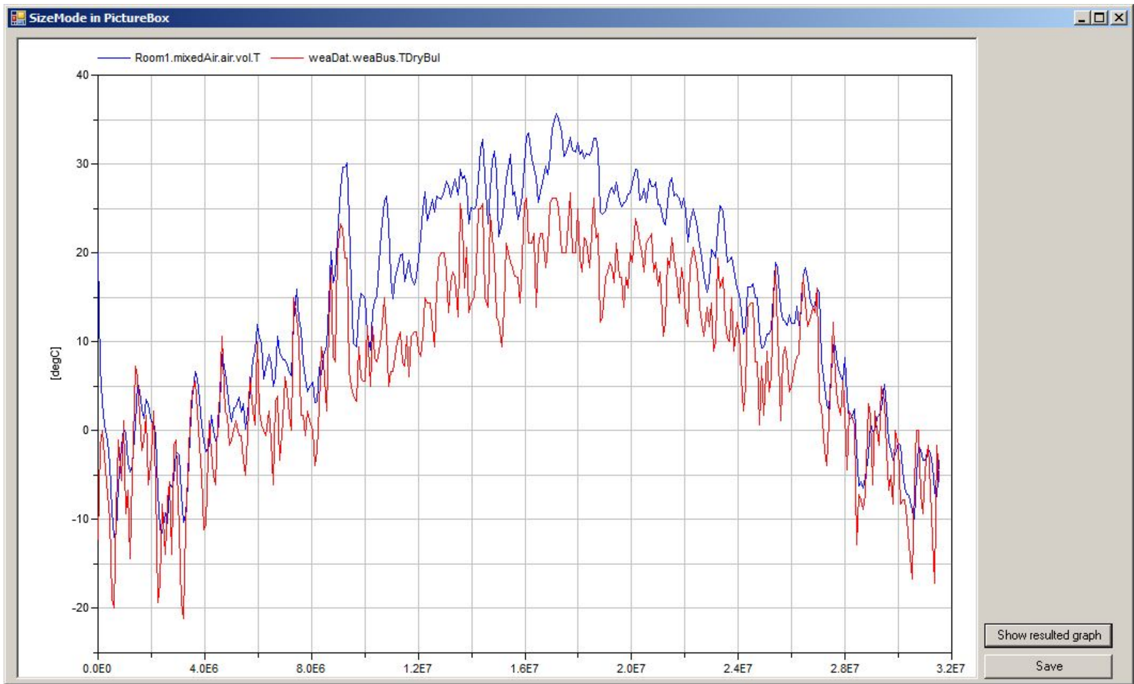


Figure 24. A user interface to provide the simulation results of selected objects from *ModelicaBEM* in Revit. A simulation result graph: a room temperature (blue, upper curve) and weather data (red, lower curve) for a year.

5. EXPERIMENTS AND VALIDATIONS

The experiments and validations include four test cases to validate the *BIM2BEM* framework. To validate the framework, I conducted (1) experiments by applying the class package and *Revit2Modelica* to BIM models and (2) simulation result comparisons between two Modelica models for each test case: one generated automatically using *Revit2Modelica* and the other manually following the LBNL Modelica Buildings library samples' approach (Test Case 1 manual model is created by LBNL Modelica Buildings library team and other manual models are created by the research team). For the experiments, four test case models are used. I hypothesized that if our exchange MVD represents all requirements, and if the implementation of the MVD and the system interface is accurate, the two Modelica models of each test case can produce close or identical simulation results.

For the test cases, I created corresponding BIM models using Autodesk Revit Architecture for a one-room model (Test Case 1), a two-room model (Test Case 2), a two-room model having two windows and an interior door (Test Case 3), and a two-story model having two windows (Test Case 4). Test Case 1 has one thermal zone, six exterior surfaces and two windows. Test Case 2 has two thermal zones, six exterior surfaces, and one interior wall. Based on Test Case 2, other models are created to present more building components such as windows, doors, and a new story. The following sections discuss the test cases and their result analyses. Test Case 1 result is presented with Test Case 1, while Test Case 2, 3, and 4 results are presented together because these cases are developed based on the same initial BIM model.

5.1. Test Case 1: A Basic Building Model with a Single Thermal Zone

To validate the developed *BIM2BEM* framework, I conducted (1) an experiment by applying *BIM2BEM* to BESTEST Case 600 and (2) simulation result comparisons. For the experiment, I created a corresponding Revit model of BESTEST Case 600, conducted BIM pre-processing and the *Revit2Modelcia* model translation and simulation processes, and demonstrated the presentation of simulation results in Revit.

The BESTEST Case 600 building model is defined below (Judkoff and Neymark 1995; Pedersen et al. 2001);

- One room of a single thermal zone dimensioned at 8.0 m, 6.0 m, 2.7 m for length, width, and height, respectively.
- The building has two south-oriented 6 m^2 windows but no doors.
- The building has building envelop material properties as shown in Table 4.
- The building has no shade and no internal heat gains from occupancy and equipment.
- The floor is located above the ground level without attaching to the ground.
- The building is located in Denver, Colorado, U.S.A.
- The building is lightweight with room temperature control set to 20 °C for heating and 27 °C for cooling.

The experiment has been performed using Autodesk Revit Architecture (a BIM authoring tool), Dymola® (a Modelica development environment), LBNL Modelica Buildings library, a solver tolerance of 10^{-6}, and a simulation interval of 3600 seconds for one year.

Table 4. Material specification for BESTEST 600 adapted from (Judkoff and Neymark 1995; ASHRAE 2010).

Material	Thickness (m)	Thermal conductivity (W/m K)	Specific heat capacity (J/ kg K)	Mass density (kg/m^3)
Exterior wall (inside to outside)				
Plasterboard	0.012	0.160	840	950
Fiberglass quilt	0.066	0.040	840	12
Wood siding	0.009	0.140	900	530
Floor (inside to outside)				
Timber flooring	0.025	0.140	1200	650
Insulation	1.003	0.040	0	0
Roof (inside to outside)				
Plasterboard	0.010	0.160	840	950
Fiberglass quit	0.1118	0.040	840	12
Wood siding	0.019	0.140	900	530

I conducted the case study to apply *BIM2BEM* to the BESTEST Case 600 building with the steps shown in Figure 25: (1) Create a BIM using Revit based on the BESTEST Case 600 building description (Figure 25a), (2) Create an extended BIM with additional thermal properties that do not exist in current BIM models, e.g. by adding wall's material layers such as plasterboard, fiberglass quilt, and wood siding (Figure 25b), (3) Assign zoning information (Figure 25c), (4) Generate the building topology to support *Revit2Modelica* in creating the construction boundary conditions (Figure 25d), and (5) Execute *Revit2Modelica* to generate the *ModelicaBEM* program code, run the thermal simulation, and report object-based simulation results such as annual indoor temperatures and outdoor dry bulb air temperatures, heating and cooling loads, and global horizontal radiation back in Revit (Figure 25e).

From the experiment, *BIM2BEM* demonstrates (1) the use of BIM as a common user interface for architectural design and building thermal simulation and (2) the potential to support interoperability between BIM and *ModelicaBEM*.

5.1.1. Simulation Result Comparisons between Two Modelica Thermal Models

For the same BESTEST Case 600 building, the simulation results from the *ModelicaBEM* created by *Revit2Modelica* are comparable with the results from the LBNL Modelica Buildings library, while the LBNL's results have been validated (Nouidui, Wetter, and Zuo 2012). Note that the LBNL's model is created by manually writing the building description code in Modelica but my model is automatically created based on the BIM model using *Revit2Modelica* (with some manual BIM preparation). I created two *ModelicaBEM* models for the BESTEST Case 600 building to demonstrate (1) a room temperature change without an HVAC system module and (2) annual heating and cooling loads with a default HVAC system module.

The room temperature of the *ModelicaBEM* model is similar to that of the LBNL Modelica Buildings library model: the highest temperature is acquired at 3pm on October 17th and the lowest temperature is obtained at 7am on January 4th. The maximum indoor temperature is 65.7 °C in the *ModelicaBEM* and 65.9 °C in the LBNL model both at 3pm on October 17th.

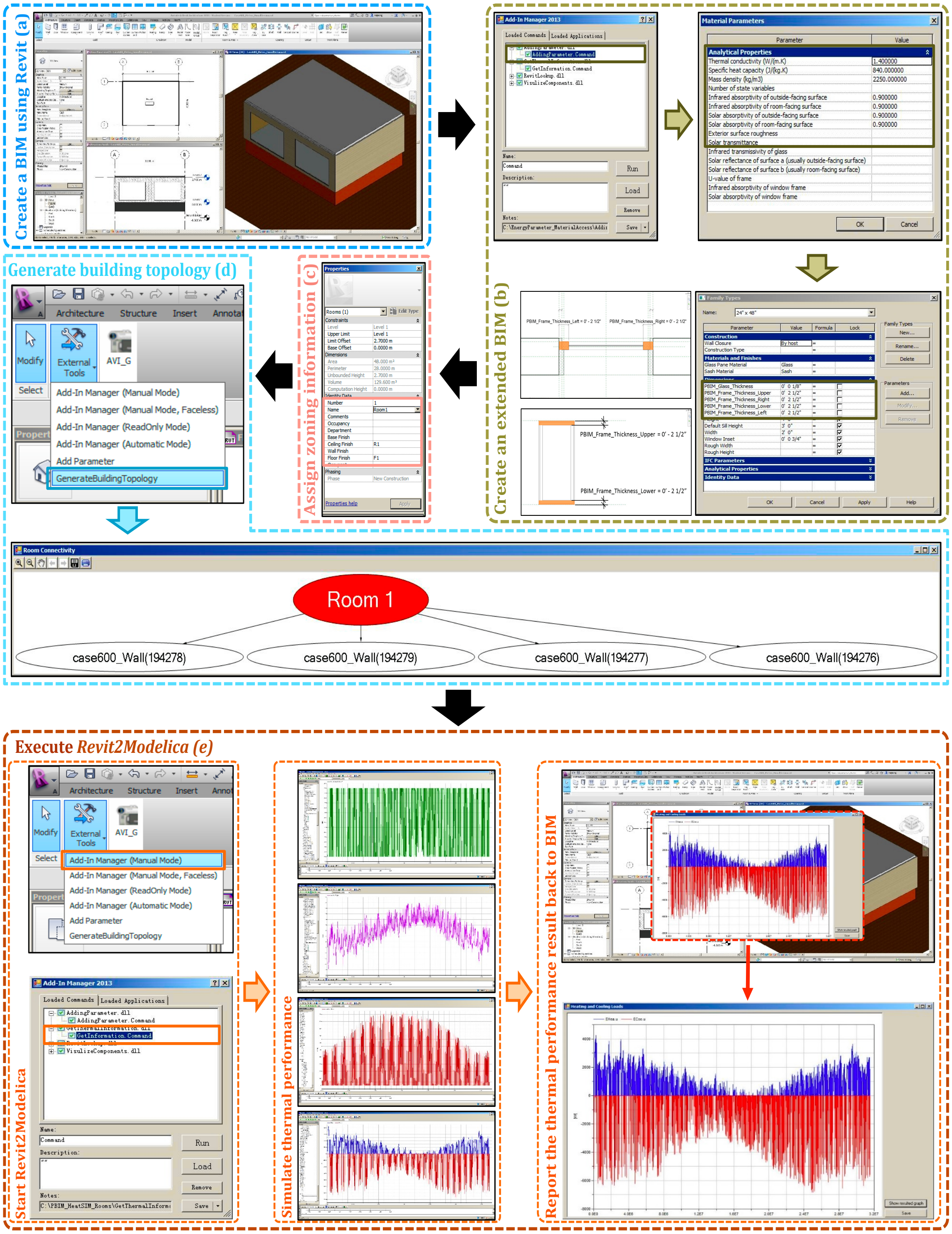


Figure 25. BIM to *ModelicaBEM* workflow using BESTEST Case 600 as a case study.

The minimum temperature is -19.85 °C in the *ModelicaBEM* and -19.83 °C in the LBNL model both at 7am on January 4th.

Figure 26 and Figure 27 show the indoor air temperature of the two models, a dry bulb outdoor temperature, and global horizontal radiation from January 1st to 8th and from October 11th to 18th respectively. The rooms in both the models have almost identical indoor temperature variation: two data series of indoor air temperature are overlapping. The maximum difference of indoor temperature between the two models is 0.249 °C.

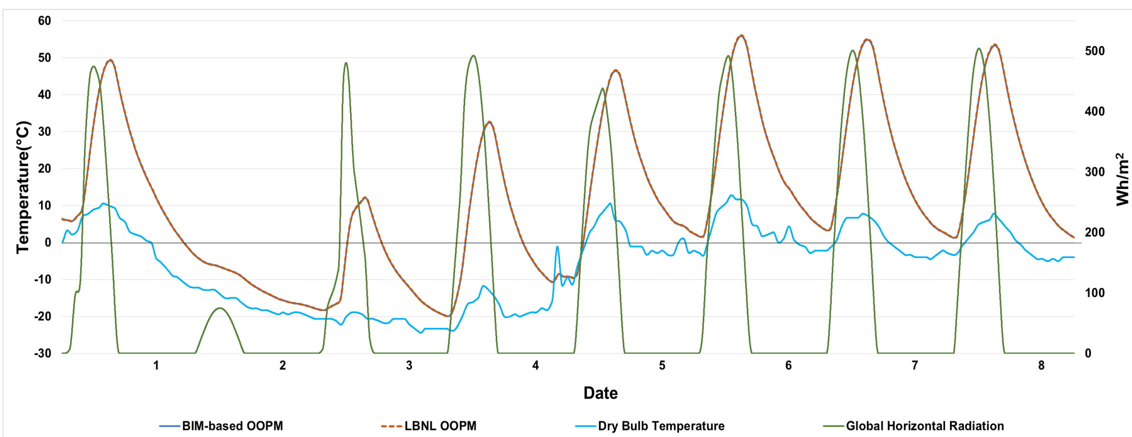


Figure 26. Minimum temperature comparison for January 1st to 8th.

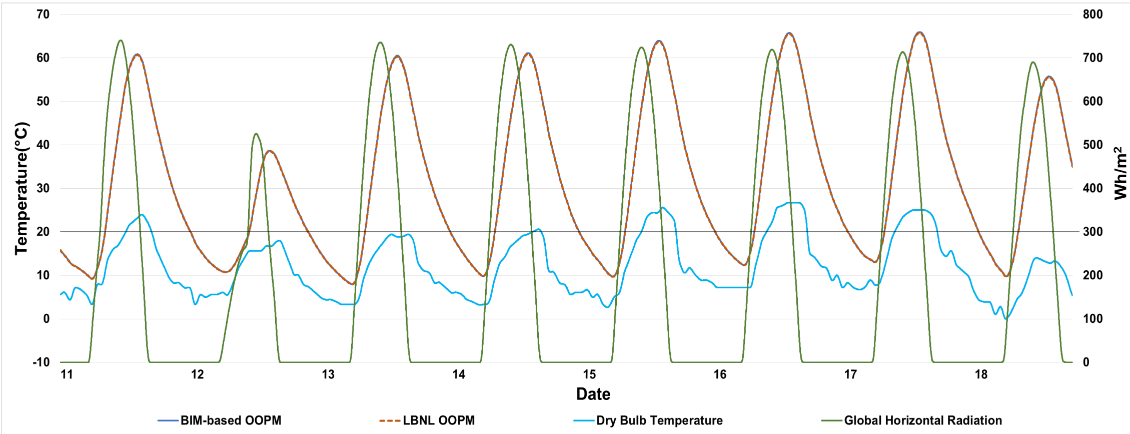


Figure 27. Maximum temperature comparison for October 11th to 18th.

In addition, the figures show that outdoor dry bulb temperature and global horizontal radiation affect the fluctuation of the indoor air temperature.

I also conducted the comparison of annual heating and cooling loads between the two models. As shown Figure 28 in and Table 5, which are adapted from the corresponding figure and tables in (Nouidui, Wetter, and Zuo 2012), all results are very close to LBNL's models; the simulation results of the *ModelicaBEM* (5.475 MWh of heating and 6.941 MWh of cooling loads) are similar to those of the LBNL's model (5.44 MWh of heating and 6.941 MWh of cooling loads) and comparable with other simulation tools in ASHRAE Standard 140-2007.

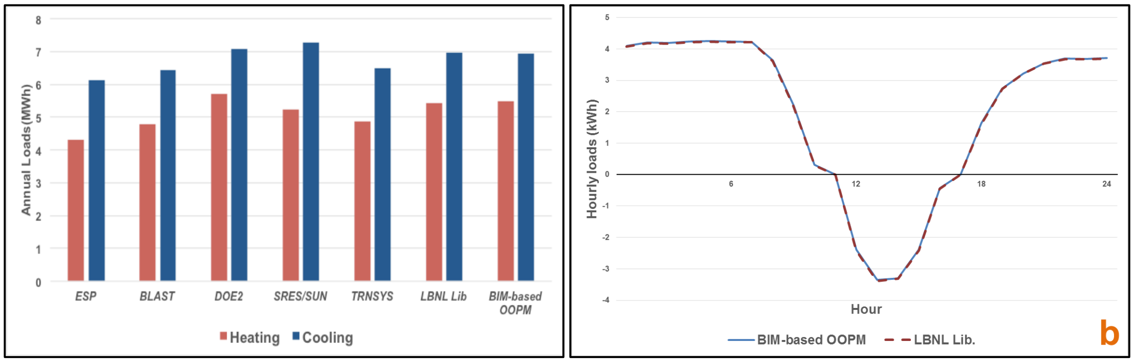


Figure 28. Comparison of annual heating and cooling loads. (a) Annual heating and cooling loads of BESTEST Case 600 from the *ModelicaBEM*, LBNL's model, and other simulation tools in (ASHRAE 2010). (b) Comparison of hourly heating and cooling loads between *ModelicaBEM* and the LBNL's model for January 4th.

Figure 28b shows the comparison of hourly heating and cooling loads on the day of peak heating loads (January 4th) between *ModelicaBEM* and the LBNL's model; the two curves in the figure are closely overlapping. The Figure 28b shows that the two models simulated cooling load from 11 am to 5 pm and heating load performance for the rest of the day identically.

Table 5 shows the comparison among *ModelicaBEM*, LBNL's model, and other simulation tools in terms of the annual peak heating and cooling loads.

Table 5. Annual hourly integrated peak heating and cooling loads of BESTEST Case 600 adapted from (Nouidui, Wetter, and Zuo 2012).

Simulation tools	Loads (KW)		Date		Hour	
	Heating	Cooling	Heating	Cooling	Heating	Cooling
ESP	3.437	6.194	January 4th	October 17th	5	13
BLAST	3.940	5.965	January 4th	October 16th	5	14
DOE2	4.045	6.656	January 4th	October 16th	5	13
SRES/SUN	4.258	6.827	January 4th	October 16th	2	14
TRNSYS	3.931	6.486	January 4th	October 16th	6	14
LBNL Lib.	4.229	6.821	January 4th	October 17th	5	13
ModelicaBEM	4.248	6.817	January 4th	October 17th	5	13

Overall, the *Revit2Modelica* produces very similar simulation results as the LBNL's model. This is expected because both of the models use the same thermal simulation algorithm employed by LBNL Modelica Buildings library.

However, there are two major differences: the model structure and the modeling approach, which may cause inconsistency in the simulation results between the *Revit2Modelica*-generated model and LBNL Modelica Buildings library-provided model, if *Revit2Modelica* were not correctly built. In terms of the model structure, *Revit2Modelica*-generated model uses the classes of objects (and their semantics, e.g. a room) that can be found in the architectural BIM models, while the LBNL's model uses the classes of objects with semantics usually found in BEM models, e.g. MixedAir. Moreover, I found that there was a difference between the two structures in terms of the order of building enclosure elements as arguments in Modelica thermal calculation functions. Reducing the model tolerance in Dymola can reduce the result differences.

In addition, the two different modeling approaches (manual and automatic) are used in creating the BESTEST model. The automatic approach allows the generated BEM to be reasonably more sophisticated to better reflect the actual building configuration. For example, the two windows on the south wall are combined into one window with an equivalent area in LBNL's model but our algorithm splits the wall into two walls each having one of the windows (our research team think this approach is better because it can deal with the case when the windows have different materials).

I use the BESTEST sample for the validation because LBNL Modelica Buildings library includes this model and uses it for its own validation.

5.2. Test Case 2: A Building Model with Two Thermal Zones

Test Case 2 presents a two-thermal-zone model without windows and doors. The building dimensions are 8.0 m*6.0 m* 2.7 m respectively (Figure 29). The basic material information is defined in Table 6.

Table 6. Material specification.

Building Components	Thermal Conductivity (W/m K)	Specific Heat Capacity (J/ kg K)	Mass Density (kg/m^3)	Thickness (m)
Walls	0.140	900	530	0.100
Floor	0.140	1200	650	1.028
Roof	0.160	840	950	0.150

To add required physical properties in Table 6, I used the *AddingParameter()* function as shown in Figure 29b.

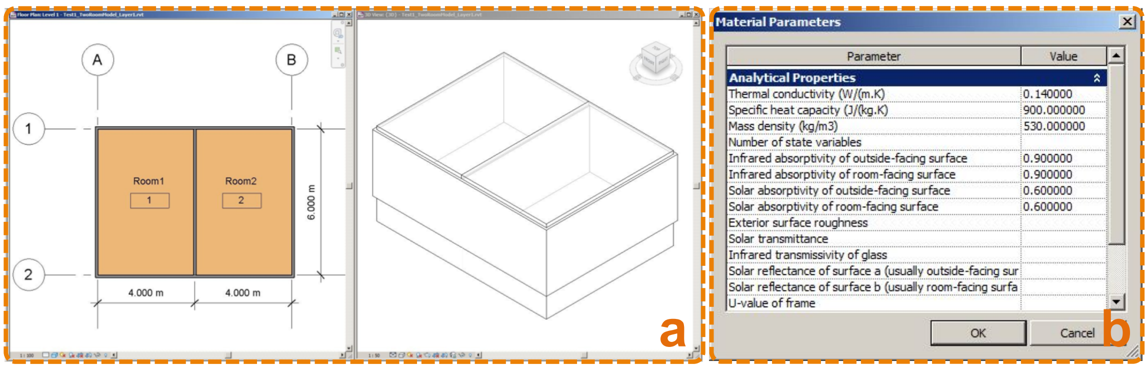


Figure 29. A two-thermal-zone Revit model. (a) Floor-plan and isometric views. (b) The custom parameter window for adding additional physical properties.

The *Revit2Modelica* prototype translates the two-room Revit model into a *ModelicaBEM*. Figure 30 shows the generated instances for *ModelicaBEM* from the *Revit2Modelica* prototype. The *GetMaterialInstances()* function in *ModelicaBIM* class can collect each material instance information used in a BIM model following the Material class of the implemented class package. Wall material objects are instantiated and values for physical properties, which are prepared through the custom parameter window, are assigned to the parameters. Then, the wall objects are instantiated based on the geometry information and the material instances.

The *GenerateBuildingTopology()* function in the *BIMtoModelica* class can retrieve the values of the boundary conditions. As shown in Figure 30, the left room has five opaque surfaces (nConExt and datConExt) for three exterior walls, a roof, and a floor, and one opaque surface that is for the interior wall between the thermal zones (nConBou and datConBou); the right room has one opaque surface on the same interior wall between the thermal zones (nSurBou and surBou). The interior wall instance is used in each room (Room instances in Figure 30).

```
//Wall material information
  PBIM.BIMPackage.Material WallsMaterial194276(x=0.1, k=0.14, c=900, d=530, R=0.714285714285714);

  PBIM.BIMPackage.Material WallsMaterial205734(x=0.1, k=0.14, c=900, d=530, R=0.714285714285714);

  PBIM.BIMPackage.Material WallsMaterial194278(x=0.1, k=0.14, c=900, d=530, R=0.714285714285714);
```

Material instances

```
//Wall information
  PBIM.BIMPackage.Wall Walls194276(structure(material={WallsMaterial194276},
    final nLay=1,absIR_a=0.9,absSol_a=0.6),area=11.0700000000014, tilt=1.5707963267949, azi=3.14159265358979);

  PBIM.BIMPackage.Wall Walls205734(structure(material={WallsMaterial205734},
    final nLay=1,absIR_a=0.9,absSol_a=0.6),area=16.200000000003, tilt=1.5707963267949, azi=-1.5707963267949);

  PBIM.BIMPackage.Wall Walls194278(structure(material={WallsMaterial194278},
    final nLay=1,absIR_a=0.9,absSol_a=0.6),area=11.0700000000014, tilt=1.5707963267949, azi=0);
```

Wall instances

```
//Room information
  PBIM.BIMPackage.Room Room1(...
    nConExt= 5,datConExt(layers={Walls194276.structure,...},
    A={Walls194276.area,...},
    til={Walls194276.tilt,...},azi={Walls194276.azi,...}),

    nConExtWin=0,

    nConBou=1,datConBou(layers={Walls205734.structure}, A={Walls205734.area}, til={Walls205734.tilt}),

    nConPar=0,nSurBou=0,nPorts=1);

  PBIM.BIMPackage.Room Room2(...
    nConExt= 5,datConExt(layers={Walls194278.structure,...},
    A={Walls194278.area,...},
    til={Walls194278.tilt,...},
    azi={Walls194278.azi,...}),

    nConExtWin=0, nConBou=0,nConPar=0,
    nSurBou=1, surBou(A={Walls205734.area},
               absIR={Walls205734.structure.absIR_a}, absSol={Walls205734.structure.absSol_a},
               til={Walls205734.tilt}),

    nPorts=1);
```

Room instances

Figure 30. A code block of the generated *ModelicaBEM* presenting material objects for a wall objects, the wall objects, and room objects using the *Revit2Modelica* prototype.

5.3. Test Case 3: Adding Windows and an Interior Door

I expanded the two-thermal-zone model by installing two windows on the south and east exterior walls, respectively, and a door in the interior wall as shown in Figure 31a. The size of each window is 6 m^2, and the additional physical material information for the glazing system defined in Table 6 is prepared through the custom parameter window (Figure 31b).

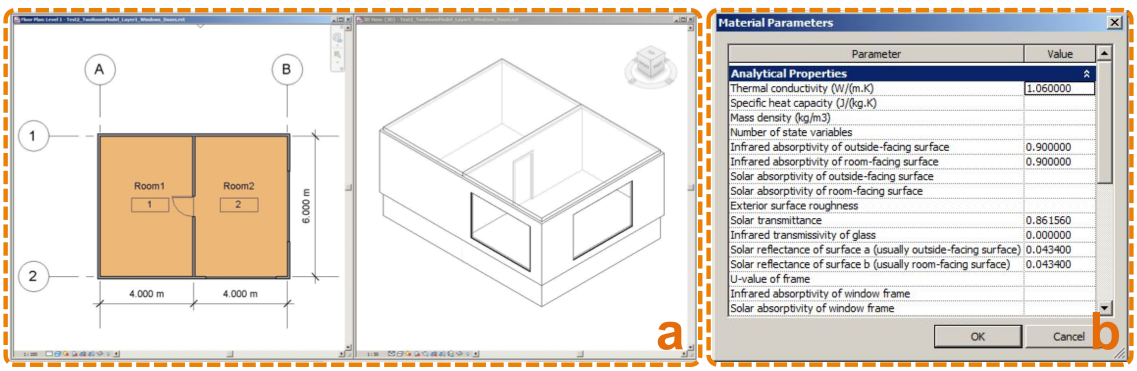


Figure 31. A two-thermal-zone Revit model with two windows and a door. (a) Floor-plan and isometric views. (b) The custom parameter window for adding additional physical properties for the glazing system.

To prepare the additional thermal information for the glazing system such as the ratio of window frame, I created new parameters in the existing *Window* family in Revit (Figure 32). The calculated value of the window frame ratio is used as a parameter of a window instance in the *ModelicaBEM* (fFraRatio of the Window instance in Figure 33).

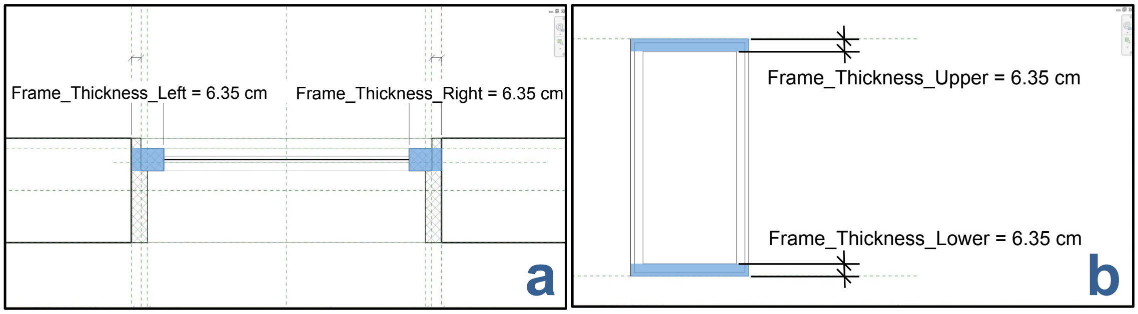


Figure 32. The Window family in Revit. (a) a plan view and (b) an elevation view.

By installing two windows in the right room, the room has three opaque surfaces (nConExt and datConExt in Room2 of Figure 33) and two opaque surfaces with windows (nConExtWin and datConExtWin in Room2 of Figure 33).

```
//Window information
PBIM.BIMPackage.Window Windows208788(
  areaWin=5.99981328229248,fFraRatio=0.000991968307418856, structureWin(glass={Glass208788},
  final nLay=1, UFra=3, absIRFra=0.8, absSolFra=0.5, haveInteriorShade=false,haveExteriorShade=false));
```

Window instance

```
//Door information
PBIM.BIMPackage.Door Doors210796(redeclare package Medium=MediumA, doorWidth=0.8636, doorHeight=2.032);
```

Door instance

```
//Room information
PBIM.BIMPackage.Room Room2(...
  nConExt=3, datConExt(layers={Walls206993.structure,R2.structure,F2.structure},
                       A={Walls206993.area,R2.area,F2.area},
                       til={Walls206993.tilt,R2.tilt,F2.tilt},
                       azi={Walls206993.azi,R2.azi,F2.azi}),

  nConExtWin=2,
  datConExtWin(layers={Walls194278.structure,Walls194279.structure},
                  A={(Walls194278.area*Windows208788.areaWin)/(5.99981328229248),
                     (Walls194279.area*Windows209156.areaWin)/(5.99981328229248)},
                  glaSys={Windows208788.structureWin,Windows209156.structureWin},
                  AWin={Windows208788.areaWin,Windows209156.areaWin},
                  fFra={Windows208788.fFraRatio,Windows209156.fFraRatio},
                  til={Walls194278.tilt,Walls194279.tilt},
                  azi={Walls194278.azi,Walls194279.azi}),

  nConBou=0,nConPar=0,

  nSurBou=1, surBou(A={Walls205734.area},
               absIR={Walls205734.structure.absIR_a},
               absSol={Walls205734.structure.absSol_a},
               til={Walls205734.tilt}),nPorts=3);
```

Room instance

```
//Connect information
connect(Doors210796.port_b1, Room1.Room_ports[2]);

connect(Doors210796.port_a2, Room1.Room_ports[3]);

connect(Doors210796.port_a1, Room2.Room_ports[2]);

connect(Doors210796.port_b2, Room2.Room_ports[3]);
```

Connect instances

Figure 33. A code block of the *ModelicaBEM* presenting a window object, a door object, and connect objects of a door generated by the *Revit2Modelica* prototype.

The line of the Modelica code for a door object is generated through *Revit2Modelica* (the Door instance in Figure 33) following the *Door* class definition in the wrapper classes. By wrapping a Door class, two Modelica *connects* need to be created to calculate bi-directional airflow between the two rooms. Therefore, four

Modelica connects are created to link the door and two rooms (Connect instances in Figure 33).

5.4. Test Case 4: Adding a New Story

To demonstrate a zoning case for vertical stacking of rooms, I created a two-story Revit model as shown in Figure 34. The building has one room on each floor and a 6 m^2 window on each of the south wall and the west wall at the first floor.

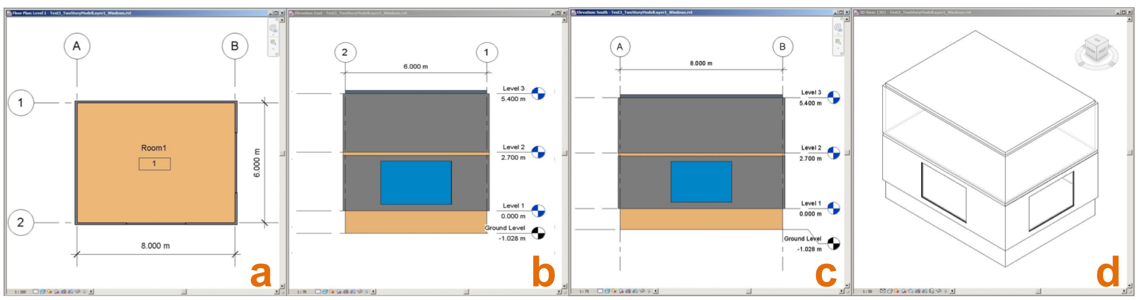


Figure 34. A two-story Revit model. (a) Floor-plan view of first level, (b) East-elevation view, (c) South-elevation view, and (d) Isometric view.

Based on the BIM of the building, *Revit2Modelica* creates *ModelicaBEM* that enables heat transfer to be simulated between two stories. The mechanism of the heat transfer is similar to Test Case 2. The major difference is that the floor object, instead of the interior wall, connects two thermal zones.

The roof instance in the lower level and the floor instance in the upper level are modeled in the *ModelicaBEM* as opaque surfaces (the Floor instance and the Roof instance in Figure 35). Then, a Modelica *connect* is created to link the opaque surfaces for conduction heat transfer calculation.

In terms of boundary conditions, the right room of Test Case 2 and the upper room in Test Case 4 have the same number of opaque surfaces (nConExt and datConExt) and another opaque surface between two thermal zones in each case (nSurBou and surBou). In Test Case 4, the five opaque surfaces in the upper room consist of four walls and a roof, and the one opaque surface between the lower room and the upper room is the floor object (Room instance in Figure 35).

```
//Floor information of upper level
  PBIM.BIMPackage.Floor F2(structure(material={Floors206671},final nLay=1),
                           area=48.0000000000001,tilt=3.14159265358979,azi=3.14159265358979);
```

Floor instance

```
//Roof information of lower level
  PBIM.BIMPackage.Roof F2206671(structure(material={Floors206671},final nLay=1),
                                area=48.0000000000001, tilt=0, azi=0);
```

Roof instance

```
//Room information of upper level
  PBIM.BIMPackage.Room Room2(...
    nConExt=5, datConExt(layers={Walls206727.structure,...,R1.structure},
                          A={Walls206727.area,...,R1.area},
                          til={Walls206727.tilt,...,R1.tilt},
                          azi={Walls206727.azi,...,R1.azi}),
    nConExtWin=0, nConBou=0,nConPar=0,
    nSurBou=1, surBou(A={F2.area},
                 absIR={F2.structure.absIR_a},
                 absSol={F2.structure.absSol_a},
                 til={F2.tilt}),nPorts=1);
```

Room instance

Figure 35. Generated *ModelicaBEM* code block of the two-story building model presenting the floor instance, the roof instance, and the room instance at the upper level.

As shown in Figure 30, Figure 33 and Figure 35, the generated three *ModelicaBEM* of the three test cases (Test Case 2, 3, and 4) demonstrate the use of the *Revit2Modelica* prototype. To validate the method and the prototype, I conducted simulation result comparisons explained blow.

5.5. Simulation Result Comparisons for Test Case 2, 3, and 4

I utilized Dymola as a Modelica development environment and LBNL Modelica Buildings library version 1.3 to perform thermal simulation with the *ModelicaBEM*. As Modelica Buildings requires designation of time intervals and tolerances, the simulation settings include time interval of 3600 seconds for a one-year period and a tolerance of 10^{-6}.

I applied the consistent model conditions for all the building models as follows:

- The floor is above the ground level.
- Each room is a single thermal zone.
- The building location is Chicago, Illinois, U.S.A.
- The building has no shading devices and no internal heat gains from equipment and occupants.
- The building has no HVAC systems.
- The windows and the door are closed.

The simulation results of each test case model agree with the results of each of the corresponding model created manually using the LBNL Modelica Buildings library sample structure by the team, in terms of annual indoor air temperature and heat flow.

The indoor air temperatures of each Modelica model in the test cases are almost identical with those of LBNL's models. For example in Test Case 3, as shown in Table 7, the highest temperatures are obtained at 12PM on August 21th in East Rooms in both models, and in Test Case 4, the lowest temperatures are obtained at 8AM on January 8th in the Upper Rooms in both models.

Table 7. Annual peak temperatures of the test cases.

Cases	Room name	Highest temperature (°C) / Date-Time	Lowest temperature (°C) / Date-Time
2	East	ModelicaBEM: 34.63 °C / July 19th - 6PM LBNL model: 34.66 °C / July 19th - 6PM	ModelicaBEM: -15.479 °C / February 7th - 8AM LBNL model: -15.471 °C / February 7th - 8AM
	West	ModelicaBEM: 35.222 °C / July 18th - 7PM LBNL model: 35.237 °C / July 18th - 7PM	ModelicaBEM: -15.655 °C / February 7th - 9AM LBNL model: -15.658 °C / February 7th - 9AM
3	East	ModelicaBEM: 42.827°C / August 21st - 12PM LBNL model: 42.968 °C / August 21st - 12PM	ModelicaBEM: -15.369 °C /February 7th - 6AM LBNL model: -15.669 °C / February 7th - 6AM
	West	ModelicaBEM: 35.485 °C / July 18th - 6PM LBNL model: 35.413°C / July 18th - 6PM	ModelicaBEM: -11.967 °C / February 7th - 8AM LBNL model: -11.955 °C / February 7th, 8AM

Table 7. Continued.

Cases	Room name	Highest temperature (°C) / Date-Time	Lowest temperature (°C) / Date-Time
4	Upper	ModelicaBEM: 38.085 °C / July 18th - 6PM LBNL model: 38.035 °C / July 18th - 6PM	ModelicaBEM: -21.289 °C / January 8th - 8AM LBNL model: -21.283 °C / January 8th - 8AM
	Lower	ModelicaBEM: 40. 144 °C / July 18th - 2PM LBNL model: 40.047 °C / July 18th - 2PM	ModelicaBEM: -10. 348 °C / February 7th - 6AM LBNL model: -10.029 °C / February 7th - 6AM

Figure 36, Figure 37, and Figure 38 show the results in Test Case 3: the indoor air temperatures of *ModelicaBEM* and LBNL's model, global horizontal radiation, and a dry bulb outdoor temperatures during different time periods and for different rooms from February 1st to 9th, from July 18th to 26th, and from August 14th to 22nd respectively.

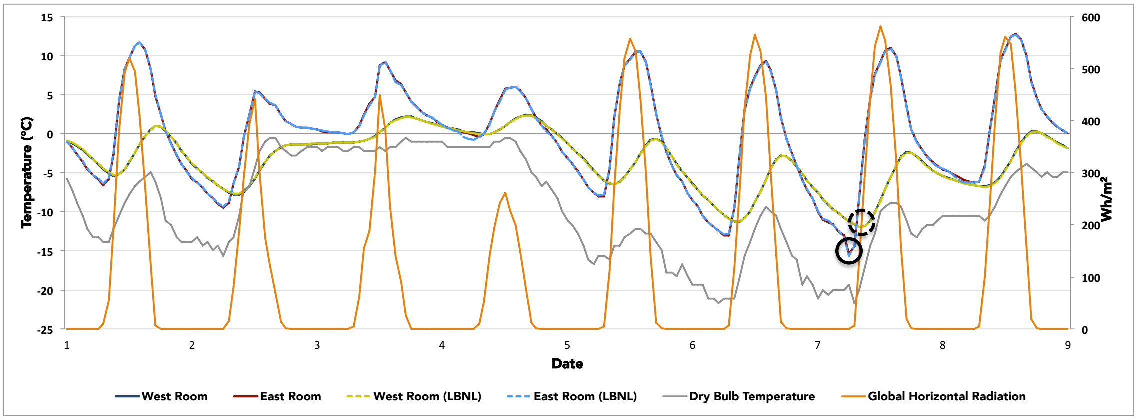


Figure 36. Temperature comparison of Test Case 3 for February 1st to 9th. Solid (6AM, February 7th) and broken (8AM, February 7th) circles point out the lowest temperature in the East and West Rooms respectively for the two models. The temperature curves overlap between the two models.

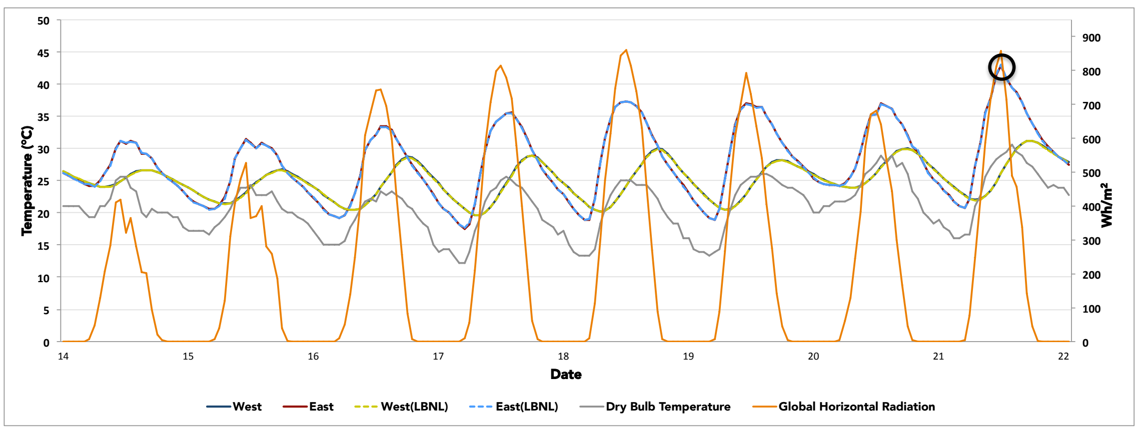


Figure 37. Temperature comparison of Test Case 3 for August 14th to 22nd. Solid circle (12PM, August 21st) points out the highest temperatures of the East Room. The temperature curves overlap between the two models.

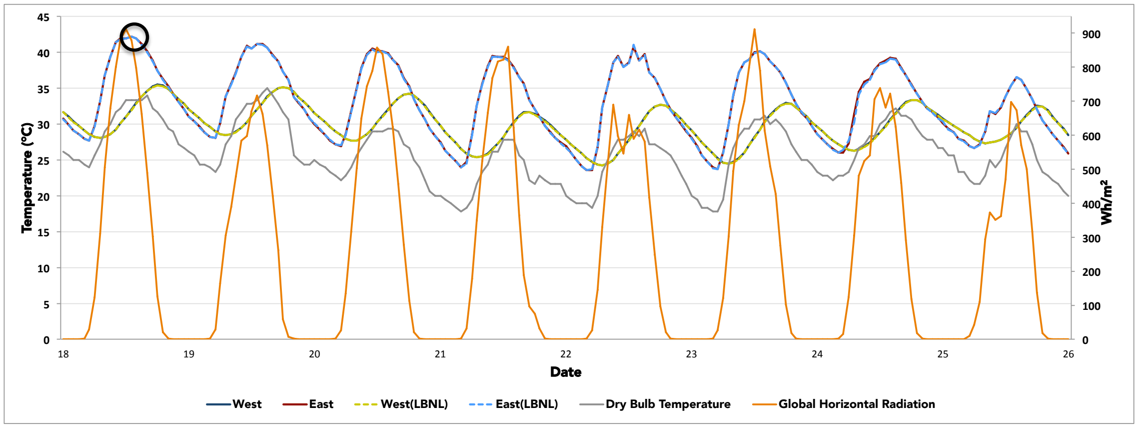


Figure 38. Temperature comparison of Test Case 3 for July 18th to 26th. Solid circle (6PM, July 18th) presents the highest temperatures of the West Room. The temperature curves overlap between the two models.

I also conducted a validation case study for component level analysis. I examined the temperatures from outside surfaces and temperatures of the inside surfaces of the east and south walls each having a window in Test Case 3. As shown in Figure 39 and Figure 40, the temperature graphs of *ModelicaBEM* almost overlap those of LBNL's model.

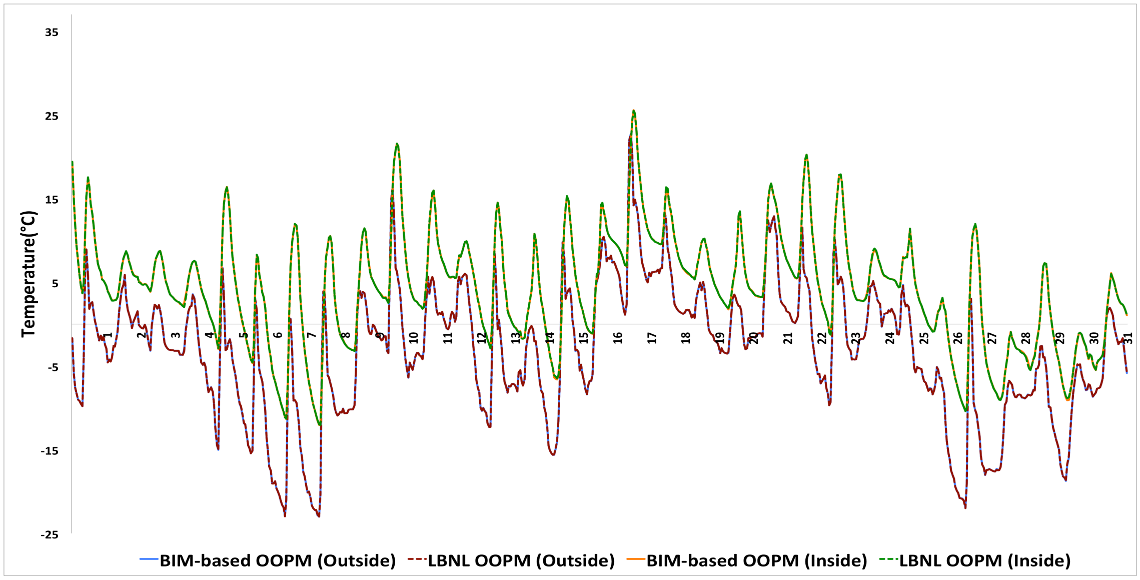


Figure 39. Temperature comparison between the two models for the east wall in Test Case 3 during January.

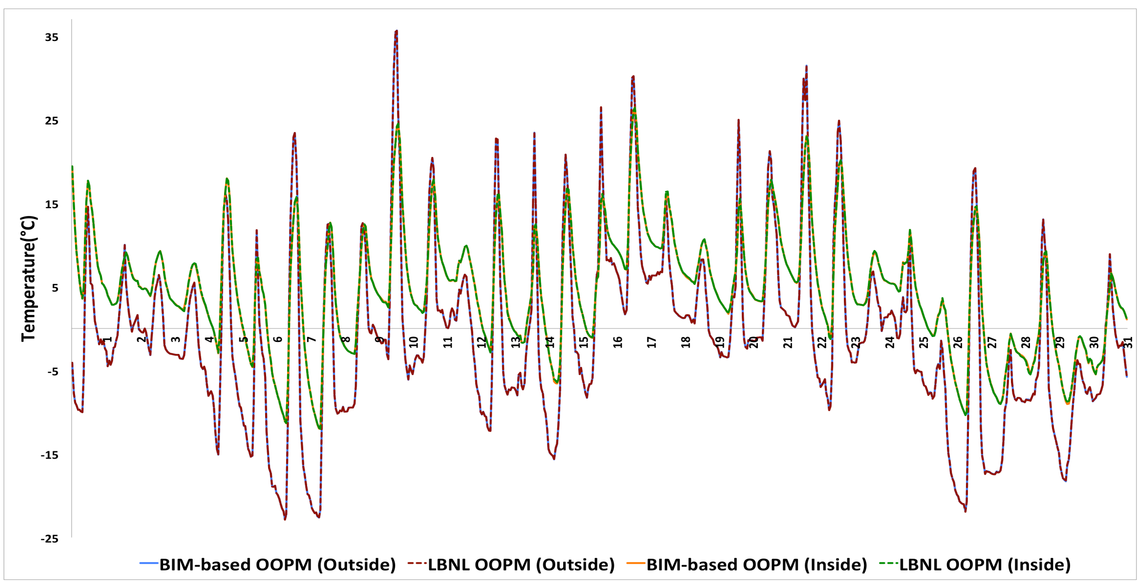


Figure 40. Temperature comparison between the two models for the south wall in Test Case 3 during January.

Overall, the BIM-based *ModelicaBEM* models created by the prototype produce very similar simulation results as LBNL's models. This is expected because the same thermal simulation algorithm provided by LBNL Modelica Buildings library is applied to all the energy models.

The modeling method (automatic versus manual) and model structures (one is BIM-based and the other is not) are the major differences between *ModelicaBEM* and the LBNL's models. The simulation results will have inconsistency during the comparison studies if the model translations were not done correctly by *Revit2Modelica*. *Revit2Modelica* generates the energy models that are reasonably more comprehensive to reflect the actual building configuration. In the example of Test Case 4, the LBNL's Modelica model represents the floor object as a roof object for the lower thermal zone and a floor object for the upper thermal zone. Based on the understanding of energy semantics, e.g., boundary condition, the shared floor object is modeled into thermal zones manually. However, our team's approach automatically splits the floor object as two components to represent the actual building configuration: one is for a floor object in the upper room and the other is for a roof object in the lower room.

In terms of the model structure, *Revit2Modelica* generated energy models present to the user architectural semantics such as rooms, instead of energy engineering-based semantics such as MixedAir. In addition, the two structures can have a difference regarding the order of building enclosure elements used as arguments in thermal calculation functions, causing the slightly different simulation results. The result differences due to this argument order difference can be reduced through decreasing the model tolerance value in Dymola.

6. APPLICATION

A visualization application is developed as a sample to demonstrate the applications of the *BIM2BEM* framework in the design process. The application enables visualizing building energy simulation results in BIM at both the building level and the component level, for designers to understand the relationship between design decisions and the building performances. The application lets architects use BIM as a common user interface for building design and performance visualization.

6.1. Introduction

A lack of information visualization after conducting a Building Energy Simulation (BES) may reduce the effective use of simulation results in developing schematic designs. I investigated the *BIM2BEM* framework for visualizing building energy performance, especially thermal energy, in Building Information Modeling (BIM) by color coding the various components of the building based on their energy flow values. First, I created a BIM model for BESTEST Case 600 (ANSI/ASHRAE Standard, 2010), using Autodesk Revit as a test case. Second, I created an Energy Performance Indicator (EPI) parameter for each building component in order to store the building energy performance results produced by the *BIM2BEM*. Third, I designed a visibility setting to color code the building components based on their energy flow values. Lastly, I developed the Building Energy Performance Visualization (BEPV) by using Revit's API, which automatically loads the BES results into the EPI parameter of each component and changes the building components' colors. The effect looks like an infrared time-lapse animation of the building.

The *BIM2BEM* framework: (1) uses the EPI parameter in BIM to indicate the results of the building's energy performance, (2) displays a series of energy performance results as color-coded building components, and (3) enables architects to utilize BIM as a common user interface for architectural modeling and BEPV.

The following sections present the details of the research methods, their implementation. Findings with regards to interfacing the BIM and BES results for this type of visualization will be discussed in the chapter Conclusions and Future Work.

6.2. Problems and Challenges

BES tools have become an important wellspring of support for sustainable design. Most current thermal simulation applications require the acquisition of building geometry from computer-aided design tools (Bazjanac 2001) or the exchange of BIM data with thermal modeling (Mitchell et al. 2007). Interoperability between these software systems remains a problem. Some BES tools support a level of integration with BIM; for example, Autodesk Green Building Studio can be used with Revit in the simulation process. In addition, research efforts to visualize energy simulation results (Sreshthaputra, Haberl, and Andrews 2004; Haberl and Akleman 2010) are motivated by an interest in achieving a better understanding of building energy performance analysis. However, the feedback provided by BES tools has, historically, been difficult to apply to the design process. Although energy performance results typically are displayed as a series of tables and numbers, it often can be difficult to discern the relationships among the simulation results and the building components. Such relationships are important for understanding, which components perform well and which do not, information that is essential to improving the design.

6.3. Goal and Approach

The goal of this application development was to enhance sustainable design by better informing design decisions through a use of BES results. To achieve this goal, I developed an integrated environment *BIM2BEM* that combines BIM and the results produced by object-based BES tools, thus integrating the capability of BES visualizations into BIM.

The method combines the advantages of BIM and OOPM to produce a visualization that combines an object-based modeling approach and object-oriented programming methods. The object-based modeling approach allows for direct access of the BIM and simulation results, and the Object-Oriented programming methods facilitate the development of a system interface that produces a useful visualization. Figure 41 shows the overall process of this research.

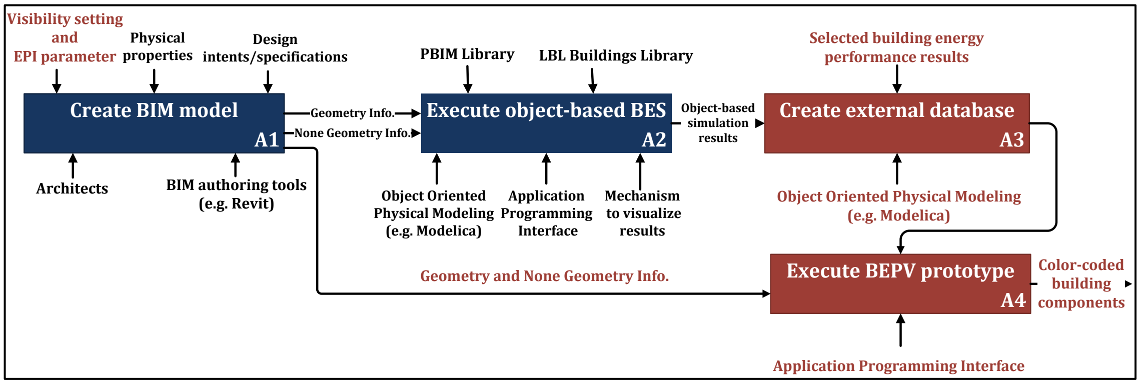


Figure 41. Overall process diagram using IDEF0. The dark blue components represent the thermal simulation prototype of the PBIM project; the dark red shows the BEPV workflow.

6.4. Method

The method involves creating a method to prepare a BIM to hold the energy simulation results for each component, creating a method for changing the visualization of the BIM based on those energy simulation results, writing a system interface through Revit's Application Programming Interface (API) for BEPV, and conducting a case study with a benchmark energy model to verify the operation of the prototype software.

6.4.1. Preparing the BIM

Before executing BEPV, the Revit model was preprocessed to include a parameter for the EPI and visualization settings to change the appearance of the model. The EPI is intended to be one of many performance indicators that can be generated by the BES. The values in the prototype represent a heat flow rate. The values of the EPI can range widely based on the kind of BEP results that need to be visualized. In a more advanced prototype, the EPI could be a function of multiple energy performance parameters, and users could select among many performance parameters for their EPI visualization.

Using Revit commands, I created the EPI parameter as a project parameter so that all of the building's components could include the information. Figure 42 illustrates how I created the EPI parameter in Revit as a project parameter of the type Number, and how I applied it to those building components that affect energy performance.

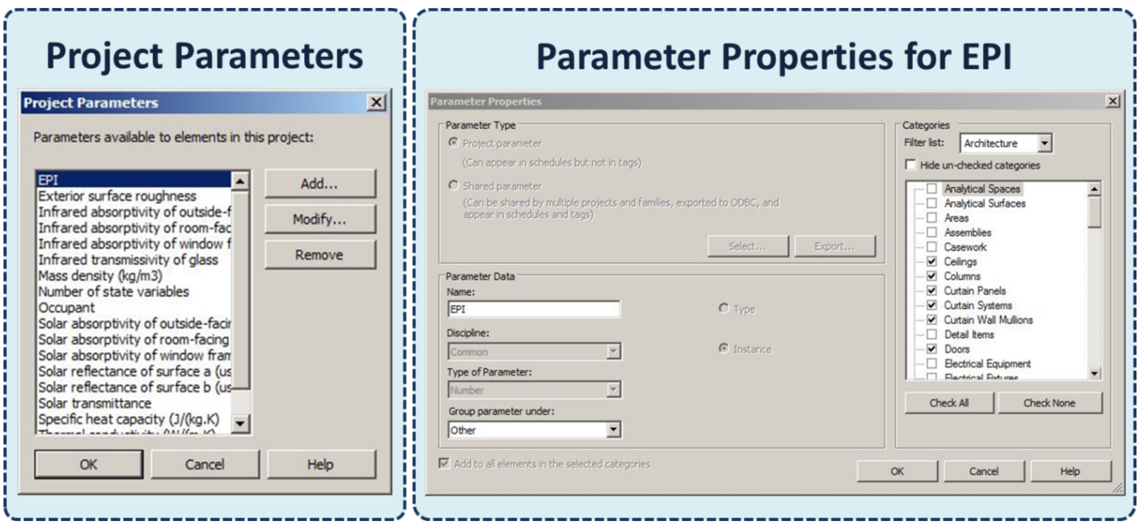


Figure 42. EPI parameter creation used to communicate BEP results in Revit.

After creating the EPI parameter, users can check the assigned parameter by selecting the building components (Figure 43).

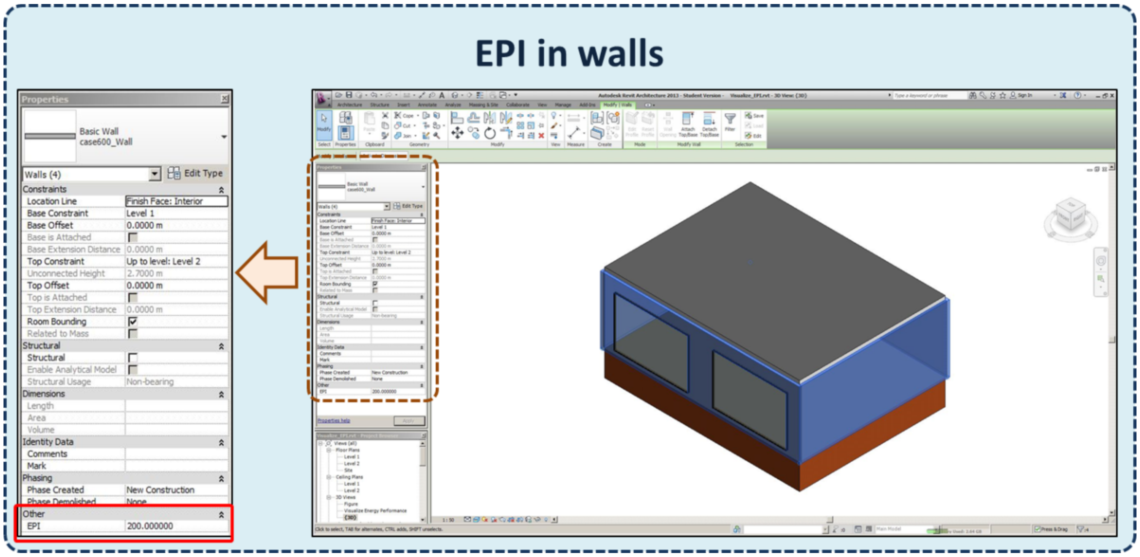


Figure 43. EPI parameter applied to wall components.

Visibility settings can then be used to create color-coded building components based on the updated values in the EPI parameter. The settings are a series of ranges, and each range has a different color. For example, a range from 101 to 130 Watts can be assigned red as its color; if the value of the EPI parameter is within the proper range, the component containing the proper value will be visualized in red. As shown in Figure 44, I used Revit commands to create a rating system by using the Filters dialog in the *Visibility/Graphic Overrides* window. The rating system has ten levels with different corresponding colors, and each level is divided into thirty increments. For example, the values in the 10^{th} level are between 301 and 330 Watts.

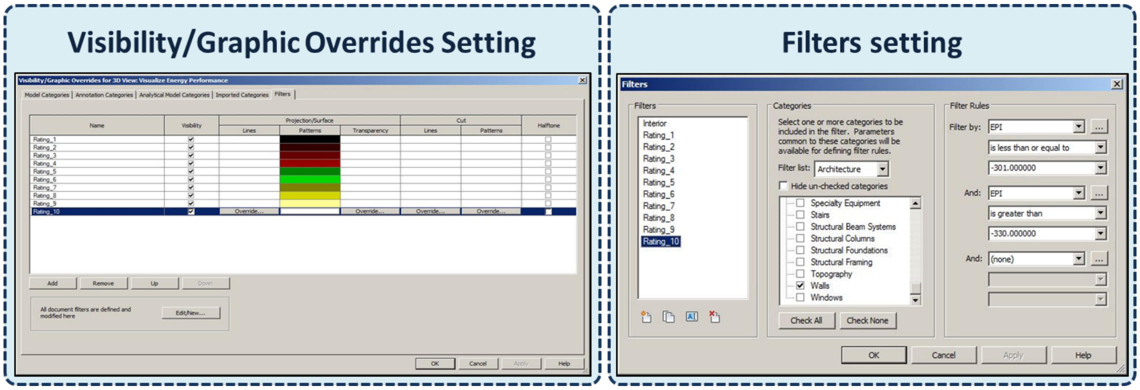


Figure 44. Visibility setting used to create color-coded building components.

6.4.2. Visualization Commands

The final step in developing the prototype was to write software to feed the information from the BES tool into the EPI parameter. When the value of the EPI parameter is updated by the software, the visibility settings change the color of the building components for the simulation time and based on the EPI values.

The thermal simulation results are then generated by the BES tool for a designated time interval and step for each building component. An export function in the BES tool produces a CSV format file that can be imported into Microsoft Excel. Sample results have been charted, as shown in Figure 45. The left and right graphs represent heat flow rates through opaque surfaces and annual indoor temperatures respectively. The results can be exported to a CSV format file by the export function in the BES tool.

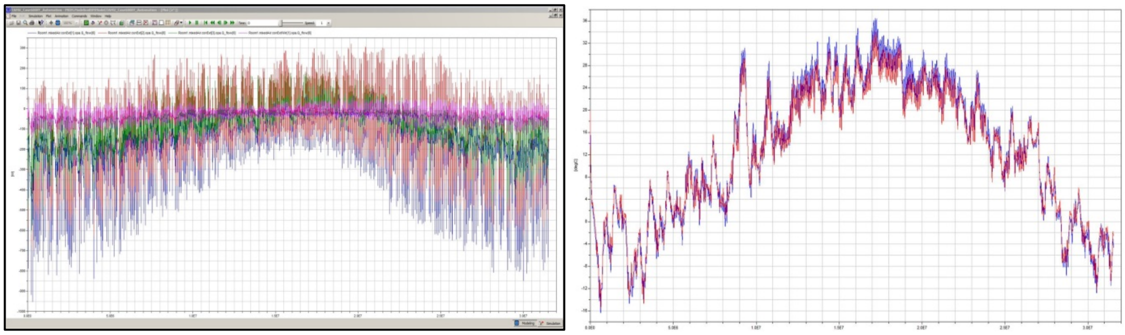


Figure 45. Object-based BES results from OOPM.

The visualization command checks each building component and saves its identification number. Using the identification number, a command retrieves the simulation results from the database and stores them in the corresponding EPI parameters in Revit. Once the building components have their EPI values, they are represented as their designated colors by automatically applying the visibility setting described above. Figure 46 shows the workflow diagram of the BEPV. After the BEPV execution, the prototype continuously updates the color of the building components according to the simulation results.

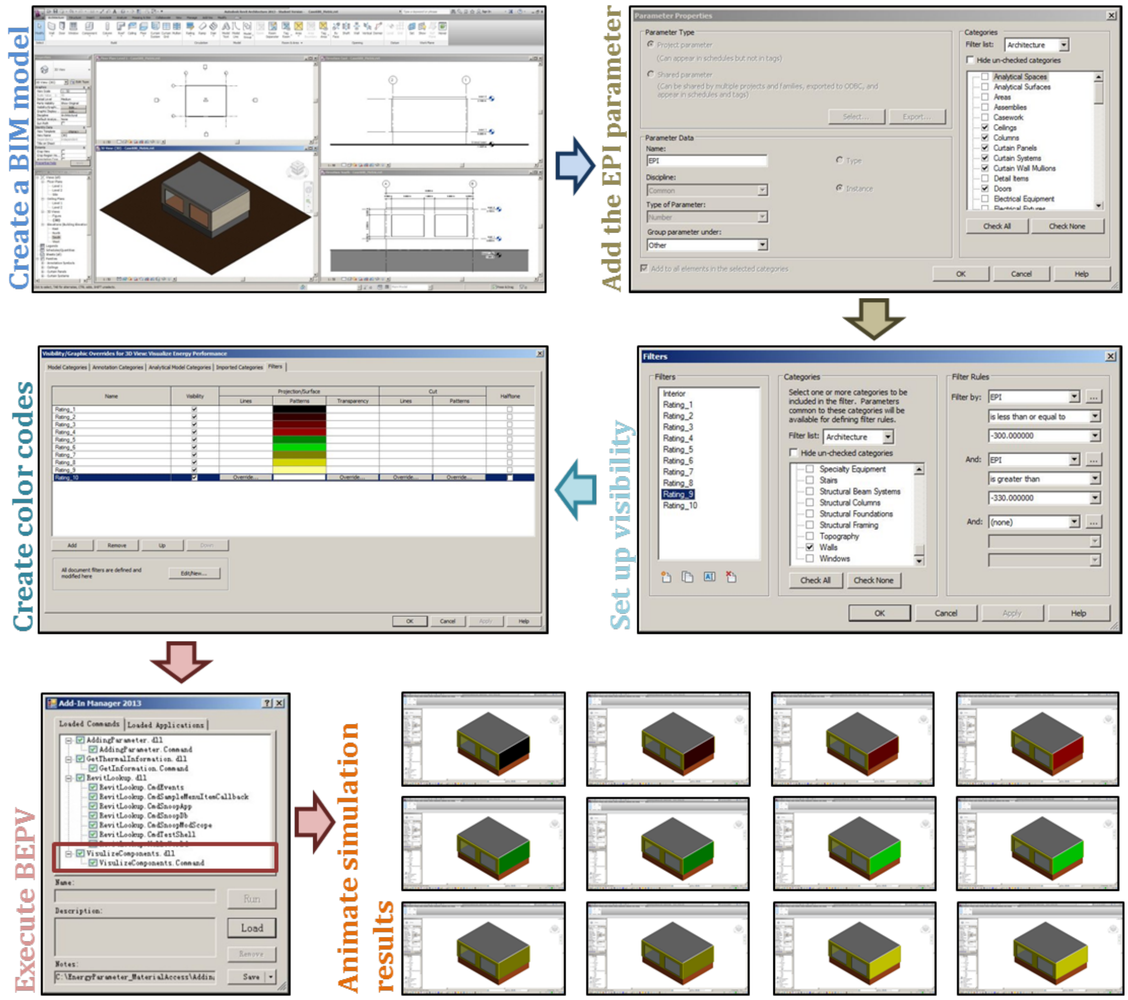


Figure 46. Workflow of BEPV.

6.5. Verification

I conducted a case study to verify the operation of the prototype using BESTEST Case 600 (Figure 47). The case study visualizes one of the thermal simulation results (heat flow rate) in the wall components. After creating the BIM model, I added the EPI parameter into the project parameters (Figure 42 and Figure 43). As shown in Figure 42, all wall components were provided with BES results through the EPI parameter. Then, I executed the object-based BES tool to generate object-based thermal simulation results. An explanation of how I executed the BES tool from BIM was presented in the previous research (Jeong et al. 2014a). The values obtained from the graphs in Figure 45 were then translated into a CSV file by the export function in Dymola.

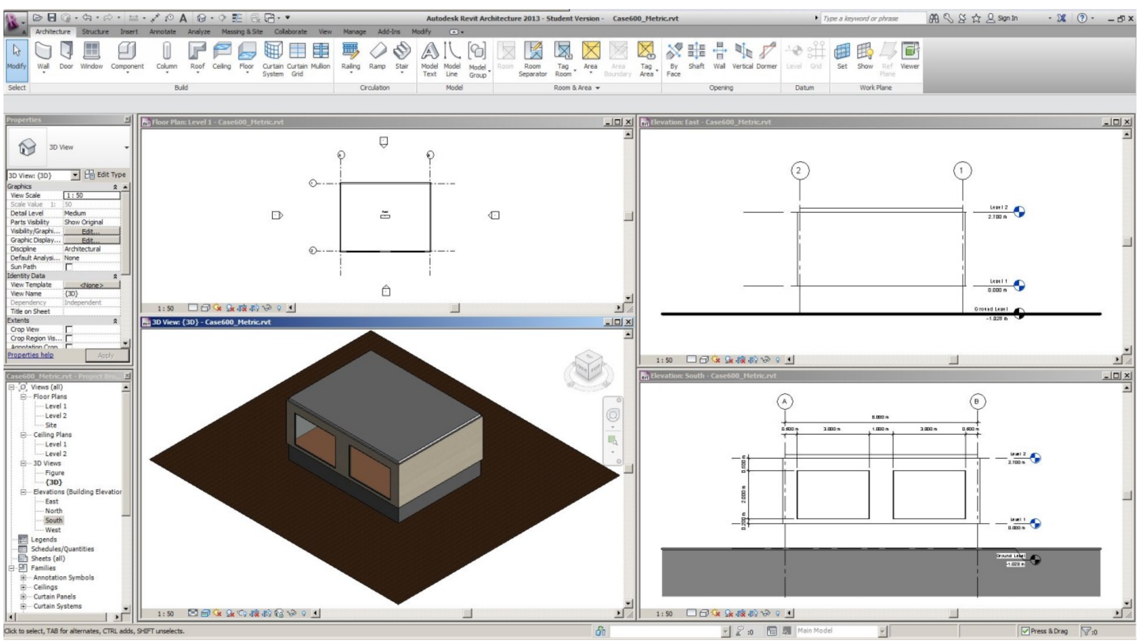


Figure 47. BESTEST Case 600 Revit model.

After generating an external database for the information regarding heat flow rate, I performed a BEPV to produce a one-year thermal simulation. While the database provided 26,283 different time snap-shots for each wall, I only retrieved twelve for each wall for the visualization. The BEPV began the visualization at the north wall, and continued it to the west wall, the south wall, and the east wall, sequentially. Table 8 shows the visualization of the EPI values for one year; each column shows the conductive heat flow through the designated opaque wall according to the range of colors selected in the visibility setting. On the west wall, the dark red color indicates a lower level of conductive heat loss than walls highlighted with the light green color.

Table 8. BEPV execution in BIM and the corresponding EPI values.

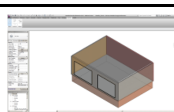
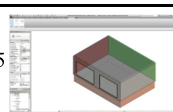
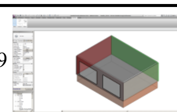
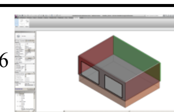
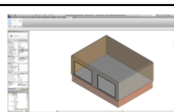
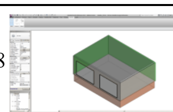
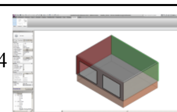
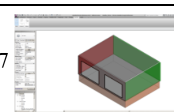
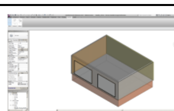
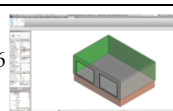
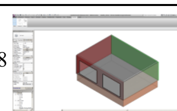
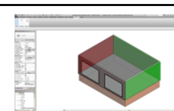
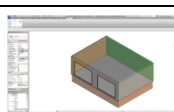
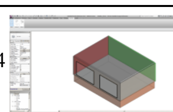
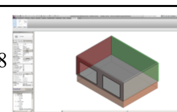
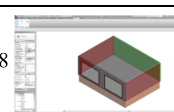
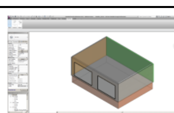
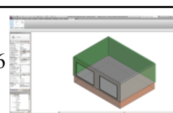
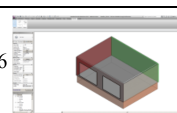
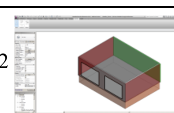
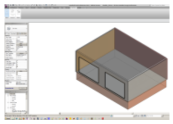
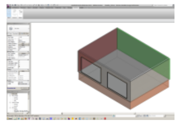
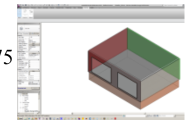
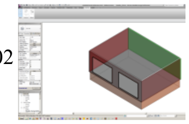
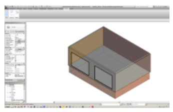
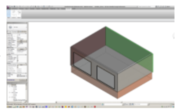
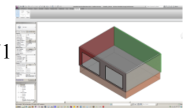
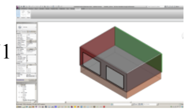
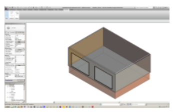
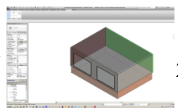
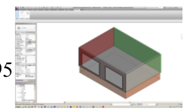
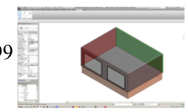
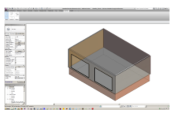
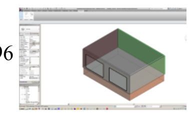
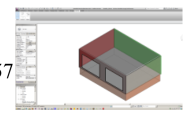
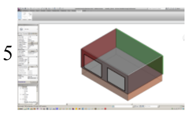
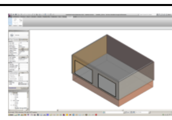
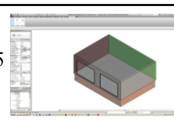
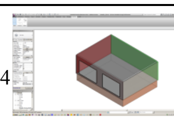
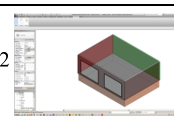
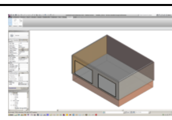
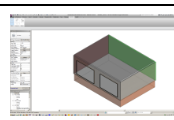
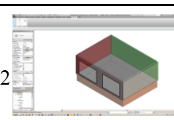
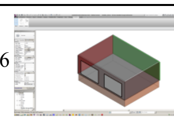
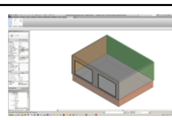
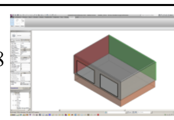
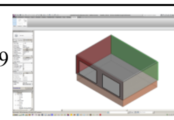
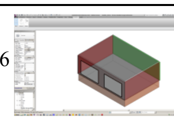
	North wall		West wall		South wall		East wall	
Simulation Day and Time (12:00AM)	BEPV	EPI values (Watts)	BEPV	EPI values (Watts)	BEPV	EPI values (Watts)	BEPV	EPI values (Watts)
1st day		-121.985		-89.1739		-27.4206		-90.982
25th day		-191.708		-137.254		-41.4957		-137.747
50th day		-233.326		-164.698		-50.05		-168.532
75th day		-183.664		-127.868		-38.6838		-127.864
100th day		-178.486		-133.836		-39.4182		-129.839

Table 8. Continued.

	North wall		West wall		South wall		East wall	
125th day		-99.7078		-74.075		-22.3002		-72.7548
150th day		-55.3042		-34.371		-9.83371		-39.9014
175th day		-29.4882		-20.8895		-4.01899		-15.3424
200th day		-32.6796		-17.2257		-4.52015		-19.5339
225th day		-31.8205		-17.1274		-4.51702		-19.57
250th day		-29.754		-15.9262		-4.19966		-21.5423
275th day		-171.188		-119.39		-36.0586		-123.102

The values of the heat flow rates were calculated in watt, and stored as EPI parameters in BIM. The heat flow rates at midnight were positive in the summer and negative in the winter, as shown in Figure 45, where the positive values indicate that heat flows are from the outside to the inside of a wall and the negative values indicate a reversed heat flow, from inside to outside. Since I visualized the values for the first hour, midnight, for each day, Table 8 shows all negative values.

Based on a visibility setting in which the range for each level is a 30 Watt interval and is assigned a color, if the EPI value for a wall falls within a certain level, the wall can be represented as a color representing that level at that certain time. When the time changes, the color may change according to the actual heat flow rate. For example, the north wall was represented by a dark orange color at 12:00AM on the 175th, 200th, 225th, and 250th days because the values for those days fell within in the same level. In addition, Table 8 shows which wall was critical to the heat flow on a specific date. The 275th day measurements of the north, west, south, and east walls are represented by green, dark red, black, and dark red, respectively.

7. CONCLUSIONS AND FUTURE WORK

In this research, I have developed the *BIM2BEM* framework through a MVD methodology and a system interface development. The application of *BIM2BEM* was demonstrated by visualizing energy performance in BIM. The framework allows the model translation between BIM and BEM that can automatically translate BIM to *ModelicaBEM* and run the thermal simulation. The simulation results can be visualized in BIM through the application to support architect's decision making. The developed framework was experimented and validated through different test cases to cover various building components.

In the following sections, I describe the conclusions, findings, and future work of the research in terms of MVD and the framework developments as well as the visualization application.

7.1. MVD for Model Translation

This study presented a translation method for integrating BIM and OOPM (Modelica) for building energy simulation. The MVD development enables interdisciplinary data exchange between architectural design and building energy simulation. The developed system interface based on the MVD can leverage the consistent use of the architect's data (such as the building geometry, materials, and even parametric objects) in building energy simulation without re-creating them in energy models manually. Reuse of the data from BIM can significantly reduce the effort required for the definition of input data in BEM. The process presented in this study has the potential to eliminate error-prone manual processes.

The data modeling approach facilitates the development of a system interface for automatic translation from BIM to BEM with high efficiency and accuracy. While the file-based translation through standard schema such as IFC and gbXML can often facilitate the translation between different BIM tools and different simulation applications, implementing the complex schemas of IFC demands enormous amount of time and efforts (Dong et al. 2007). The developed prototype based on the investigated Exchange MVD enables more seamless design-simulation integration while the BIM tools (such as Revit) can preserve the parametric modeling capability in the process. The current version of the MVD is applicable for Revit; however, the MVD and the system interface can be developed to support other BIM tools such as ArchiCAD, AECOsim Building Designer V8i, and Allplan. Nevertheless, the use of IFC can better bridge between multiple BIM authoring tools and diverse simulation tools.

A major advantage of our approach is that Modelica is supported by a growing community of researchers who are developing various physics-based modules for simulation. The approach is generalizable to integration of BIM to other physics-based simulations. Currently, the MVD approach is focused on thermal simulation. In future work, I will expand MVD to cover more simulation domains including daylight and photovoltaic. Moreover, I will apply the prototype to test more building types including complex buildings to enhance the MVD translation method and collect measured data from real-world project to validate the MVD approach. I will also examine more general boundary condition generating methods, e.g. (Rose and Bazjanac 2013), and apply them into the system interface of *BIM2BEM* framework.

7.2. *BIM2BEM* Framework

The interfacing framework demonstrated the use of BIM as a common user interface for architectural design and building thermal simulation. The automatic creation of energy models from BIM will reduce the interoperability problem that exists between architectural design models and energy simulation models. BIM facilitates building objects to be maintained for data integrity between the two models. The automatic simulation process, after creating energy models, provides the whole building's as well as its individual elements' thermal performance analyses. Object-based simulation results enabled by OOPM and displayed in BIM allow architects to be better informed about design options and their performances in order to improve the design.

The prototype facilitates an automatic translation from BIM to energy models with high efficiency and accuracy. The strategy to map BIM (e.g. Revit models) to OOPM (*ModelicaBEM*) is proven to be successful. The strategy is faster to implement than using a legacy thermal modeling library (e.g. the ASHRAE Toolkit for Building Load Calculations (Pedersen et al. 2001) written in a procedural programming language. The PBIM research team has also investigated and experimented the Toolkit for BIM to thermal translation). This study shows that the use of the two Object-Oriented Modeling methods (BIM and Modelica-based energy modeling) is an effective approach to map building design models to energy models, and improve interoperability in general.

The *BIM2BEM* framework will be further developed to cover diverse building topologies, complex building geometry, and multi-domain simulations. The goal is to create a user-friendly and powerful design process, in which BIM will act as a user interface for the design, simulation, and visualization of sustainable buildings. This design process will be supported by further research and development of system interfaces between BIM and simulation/visualization applications.

In the current version of the *Revit2Modelica* prototype, the users are expected to assign additional material parameters in a BIM model. In the future development, assigning materials will be linked to existing material databases such as the one in the International Building Physics Toolbox (Kalagasidis 2003; Nielsen et al. 2002; Kalagasidis et al. 2007) or the specific computational building design and simulation tool (SEMPER) (Mahdavi 1999; Lam et al. 2002).

In defining the building topology in Revit, the prototype uses a manual object naming method for helping *Revit2Modelica* to recognize the room-floor-roof relationship. Further algorithm development will automatically recognize this kind of relationships among building objects.

Currently, the framework does not include the automatic generation of Modelica graphical representation, which is a powerful visualization and editing tool for BEM and therefore will be investigated in the future.

Based on that a fully populated BIM model may include a very large dataset with unnecessary details for BEM (Bazjanac 2008), I will adopt the guideline for converting BIM to BEM regarding data subtraction, addition, translation, and derivation (Clayton et al. 2013).

Finally, I will apply the framework and the prototype developed to more building types and collect measured data from real-world projects for enhancing and validating *ModelicaBEM*.

7.3. Visualization of Building Energy Performance in BIM

The implemented approach demonstrated a new method of visualizing building energy performance for BIM users. Use of this method may benefit sustainable building design by illustrating building components' thermal performances within the BIM authorizing environment. In addition, the prototype showed the capability of BIM as a common user interface that enables architects to check BES results through visualized building components explicitly for building design. Moreover, the method allows architects to identify the building components that are critical to energy efficiency by investigating color coded building components. The prototype illustrated how architects would be better informed to improve their sustainable design.

I expect to further investigate the integration of more object-based simulation results (e.g., infiltration, indoor air temperature, and solar radiation) into BIM. Furthermore, I plan to conduct a user study regarding how designers can become better informed from using the visualization prototype.

REFERENCES

Aksamija, A. 2010. "Analysis and Computation: Sustainable Design in Practice." *Design Principles and Practices*: *An International Journal* 4 (4): 291-314.

Aksamija, A. 2012. BIM-Based Building Performance Analysis: Evaluation and Simulation of Design Decisions. *Proceedings of the 2012 ACEEE Summer Study on Energy Efficiency in Buildings*, Pacific Grove, California. 12-1-12-12.

ApacheSim. 2014. "ApacheSim User Guide <Virtual Environment> 6.0." Accessed August 5, 2014. http://www.iesve.com/downloads/help/Thermal/ApacheSim.pdf.

ASHRAE. 2010. *ANSI/ASHRAE Standard 140-2007, Standard Method of Test for the Evaluation of Building Energy Analysis Computer Programs*. Atlanta, GA: American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc.

Attia, S. 2011. "State of the Art of Existing Early Design Simulation Tools for Net Zero Energy Buildings: A Comparison of Ten Tools." Technical report. Louvain La Neuve, Belgium: Université catholique de Louvain.

Attia, S., L. Beltrán, A. D. Herde, and J. Hensen. 2009. Architect Friendly: A Comparison of Ten Different Building Performance Simulation Tools. *Proceedings of Building Simulation Conference*, Glasgow, Scotland. 204–11.

Autodesk Inc. 2014a. "Revit Architecture - Building Design Software - Autodesk." Accessed August 5, 2014. http://usa.autodesk.com/revit-architecture.

Autodesk Inc. 2014b. “Autodesk - Developer Center - Autodesk® Revit® Architecture, Autodesk® Revit® Structure and Autodesk® Revit® MEP,” Accessed August 5, 2014. http://usa.autodesk.com/adsk/servlet/index?siteID=123112&id=2484975.

Bazjanac, V. 2001. Acquisition of Building Geometry in the Simulation of Energy Performance. *Proceedings of Building Simulation Conference*, Rio de Janeiro, Brazil. pp. 305-311.

Bazjanac, V. 2008. “IFC BIM-Based Methodology for Semi-Automated Building Energy Performance Simulation.” Lawrence Berkeley National Laboratory, Berkeley, CA, USA.

Bazjanac, V. 2009. Implementation of semi-automated energy performance simulation: building geometry. *Proceedings of CIB W,* Istanbul, Turkey, 78: 595–602.

Bazjanac, V., and A. Kiviniemi. 2007. Reduction, Simplification, Translation and Interpretation in the Exchange of Model Data. *Proceedings of CIB W,* Maribor, Slovenia, 78: 163–68.

Bazjanac, V. and T. Maile. 2004. “IFC HVAC interface to EnergyPlus-A case of expanded interoperability for energy simulation,” Lawrence Berkeley National Laboratory, Berkeley, CA, USA.

Bazjanac, V., T. Maile, C. Rose, J. T. O’Donnell, N. Mrazović, E. Morrissey, and B. R. Welle. 2011. An Assessment of the Use of Building Energy Performance

Simulation in Early Design. *Proceedings of Building Simulation Conference*, Glasgow, Sydney, Australia. pp. 1579-1585

Bentley System. 2014. “Building Information Modeling - Bentley Architecture.” Accessed August 5, 2014. http://www.bentley.com/en-US/Products/Bentley+Architecture.

Clarke, J. 2001. *Energy Simulation in Building Design*. Adam Hilger, Bristol, England.

Clayton, M., J. Haberl, W. Yan, S. Kota, F. Farias, W. Jeong, J. B. Kim, and J. L. Bermudez Alcocer. 2013. “Development of a Reference Building Information Model (BIM) for Thermal Model Compliance Testing.” RP-1468, ASHRAE.

Crawley D.B., L. K. Lawrie, F. C. Winkelmann, W. F. Buhl, Y. J. Huang, C. O. Pedersen, R. K. Strand, R. J. Liesen, D. E. Fisher, M. J. Witte, and J. Glazer. 2001. “EnergyPlus: creating a new-generation building energy simulation program.” *Energy and Buildings* 33(4): 443–457.

Crawley, D. B., J. W. Hand, M. Kummert, and B. T. Griffith. 2008. “Contrasting the Capabilities of Building Energy Performance Simulation Programs.” *Building and Environment* 43 (4): 661–73.

DesignBuilder Software. 2014. “DesignBuilder - Building design, simulation and visualisation - Building Simulation... Made Easy.” Accessed August 5. 2014. http://www.designbuilder.co.uk.

Dassault Systèmes. 2014. “Multi-Engineering Modeling and Simulation - Dymola - CATIA - Dassault Systèmes.” Accessed August 5, 2014. http://www.3ds.com/products-services/catia/portfolio/dymola/overview.

Dong, B., K. P. Lam, Y. C. Huang, and G. M. Dobbs. 2007. A comparative study of the IFC and gbXML informational infrastructures for data exchange in computational design support environments. *Proceedings of the 10th International IBPSA Conference*, Beijing, China. pp. 1530-1537.

Eastman, C. M., P. Teicholz, R. Sacks, and K. Liston. 2008. *BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors*. Wiley, NY.

Elmasri, R., and S. Navathe. 2010. *Fundamentals of Database Systems, 6th ed.* Addison-Wesley, NY, USA.

Fritzson, P. 2010. *Principles of Object-Oriented Modeling and Simulation with Modelica 2.1*. Wiley, NY, USA.

Fritzson, P., and P. Bunus. 2002. Modelica-A General Object-Oriented Language for Continuous and Discrete-Event System Modeling and Simulation. *Proceedings of the 35th Annual Simulation Symposium*, 35:365–80.

General Services Administration. 2014. “3D-4D Building Information Modeling.” Accessed August 5, 2014.

http://www.gsa.gov/portal/content/105075?utm_source=PBS&utm_medium=print-radio&utm_term=bim&utm_campaign=shortcuts.

Good, J., A. Frisque, and D. Phillips. 2008. The Role of Wind in Natural Ventilation Simulations Using Airflow Network Models. *Proceedings of the Third National Conference of IBPSA-USA*, Berkeley, California. pp. 140-147.

GRAPHISOFT. 2014. “About ArchiCAD – A 3D CAD software for architectural design & modeling.” Accessed August 5, 2014. http://www.graphisoft.com/products/archicad.

Gratia, E., and de A. Herde. 2002. “A Simple Design Tool for the Thermal Study of an Office Building.” *Energy and Buildings* 34: 279-289.

Haberl, J. S. 2008. “SIMBUILD Survey: Academic Use of Simulation Software” Invited presentation at the 3rd National Conference of IBPSA-USA, Berkeley, California, July 30-August 1.

Haberl, J. S., and E. Akleman. 2010. “Demonstration of the Use of Multimedia Electronic Information Enhancements for a Chapter Handbook CD-ROM: 3D Modeling and animation. (Report)” *ASHRAE Transactions*.

Hand, J. W., D. B. Crawley, M. Donn, and L. K. Lawrie. 2005. Improving the data available to simulation programs. *Proceedings of Building Simulation Conference*. pp. 373–380.

International Alliance for Interoperability (IAI). 2014. "IFC Model View Definition," Accessed August 5, 2014. http://www.blis-project.org/IAI-MVD/.

IESVE. 2014. "<Virtual Environment> - Integrated Environmental Solutions." Accessed August 5, 2014. http://www.iesve.com/software/ve-pro/tutorial-files/ies_ve_tutorial.pdf.

ISO TC 184/SC 4. 1994. "ISO 10303-11: Industrial automation systems and integration- Product data representation and exchange- Part 1: Overview and fundamental principles." International Organization for Standardization, Geneva, Switzerland.

Jeong, W., J. Kim, M. J. Clayton, J. S. Haberl, and W. Yan. 2013. Visualization of Building Energy Performance in Building Information Models. *Proceedings of The Association for Computer-Aided Design in Architecture) 2013 International Conference*, Cambridge, Ontario, Canada. pp. 87-92.

Jeong, W., J. Kim, M. J. Clayton, J. S. Haberl, and W. Yan. 2014a. "A Framework to Integrate Object-Oriented Physical Modelling with Building Information Modelling for Building Thermal Simulation." under-reviewed in *Journal of Building Performance Simulation*.

Jeong, W., J. Kim, M. J. Clayton, J. S. Haberl, W. Yan. 2014b. "Translating Building Information Modeling to Building Energy Modeling using Model View Definition." in press in *The Scientific World Journal*.

Judkoff, R., and J. Neymark. 1995. “International Energy Agency Building Energy Simulation Test (bestest) and Diagnostic Method”. NREL/TP--472-6231. National Renewable Energy Lab., Golden, Colorado, U.S.

Kalagasidis, A. S. 2003, “HAM-Tools, International Building Physics Toolbox, Block documentation.” Report R-02:6, Department of Building Technology, Chalmers University of Technology, Gothenburg, Sweden.

Kalagasidis, A. S., P. Weitzmann, T. R. Nielsen, R. Peuhkuri, C. Hagentoft, and C. Rode. 2007 "The international building physics toolbox in Simulink." *Energy and Buildings* 39(6): 665-674.

Kim, J., W. Jeong, M. J. Clayton, J. S. Haberl, and W. Yan. 2014. “Developing a Physical BIM Library for Building Thermal Energy Simulation.” under-reviewed in *Automation in Construction Journal*.

Kota, S., J. S. Haberl, M. J. Clayton, and W. Yan. 2014. “Building Information Modeling (BIM)-Based Daylighting Simulation and Analysis.” *Energy and Buildings* 81: 391–403.

Lam, K. P., A. Mahdavi, S. Gupta, N. H. Wong, R. Brahme, Z. Kang. 2002. “Integrated and distributed computational support for building performance evaluation.” *Advances in Engineering Software* 33 (4): 199-206.

Lee, G., C. M. Eastman, and R. Sacks. 2007. “Eliciting information for product modeling using process modeling.” *Data & Knowledge Engineering* 62 (2): 292–307.

Lee, G., R. Sacks, and C. M. Eastman. 2006. “Specifying Parametric Building Object Behavior (BOB) for a Building Information Modeling System.” *Automation in Construction* 15 (6): 758–76.

Lee, G., R. Sacks, and C. M. Eastman. 2007. “Product data modeling using GTPPM -- A case study.” *Automation in Construction* 16(3): 392–407.

Mahdavi, A. 1999. “A comprehensive computational environment for performance based reasoning in building design and evaluation.” *Automation in Construction* 8 (4): 427-435.

Maile, T., M. Fischer, and V. Bazjanac. 2007. “Building Energy Performance Simulation Tools-a Life-Cycle and Interoperable Perspective.” Center for Integrated Facility Engineering (CIFE) Working Paper #WP107, Stanford University, CA, USA.

Microsoft Corporation. 2014. “NDde - Home.” Accessed August 5, 2014. http://ndde.codeplex.com.

National Institute of Standards and Technology (NIST). 1993. *FIPS Publication 183: Integration Definition of Function Modeling (IDEF0)*. National Institute of Standards and Technology, MD, USA.

Natural Resources Canada. 2014. "WATSUN 2009 | Natural Resources Canada." Accessed August 5, 2014. http://www.nrcan.gc.ca/energy/software-tools/7435.

Nielsen, T. R., R. H. Peuhkuri, P. Weitzman, and C. Gudum. 2002, "Modelling Building Physics in Simulink" Report SR-02-03, Department of Civil Engineering, Technical University, Denmark.

Nouidui, T. S., M. Wetter, and W. Zuo. 2012. Validation and Application of the Room Model of the Modelica Buildings Library. *Proceedings of the 9th International Modelica Conference*. Munich, Germany, September. Modelica Association. pp. 727-736.

O'Donnell, J. T., R. See, C. Rose, T. Maile, V. Bazjanac, and P. Haves. 2011. SimModel: A domain data model for whole building energy simulation. *Proceedings of Building Simulation Conference*, Sydney, Australia. pp. 382-389.

Object Management Group. 2014. "Unified Modeling Language (UML)." Accessed August 5, 2014. http://www.uml.org/#Links-Tools.

Pedersen, C. O., R. J. Liesen, R. K. Strand, and D. E. Fisher. 2001. *A Toolkit for Building Load Calculations*. Atlanta, GA: American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc.

Punjabi, S., and V. Miranda. 2005. Development of an Integrated Building Design Information Interface. *Proceedings of IBPSA '05 Buildings Simulation Conference*, Montreal. pp. 969-976.

Rose, C. M., and V. Bazjanac. 2013. "An algorithm to generate space boundaries for building energy simulation," *Engineering with Computers*: 1–10.

Sacks,R., C. M. Eastman, and G. Lee. 2004. "Parametric 3D modeling in building construction with examples from precast concrete," *Automation in Construction* 13 (3): 291–312.

Sreshthaputra, A, J. S. Haberl, and M. J. Andrews. 2004. "Improving Building Design and Operation of a Thai Buddhist Temple." *Energy and Buildings* 36 (6) (June): 481–494.

SunCast. 2014. "SunCast User Guide <Virtual Environment> 6.0." Accessed August 5, 2014. http://www.iesve.com/downloads/help/Solar/SunCast.pdf

Tiller, M. 2001. *Introduction to physical modeling with Modelica*. Springer, Boston, MA.

U.S. Department of Energy (U.S. DOE). 2014. "Building Technologies Office: Building Energy Software Tools Directory." Accessed August 5, 2014. http://apps1.eere.energy.gov/buildings/tools_directory.

Wetter, M. 2006. Multizone airflow model in Modelica. *Proceedings of the 5th International Modelica Conference*, Vienna, Austria. vol. 2, pp. 431–440.

Wetter, M. 2009. "Modelica-Based Modeling and Simulation to Support Research and Development in Building Energy and Control Systems." *Journal of Building Performance Simulation* 2 (2): 143–61.

Wetter, M., and T. S. Nouidui. 2011. Modeling of Heat Transfer in Rooms in the Modelica 'Buildings' Library. *Proceedings of the 12th Conference of International Building Performance Simulation Association*, Sydney, Australia. pp. 1096-1103.

Wetter, M., W. Zuo, T. S. Nouidui, and X. Pang. 2014. "Modelica Buildings library," *Journal of Building Performance Simulation* 7 (4): 253–270.

WFMC. 1999. *Workflow Management Coalition Terminology & Glossary (WFMC-TC-1011*. Hampshire, UK: Workflow Management Coalition Specification.

Winkelmann, F.C., B. E. Birsdall, W. F. Buhl, K. L. Ellington, A. E. Erdem, J. J. Hirsch, and S. Gates. 1993. "DOE-2 supplement, version 2.1E." Technical Report, Report No. LBL-34947, Lawrence Berkeley National Laboratory, Berkeley, CA.

Yan, W., M. J. Clayton, J. S. Haberl, W. Jeong, J. Kim, S. Kota, J. Bermudez Alcocer, and M. Dixit. 2013. Interfacing BIM with Building Thermal and Daylighting Modeling. *Proceedings of the 13th IBPSA Conference*, Chambery, France. pp. 3521-3528.