Technische Universität München
Ingenieurfakultät Bau Geo Umwelt
Lehrstuhl für Bauphysik

A Methodology for a Scalable Building Performance Simulation based on Modular Components

Matthias Josef Mitterhofer, M.Sc.

Vollständiger Abdruck der von der Ingenieurfakultät Bau Geo Umwelt der
Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Prüfungsvorsitzender: Prof. Dr.-Ing. Stefan Winter

Prüfende der Dissertation:

1. Prüfer: Prof. Dr.-Ing. Klaus Peter Sedlbauer
2. Prüfer: Prof. Dipl.-Ing. Thomas Auer
3. Prüfer: Prof. Dr. techn. Thomas Bednar

Die Dissertation wurde am 06.10.2017 bei der Technischen Universität München eingereicht
und durch die Ingenieurfakultät Bau Geo Umwelt am 05.12.2017 angenommen.

Preamble

The presented research in this thesis has been realized during my work as a research engineer at the Fraunhofer Institute for Building Physics (IBP) in Holzkirchen, Germany. I am very grateful that this time turned out to be a very exciting and instructive period for me, mostly due to the many people who were involved.

I want to express my greatest gratitude towards my colleagues. Advice, support and many hours of fruitful discussions were resources they provided with an open door at any time of the day. Therefore, a thank you to Sumee, Georg, Victor, Arnav, Andreas and many more.

I also want to thank Sebastian Stratbücker and Prof. Dr.-Ing. Gunnar Grün for their continuous support and trust in my abilities. The freedom they provided was the basis for creating new concepts and ideas within this thesis.

Furthermore, I express my gratitude towards Prof. Dr.-Ing. Klaus Peter Sedlbauer who guided me through this research process and provided orientation whenever needed. Moreover, I thank Prof. Dipl.-Ing. Thomas Auer and Prof. Dr. techn. Thomas Bednar for offering their expertise on building simulation.

I furthermore want to thank the scientific community at this stage. Numerous discussions at conferences, project meetings etc. have shaped the ideas leading to the development within this thesis. The feedback provided by external colleagues could not have been more valuable to me.

Last but not least, I thank my family and friends for their support and that stable and comfortable environment they provide every single day, which is the absolute foundation for so many things.

During my research I also produced important preliminary work which ultimately led to the findings of this thesis. Corresponding excerpts have been published in [76] and [78]. Parts of the presented results in this thesis have furthermore been published in [77] and [79] with the former article being awarded the best poster award of the Building Simulation conference 2017 in San Francisco, a biannual conference organized by the International Building Performance Simulation Association (IBPSA).

Contents

Abstract

Recent literature identifies a stagnation of Building Performance Simulation (BPS) as a supporting technology during the design process of buildings. Shortcomings regarding an integrated, scalable simulation which provides a holistic assessment of buildings while considering their dynamic nature have been remarked. The concept of modular building performance simulation has been accredited with considerable potential to overcome these issues and realize the vision of an integrated building simulation platform for several stakeholders.

This thesis presents a methodology incorporating the concept of modularity in order to generate a scalable BPS, which integrates different level of details (LOD), namely a single-zone and a multi-zone model for energy assessment and a zonal airflow representation for indoor climate assessment. The procedure builds upon the Functional Mock-up Interface (FMI) for tool-independent model exchange and co-simulation of Functional Mock-up Units (FMUs). Recent innovations in information technology, namely the field of Semantic Web Technologies (SWT), are deployed to introduce machine-readable, semantic information in FMUs in order to identify relations between them and automatically initialize the modular simulation. Therefore, an overarching data framework formulated in the Web Ontology Language (OWL) was implemented. This framework consists of a formally specified information model, which allows for specification of FMUs within the context of building simulation. Hence, the role of a simulation module within a BPS can be determined. The addition of project-specific information in the form of a digital representation of a building, corresponding to the paradigm of Building Information Modeling (BIM), completes the description of FMUs. The resulting ontology-based description enables an algorithm to automatically infer the simulation topology of an arbitrary number of contributing simulation modules, i.e. FMUs, by means of reasoning. Through the combination of a knowledge representation in the form of an ontology and a reasoner capable of inferring new information from this ontology, the method is by definition a knowledge-based approach, a discipline from the field of Artificial Intelligence (AI).

A case study featuring the Twin House, a building at the test site of the Fraunhofer Institute for Building Physics (IBP) in Holzkirchen, Germany, illustrates the feasibility of the approach and indicates its technological merit. The study deploys a digital building representation in the SimModel format and emphasizes the aspect of scalability in the simulation. It demonstrates a continuously refined simulation of the building through exchange and addition of simulation modules. A second case study proofs the independence from specific BIM formats and illustrates the compatibility with IFC (Industry Foundation Classes). A building from the IFC example repository serves as the test subject.

Kurzfassung

Seit einigen Jahren stagniert die Anwendung von energetischer Gebäudesimulation als unterstützende Technologie während der Planung von Gebäuden. Die Literatur identifiziert Defizite bezüglich einer integrierten, skalierbaren Simulation, die eine ganzheitliche Beurteilung von Gebäuden unter Berücksichtigung dessen dynamischer Natur bietet. Modulare Simulationskonzepte bieten eine Möglichkeit diese Lücke zu schließen und zur Realisierung der Vision einer integrierten Simulationsplattform für mehrere Stakeholder beizutragen.

Die vorliegende Arbeit stellt eine Methodik vor, die sich auf das Konzept der Modularität stützt um eine skalierbare Gebäudesimulation zu generieren, die verschiedene Detailstufen berücksichtigt. Diese umfassen eine einzonale und eine mehrzonale Abbildung des Gebäudes, um energetische Fragestellungen zu untersuchen, sowie ein zonales Strömungsmodell für raumklimatische Fragen. Die Basis für die Methodik ist das Functional Mock-up Interface (FMI), das einen tool-unabhängigen Modellaustausch und die Co-Simulation von Functional Mock-up Units (FMUs) ermöglicht. Entwicklungen aus der Informationstechnologie im Bereich von Semantic Web Technologien (SWT) werden eingesetzt um FMUs maschinen-lesbar in ihrer Semantik zu beschreiben. Dadurch können Beziehungen zwischen diesen erkannt und schließlich eine modulare Simulation automatisch initialisiert werden. Hierfür wird ein übergeordnetes Datenmodell, formuliert in der Web Ontology Language (OWL), eingesetzt. Dieses Datenmodel ist ein formal spezifiziertes Informationsmodell, das die Beschreibung von FMUs im thematischen Kontext einer Gebäudesimulation ermöglicht. Dadurch kann die Rolle eines Simulationsmoduls innerhalb einer Gebäudesimulation bestimmt werden. Die Ergänzung von projektspezifischen Informationen in Form eines digitalen Gebäudeinformationsmodels, entsprechend dem Konzept von Building Information Modeling (BIM), komplettiert die Beschreibung von FMUs. Die resultierende, ontologie-basierte Beschreibung ermöglicht es einem Algorithmus die Simulationstopologie einer beliebigen Anzahl an Simulationsmodulen, d.h. FMUs, durch Reasoning abzuleiten. Durch die Kombination von Ontologie und einem Reasoning-Prozess, der in der Lage ist neue Informationen aus dieser abzuleiten, ist die Methode per Definition ein wissensbasierter Ansatz, welches ein Teilgebiet von Künstlicher Intelligenz darstellt.

Eine Fallstudie basierend auf dem Twin House, einem Gebäude auf dem Testgelände des Fraunhofer Instituts für Bauphysik (IBP), zeigt die Durchführbarkeit der Methodik und demonstriert ihr technologisches Potenzial. Während dieses Beispiel ein Gebäudemodell im SimModel Format einsetzt und den Aspekt der Skalierbarkeit hervor hebt, zeigt eine zweite Fallstudie die Unabhängigkeit des Ansatzes von spezifischen BIM-Formaten und demonstriert die Anwendbarkeit mit einem Beispielmodel gemäß IFC (Industry Foundation Classes).

Nomenclature

Acronyms

Abox	Assertion box
AI	Artificial Intelligence
BCVTB	Building Controls Virtual Test Bed
BIM	Building Information Modeling
BPS	Building Performance Simulation
CAD	Computer-Aided-Design
CFD	Computational Fluid Dynamics
COP	Coefficient of Performance
DAE	Differential Algebraic Equation
DAI	Design Analysis Initiative
dll	Dynamic Link Library
FMI	Functional Mock-up Interface
FMU	Functional Mock-up Unit
gbXML	Green Building Extensible Markup Language
GIS	Geographical Information Systems
GUI	Graphical User Interface
HiL	Hardware-in-the-Loop
HVAC	Heating, Ventilation and Air Conditioning
IBP	Institute for Building Physics
IBPSA	International Building Performance Simulation Association
IDP	Integrated Design Process
IFC	Industry Foundation Classes

KPI	Key Performance Indicator
LOD	Level of Detail
MIL	Model-in-the-Loop
MPC	Model Predictive Control
OEM	Original Equipment Manufacturers
OWL	Web Ontology Language
PMV	Predicted Mean Vote
POMA	Pressurized zOnal Model with Air-diffuser
PV	Photovoltaics
SOA	Service-Oriented Architecture
SPARQL	SPARQL Protocol and RDF Query Language
SSP	System Structure and Parameterization
SWRL	Semantic Web Rule Language
SWT	Semantic Web Technology
Tbox	Terminological box
UML	Unified Modeling Language
URI	Unique Resource Identifiers
URL	Uniform Resource Locators
VEPZO	Velocity Propagating Zonal Model
W3C	World Wide Web Consortium
www	World Wide Web
XML	eXtensible Markup Language

Symbols

$\underline{I}$	Input Vector
$\underline{O}$	Output Vector
i	Input Variable
l	Number of Output Variables

m	Number of Input Variables
n	Counter
o	Output Variable
P	Power
Q	Heat Flow Rate
T	Temperature

Subscripts

C	Cold
con	Connections
cond	Condenser
el	Electric
eva	Evaporator
H	Hot
i	Index
in	Inlet
j	Index
tot	Total

Chapter 1

Motivation and Methodological Approach

The role of buildings in human society has changed throughout the past decades. Several years ago, buildings served as simple habitats providing protection against the outdoor climate. Nowadays, they present complex, technological systems consisting of several sub-systems that are characterized through various inter-dependencies. New disciplines, such as the design and optimization of entire quarters, where each building serves a designated purpose in order to exploit synergies, emerged.

This change was motivated by the growing awareness for potential energy savings in the building sector. Public initiatives, among others initiated by lawmakers, led the way for the development of numerous new technologies targeting resource and energy efficiency in buildings. Ever since, this wave of technical progress has been continued and fueled by advancements in other fields, such as information technology. The technical merit and prospect led to steadily increasing legal requirements for buildings while more and more key performance indicators (KPIs) are integrated. Besides regulations for energy usage, other criteria, often concerned with health aspects, moved into the focus of directives. Among others, these comprise acoustics, day-lighting, fire protection or air quality. In addition to that, further indicators regarding environmental aspects were established due to the continuously spawning demand of buildings in the market. Recently defined KPIs comprise, e.g. carbon footprint or greenhouse gas emission. However, not only lawmakers set benchmarks for the performance of buildings, also the expectations of occupants have increased and are now a major concern in the design process. This especially involves office buildings and aspects, such as thermal comfort or humidity. In addition to these emerging KPIs, familiar parameters like costs or time are still key aspects and often predetermined when designing a building. The growing list of criteria to be considered in the design of buildings ultimately leads to an increasingly complex task for engineers and architects.

The development of several new technologies led to improved performance of buildings regarding these KPIs. In this sense, especially renewable energy systems allow for accessing a range of possibilities in order to cope with the requirements regarding environmental aspects. However, many of these technologies benefit from synergy effects and rely on external influences, like weather conditions. The increasing level of inter-dependencies may lead to

unexpected situations and effects during operation. Furthermore, changes in the design phase may not only affect a single, but multiple planning domains and even multiple physical domains. An individual performance proof of a single sub-system is therefore hardly enough to ensure reliable operation of the entire building. Besides technical performance, many of these innovations require an extended assessment beyond the completion of a building. Operation and maintenance costs often play a key role when comparing design options. The operation phase of a building is therefore becoming increasingly important, leading to the necessity of considering further effects, such as induced through occupant behavior or control systems.
Regarding the described development of buildings, the concept of Building Performance Simulation (BPS) is a valuable remedy for stakeholders in the building sector. It allows one to gain insights into the non-intuitive behavior of buildings by capturing their complexity in computational models. Based on these models, pre-determined KPIs can be assessed and the uncertainty for planners can be decreased. This is especially important when new technologies or innovative solutions are involved in the building design. BPS can therefore also serve as a vehicle to provide market-access to new innovations and foster their application in practice.

Figure 1. Development of buildings from mere habitats to complex technical systems designed to meet continuously increasing requirements.

In a brief summary of BPS history Clarke [20] determines the start of BPS techniques to be the 1970s, when computer engineering started to spread from development stage into application. Disparate models, formulated as steady-state representations served to assess specific issues, especially regarding the behavior of single constructional elements. Second generation tools included dynamic effects for improved consideration of thermal inertia and varying boundary conditions. Also models for Heating, Ventilation and Air Conditioning (HVAC) systems were developed based on steady state formulations. In the 1980s, the integral aspect of modeling was emphasized. Previously isolated heat transfer processes were coupled in space and time, leading to a more complete representation of buildings. This process was extended in the 1990s with the integration of several domains, such as the building envelope and HVAC domains, into common models. In the following years, further developments involving among

others airflow modeling and dynamic models for numerous HVAC systems, especially renewable energy systems, were realized.
In order to coordinate and foster work in the field of BPS, the International Building Performance Simulation Association (IBPSA) was founded in the mid 1980s. IBPSA provides a platform for discussion and exchange to support and align the evolution of BPS. The organization consists of several stakeholders in the building sector, especially stakeholders involved with computational design processes. The diversity allows for multiple insights into the needs and prospects of BPS. Periodically published agreements of these views are aimed at addressing the current deficiencies and suggesting corresponding development paths for the future.
In 2015, Clarke and Hensen reviewed recent activities in the field of BPS on behalf of the organization [17]. They state that the development of BPS has in general led to an improvement of productivity in the building sector. Nevertheless, like other technologies, BPS is subject to the so-called *hype cycle*. The article is aimed to create awareness of current BPS issues and, as additionally stated in [50], intends to avoid a phase of disillusionment in this cycle. Potential for development of next generation BPS tools is found in the integration of technologies from other fields. Promising possibilities to enhance functionality and to extend application of BPS are especially seen, among others, in the World Wide Web (www) revolution or the advent of the digitization era.
Earlier, in an effort to provide a guideline for next generation BPS, Wetter [128] sketches a vision of future application scenarios. He describes an integrated design team working on individual tasks to form an aggregated result. Each member contributes with specific models to a simulation. The single modules of the simulation allow for fast evaluation of design options via module exchange. As an example, the HVAC engineer is able to drag one of numerous catalog models, developed by manufacturers during the product design phase, into the simulation to test the product's performance within the planned building system. The building physicist changes window properties influencing dynamic simulation models to quantify the effect on KPIs of the whole building, including the resulting effects on other domains like occupant behavior, HVAC performance etc.

The purpose of this thesis is to contribute to the development of BPS in alignment with the efforts from IBPSA in order to approach such visions. The thesis specifically investigates the possibility of a modular simulation methodology to mediate the technological development of BPS. The characteristics of the innovation are aimed to foster digital planning methods for overcoming the challenges that are faced by practitioners involved in the increasingly complex design phase. Figure 2 illustrates the methodological approach of this work.
The research is built upon an elaboration of the status quo in BPS, which is started with a brief overview of modeling strategies for thermal processes in buildings. Based on the literature, ideal characteristics of simulation functionality and the integration of simulation into the design process are derived. This section provides a reference for this work through drawing the ideal picture of an optimized simulation procedure. In the following, simulation approaches are discussed regarding their advantages and disadvantages to meet this optimum. This allows to formulate the shortcomings of these processes and to identify potential paths to solutions.

In an effort to determine technological vehicles for the development, recent innovations in the field of simulation and information technology are illuminated and assessed towards their capabilities to improve building simulation processes with regards to the determined characteristics. Subsequently, the required development to approach this goal is identified and aligned with current activities.

Based on the elaborated state of the art, a methodology for a modular BPS with scalable components is developed. The previously discussed technologies are combined to pave the way for a process featuring the desired characteristics of future BPS. In the following, the methodology is implemented on the Python platform and tested for feasibility and applicability in two case studies. The first demonstration example features an office building located at the test site of the Fraunhofer Institute for Building Physics (IBP) in Holzkirchen, Germany. A second case study is based on a building described corresponding to the specifications of IFC (Industry Foundation Classes). The procedure is subsequently evaluated and discussed regarding its theoretical and practical potential to meet the determined requirements for BPS. Based on these findings, a subsequent conclusion and proposals for future work are formulated.

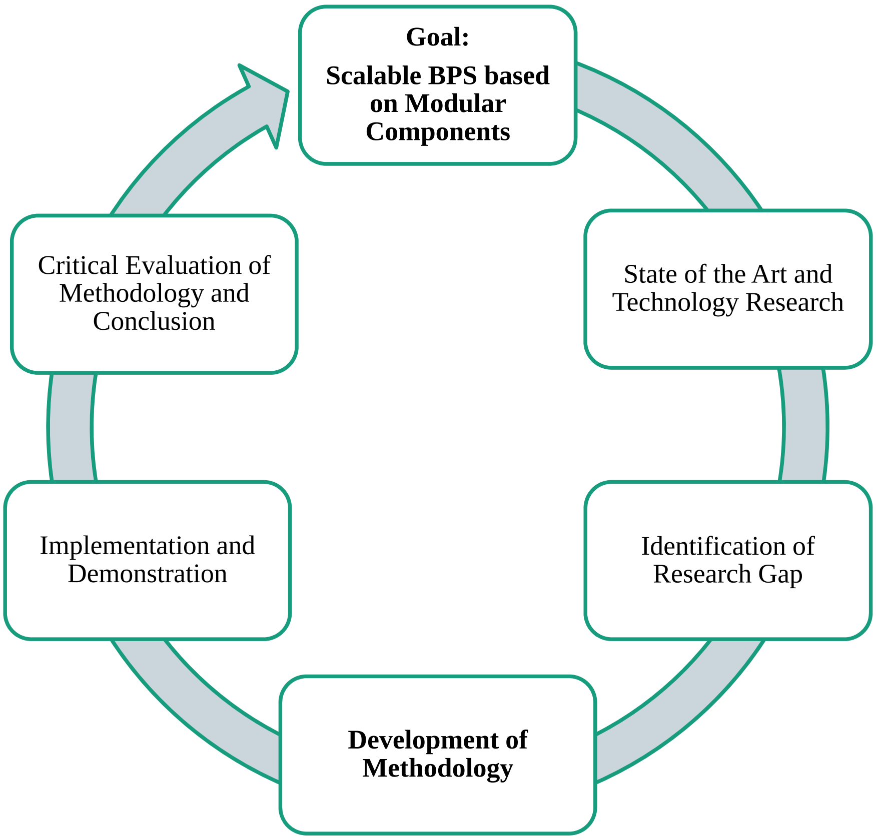


Figure 2. Step-wise approach of the thesis to reach the initially formulated goal of a modular BPS. After a technology review, drawbacks of current simulation procedures are identified and a theoretical concept to overcome these issues is derived. Two case studies demonstrate the feasibility of the methodology and set the ground for a critical review. Subsequently, a conclusion of the thesis is provided and possibilities for future work are illuminated.

Chapter 2

State of the Art

The elaboration of the state of the art provides an overview of developed approaches and recent developments in the building simulation sector. Goals for future innovations, embodying solutions to current issues in simulation procedures, are derived from the literature. The chapter gradually presents the available technologies and procedures to meet these goals. Finally, the determination of current shortcomings and the potential of recent innovations leads to the identification of the research question of this thesis.

2.1 Building Simulation Models

Physical models for BPS comprise several areas like acoustics, moisture or thermal energy. The following description is focused on the latter, emphasizing the thermal performance of a building system. In this context, the term *building system* describes the entirety of a building. Hensen [48] defines this entirety to consist of several dynamically interacting sub-systems, namely

- building envelope,
- environment,
- people,
- HVAC system and
- building equipment.

Generally, models to represent these sub-systems can be formulated as stationary or dynamic representations. Where a stationary computation is often applied in normative calculation procedures, such as the DIN 18599 [27] in Germany, in order to simplify application, dynamic simulation models offer detailed information about the behavior of single components at a high temporal resolution. Such models consider additional effects like thermal inertia through the storage capacity of a building system. This thesis is primarily concerned with the development of dynamic simulation techniques. Hence, the formulations used in the following are related to dynamic models.
In terms of accuracy and detail, different strategies can be applied within models. At this

stage, the representation of a building envelope serves to illuminate this aspect. In a BPS, the building envelope model serves as the response model reacting to influences from other subsystems, also called domains. Modeling strategies for this response behavior can be subdivided into four categories. Each category has its own designation. Where the first two are primarily intended to compute the annual use of energy, the latter feature appropriate characteristics regarding the assessment of thermal comfort or the treatment of specific situtations requiring a high-resolution approach. The following sections discuss the basic concept and highlight the differences of these strategies. For detailed descriptions of the governing physics it is referred to [20] and [116], which provide a sound overview of the resulting mathematical equations.

2.1.1 Single-Zone Model

The term single-zone model reveals that the energy transfer processes inside a building are aggregated to a single node as represented in Figure 3. The dynamic energy balance considers solar gains, conduction trough external walls, the thermal mass of the building, longwave radiation exchange, internal gains as well as convective heat flow by means of infiltration and ventilation. The single node representation also implies that e.g. internal loads or setpoints determined through occupancy must be combined for the entire building. Similar, heat or cooling power from heat exchange systems is aggregated corresponding to the homogeneous representation of the building. As mentioned in [21], such a model can serve as a first, qualitative estimation of annual energy usage in early project stages in order to provide fast insights on effects of design changes.

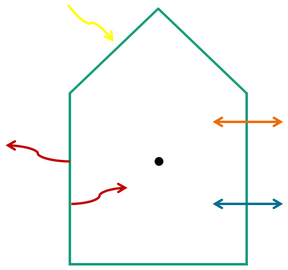


Figure 3. Structure of a single-zone model. (figure adapted from [91])

2.1.2 Multi-Zone Model

Compared to the single-zone model, the extended level of detail (LOD) in a multi-zone model lies in the differentiation of several thermal zones in a building. These zones can include one or more enclosed rooms. The energy balance is computed for every thermal zone resulting in a temperature information for each, as depicted in Figure 4. The consideration of different temperature levels within the building allows for computation of heat exchange between the zones due to the resulting temperature gradient. Energy flow from internal loads or HVAC systems can be distributed to individual zones. The increased LOD enables quantitative studies of used energy carried out over a year and can serve as an agent to several further applications. In [104], a multi-zone simulation model representing an office building is applied for model predictive control (MPC). Furthermore, optimization algorithms can be coupled. In [2], Asadi et al. performed a multi-objective optimization of design options for building retrofit. Similarly, a genetic algorithm was applied in [126] to improve green building design. A further example targeting energy management control can be found in [130].

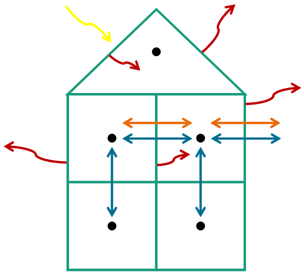


Figure 4. Structure of a multi-zone model. (figure adapted from [91])

2.1.3 Zonal Flow Models

Zonal flow models inherit the capability to predict flow patterns in enclosed spaces. Contrary to the above models they are able subdivide a single room into a coarse grid of air cells. Typically 10-100 air volumes are applied. This enables the computation of air movement inside a room. The mentioned energy balance is therefore extended with the Navier-Stokes equations allowing for calculation of the temperature distribution within the enclosed air space. Application ranges from one to a few connected rooms and shows especially benefits for the investigation of ventilation strategies or in cases when large temperature differences in a room can be expected. Dynamic effects are typically considered over the course of a few days leading to an assessment of the indoor thermal comfort at the corresponding boundary conditions. In [57] Inard et al. implemented such a model. Similarly to the pressurized zonal model with air-diffuser called POMA presented in [43] and the effort of Musy et al. [87], it is distinguished between zones of varying characteristics, e.g. low momentum or jet flow. In order to avoid the prerequisite to know the nature of the airflow in a zone, Norrefeldt [91] developed the velocity propagating zonal model (VEPZO), which computes a characteristic velocity for each zone. This property is propagated further downstream to ensure the distribution of driving airflow. In a subsequent effort, this model was extended with the RADZO model to compute thermal radiation on the same zonal grid. Validation and application of these models to comfort assessment in e.g. aircraft cabins or classrooms are provided in [91], [92] or [96].

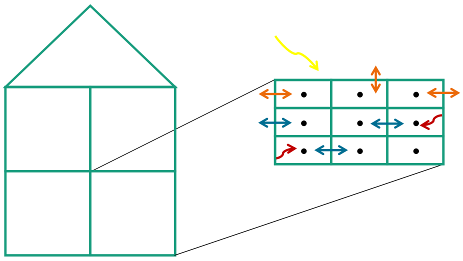


Figure 5. Structure of a zonal model. (figure adapted from [91])

2.1.4 Computational Fluid Dynamics

Similar to zonal models, Computational Fluid Dynamics (CFD) allows one to compute the movement of air. In contrast to the zonal approach, CFD methods subdivide air volumes into a fine mesh in the range of a few millimeter, as shown in Figure 6. This allows for detailed computation of specific flow situations, including turbulence, that may lead to e.g. uncomfortable velocities or ineffective ventilation. Due to long run-times, simulations are often restricted to stationary situations, a sub-volume of the enclosed space or time-frames of a few hours [133]. Nevertheless, application has been shown especially for the coupling to multi-zonal models in order to improve surface convection simulation [8, 133] and prediction of ventilation processes [114, 123]. Furthermore, in the case of particular situations demanding for great detail, such as personalized ventilation [38] or convectiive heat transfer from body segments [13], CFD provides the required high-resolution information.

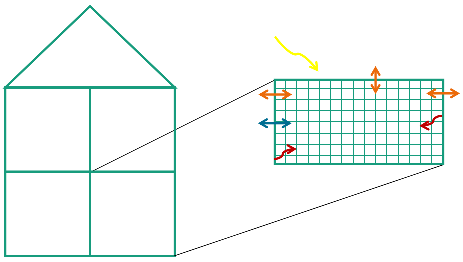


Figure 6. Structure of a CFD Model. (figure adapted from [91])

2.2 Building Simulation for the Design Process

Simulation tools and their integration into the design process have shown a rapid and dynamic evolution. KPIs in the planning landscape spawned, leading to an extension of planner's responsibilities and required expertise. Simultaneously, new developments in information technology and tool functionality emerged. These aspects show strong correlation and jointly form the basis for the following review, which aims at identifying the desired characteristics of BPS that have been formulated over the years. Subsequently, widely recognized BPS approaches are presented and evaluated corresponding to these attributes.

2.2.1 The Building Simulation Wish-List

BPS is meant to simulate the performance of buildings in many domains, as mentioned in section 2.1. Early simulation models assessed these domains separately through corresponding models. Tools were therefore tailored to a specific problem, e.g. load calculation of a building envelope or sizing of HVAC system components. Over the course of the last decades, numerous technical innovations, e.g. renewable energy systems, activated building components, phase change materials (PCM), etc. increased the inter-dependencies in buildings. The improvement of building envelopes in terms of thermal performance augmented the significance of detailed HVAC system analysis. Additional findings regarding, e.g. the influence of occupancy [73, 108] on energy usage in buildings, led to new domains that need to be considered in simulations. Therefore, an *integrated* simulation for holistic assessement of buildings has repeatedly been demanded [16, 18, 19]. In 2004, Hensen [48] writes that *"a frequently encountered problem by engineers, who would like to simulate the future behavior of building and system design alternatives, is that certain performance aspects or specific building and system components are only represented in one simulation environment while other performance aspects or components are only available in other software"*. For this purpose, an *open* simulation environment, that allows for component and model sharing among several stakeholders including manufacturers, producers and even building owners, is suggested. However, in 2015 Clarke and Hensen [17] conclude that little progress was made concerning the communication between tools to promote shared functionality.
They furthermore argue that the functionality of tools is often restricted to steady-state models. However, when assessing interactions between domains, representations of sub-models in a *dynamic* manner are a pre-condition. A truly integrated simulation can only rely on capturing the dynamics, which occur between sub-domains. Certain technologies, especially recent innovations in the HVAC domain, can only be assessed correctly through consideration of their own dynamic nature as well as the dynamic interaction with other components.

Besides functionality, integration into the design process is a further important aspect on the research agenda for BPS. An essential barrier for its use in practice is the complexity for simulation pre- and postprocessing [86]. Frequent application is required to efficiently setup a simulation targeting the designated purpose. Morbitzer [84] shows that the recognized benefit of BPS does not compensate for missing ease of use and often leads to outsourcing of simulation services to experts. In such a scenario the description of a design option is provided by one person and subsequently processed to be included in the simulation by a second person who then returns the result. The delay in designing and receiving performance feedback is unsatisfactory in the decision-making process [17]. Bleil de Souza and Knight [10] as well as Peterson [101] deduce that this subcontracting mechanism decreases the potential benefits of BPS. Instead of providing immediate design support by comparing various design options, simulations are applied for performance confirmation. Results from an inquiry [71] among austrian engineers and architects support this finding. Among those who consult results from a BPS, 76% are motivated only to ensure conformance to legal requirements. 3% state design improvement as a reason. Hence, BPS is primarily used in the design documentation phase (58%) instead of the early design stage (6%). Attia et al. [4] term the discrepancy in individuals generating the need for a simulation and individuals executing the simulation one of the largest problems in early design optimization. Accordingly, they demand an opportunity for *multidisciplinary users* to execute a simulation in order to get immediate performance feedback on their domain-specific design while considering the influence of, as well as on other domains.

The need for simulation-based collaboration of several planners originates from the idea of an integrated design process (IDP). The development of buildings towards highly technical, integrated systems motivates this approach. Negendahl [88] concludes from current studies that best performance outcomes of highly integrated buildings can be expected through multidisciplinary planning in mixed design teams. A survey from 2010 [3] reveals that practitioners prefer tools allowing for multidisciplinary planning. BPS is believed to be the mediating technology to promote and realize collaborative design. Besides direct integration of own expertise, planners develop an understanding for other domains and the simulation can serve as a catalyst for productive discussions directly involving multiple parties, including building owners.

A further aspect for successfull process integration of BPS is the capability to accomodate enriched input information over the course of a project [4]. Congruently, de Wilde and Voorden [26] demand an evolving simulation flexible in accomodating different LODs in different domains through *scalable* components. Being confirmed by Spitler [111] he pictures a scenario where non-simulation experts can apply simulation models in early building design stages that lead to detailed technical studies during the course of the project. Such an escorting simulation might even be operated by different users with different backgrounds working on different parts of the simulation. In 2012 Struck [113] re-emphasizes the need for dynamic resolution scaling of simulation models in order to accompany the design process and provide different support depending on its stage.

As emphasized in the past [5, 69], the overarching goals of research efforts in BPS are the integration into the design process and an holistic assessment of buildings. In order to achieve the former aspect, remarks from the literature identify the following attributes to be crucial for a BPS procedure:

- scalability
- multidisciplinarity
- openness

The holistic aspect must be ensured through simulations with the subsequent properties.

- integrated
- dynamic

With regards to these aspects, different BPS procedures have been developed over time. In [16], Citherlet identified four categories differing in their handling of input data and simulation functionality. The following description is adapted from these findings and extended with insights from Hensen [48].

2.2.2 Stand-Alone

In the stand-alone scenario, each planner is working in his own tool based on a tool-specific input data model, as represented in Figure 7. This approach originates from the early development stage of BPS when specialized tools were designed to serve a specific purpose. Performance of, e.g. HVAC systems, a building envelope, a wall construction etc. can be assessed individually. The advantage lies in the usage of tailored tools providing the most suitable functionality and processing capabilities for the problem under consideration. This guarantees high quality modeling based on detailed input data models. At the same time, the highly specialized functionality poses a limit to application for other purposes leading to the necessity of several tools for the assessment of several different problems. The missing possibility to consider inter-dependencies between these analyses is a further, decisive drawback of this approach. The desired integral view of a building can not be realized. Besides that, domain experts with the required expertise to apply the single tools are necessary. This leads to a high effort for data exchange between the project data level, which is generally governed by building designers, and the simulation input data level, which is managed by simulation experts. A development of simplified tools was aimed at moving the simulation process to the designer who operates directly on the project data level. However, Augenbroe [5] argues that the use of *"designer friendly"* tools poses a threat to simulation quality and furthermore becomes obsolete due to the advent of the www stimulating the delegation of remote services. The following approach, with the intention to improve integration of BPS into the design process, therefore aims to simplify the process from simulation request to simulation execution.

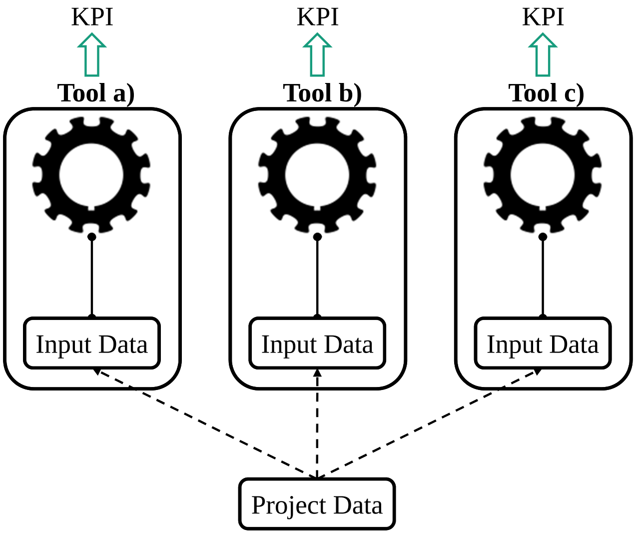


Figure 7. Stand-alone approach for application of simulation within the design process. Several specialized tools provide simulations based on individual input data models.

2.2.3 Data Model Interoperation

In this concept, similar to the stand-alone approach, tools are deployed individually to perform specific tasks. However, a central data model or standardized data exchange facilitates the generation of a simulation. Either option is stimulated by Building Information Modeling (BIM), a paradigm to integrate several stakeholders in a consistent data infrastructure through a building data repository and standardized data exchange. A sound overview of BIM, including its procedural implementation in the daily design process, is provided in [14] and [33].

Figure 8 sketches the concept of a central data container. The data model serves as the common information source to all tools. This requires an advanced data management system to ensure efficient storage, transaction and integrity after updates. Early approaches such as the Building Design Advisor [95], the Design Analysis Initiative (DAI) [6], the SEMPER project [72] or the Integrated Building Design System [20] incorporate several simulation tools to assess several KPIs. The automated instantiation of the simulation is based on proprietary data and software structure. This approach merges the input and project data layer. Individual characteristics, such as locally distributed access through online interfaces in SEMPER2, differentiate these concepts. The other option lies in the exchange of data between tools via standardized interfaces as shown in Figure 9. Formats, such as the IFC [15], as applied in [7], or the Green Building Extensible Markup Language (gbXML) [39], illustrated in its application in [45], are two possibilities to realize the data exchange. The process of data transportation, however, still poses a threat to consistency due to possibly imperfect import or export functionality or simply temporal discrepancies in the working documents of each tool.

Although the simulation model generation is facilitated in this approach, the discrepancy between the executing person and the designing person remains. A further disadvantage of either option is the still missing consideration of interactions between simulation domains. The following two developments found their motivation in this issue. Since the handling of input data can be managed similarly to the BIM-approach, it is disregarded in the following descriptions.

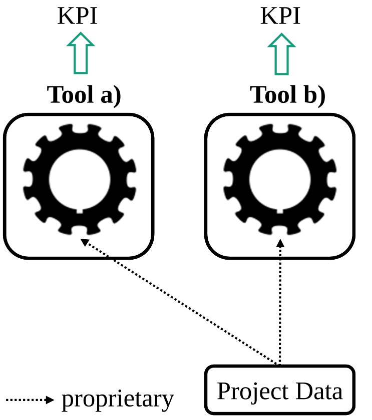


Figure 8. Data interoperation approach for application of simulation within the design process based on a common data repository.

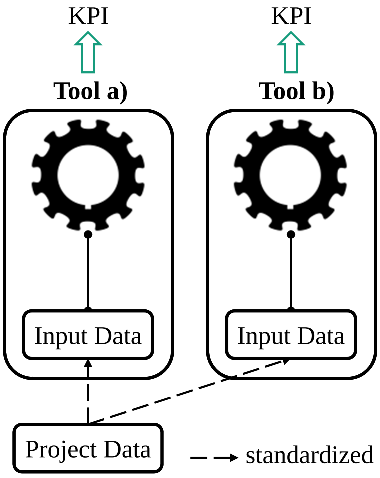


Figure 9. Data interoperation approach for application of simulation within the design process based on data exchange through standardized interface formats, e.g. IFC or gbXML.

2.2.4 Process Model Interoperation

An approach to realize a holistic building view is the integration of every domain on a physical process level, leading to a single simulation to assess multiple KPIs as shown in Figure 10. Efforts, such as in [1, 32] or [54], led to the extension of existing software, such as EnergyPlus [122], ESP-r [34] or TRNSYS [117], with functionality allowing for integrated analyses. This includes, amongst others, dynamic HVAC and building models as well as models for daylighting and airflow. A broad overview of further tools and the covered functionality is provided in [21] and [55].

Platforms, such as SEMPER2, integrated these approaches with the intention to combine the process integration aspect and a holistic building view [65]. However, as stated by Hensen in [48], these concepts only represent temporary solutions due to their proprietary structure. Besides that, the Modelica [82] platform has emerged featuring the required capabilities for multidisciplinary modeling on a code basis. The equation-based language allows for fast modeling and reuse of models for inheritance or independent use. Due to the encapsulation of each domain in a single model, the resulting (DAE) system can mathematically be simplified and optimized as a whole to be processed effectively by a tailored solver.

The concept extends the possibilities of simulation, however, integration into the daily design process, is an unresolved issue. The necessity of expertise in the extensive tool functionality and each planning domain reduce the possible user pool. Furthermore, quite established tools in certain areas hinder the adaptation and collaboration within or across companies [64].

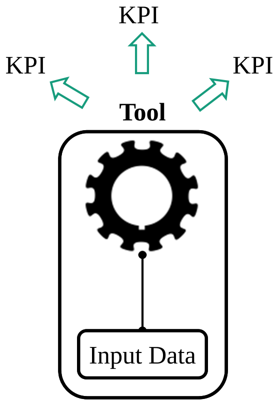


Figure 10. Integration of simulation models from multidisciplinary sources to form a whole building simulation - the integration of project specific data from a digital building representation provides project specific context required to combine the heterogeneous simulation modules.

2.2.5 Process Model Co-Operation

The missing tool interaction is targeted in the co-operation approach, as illustrated in Figure 11. Simulations on several domains are executed using several tools while exchanging information at interaction points during run-time. This allows for consideration of interdependencies between domains and enables an integral simulation instead of separated performance testing. Not every contributing model in this scenario must be able to compute a KPI. Instead, its purpose can also be reduced to complement the holistic view of a building. Citherlet [16] identifies the only problem with this approach to be the maintenance of link consistency between the different simulations. In [120], Trcka further mentions the required time and expertise for the realization of communication protocols between tools. It is concluded, however, that the dominating effects are advantages regarding flexibility of the simulation and extension of tool capabilities. An additional drawback is the increased simulation time due to the data exchange. Furthermore, solver specific optimization of the simulation DAE is not possible to the extend above, because of the decomposition into several single equation systems.

Nevertheless, Hensen [48] accredits the approach with considerable potential. It enables integration of different performance aspects as well as models inheriting different resolutions in terms of time and space. This is the prerequesite for a growing simulation through the course of a planning process. Clarke and Hensen [17] mention further advantages. New domain models can be coupled to existing models and extend their functionality. This can lead to the desired collaboration among stakeholders and even include software developers,

thereby allowing for multidisciplinary users. Furthermore, internal model characteristics can vary. This enables the integration of, e.g. empirical or statistical models next to physical representations. However, they remark the absence of an overarching framework for such an environment. Earlier, Mazzarella and Pasini [74] argued that a technology driven research is required on the path to such modular solutions. This especially involves efforts regarding the possibilities of current information technology.
The literature shows broad consent concerning the potential of the approach to leverage the development of BPS towards the desired characteristics described in section 2.2.1. A simulation consisting of single modules, exchangeable throughout the design process is envisioned to meet the demanded requirements. The following chapter is therefore devoted to the concept of modular simulation.

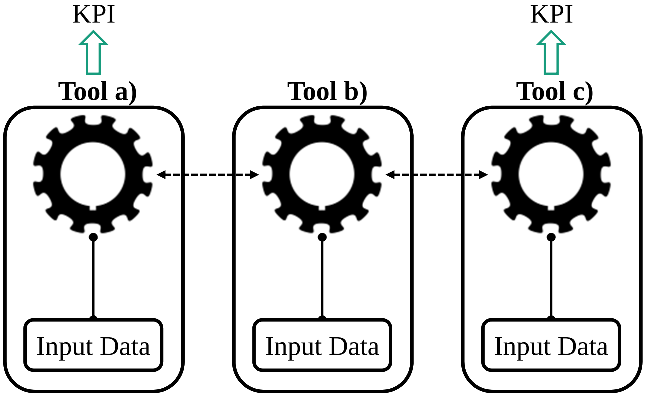


Figure 11. Co-operation approach for application of simulation within the design process realized through coupling of simulation tools.

2.3 Modular Simulation

This chapter illuminates the concept of modular simulation. In order to support a common understanding of the terminology, an established definition of the concept is provided. Subsequently, recent work in the field is highlighted and evaluated regarding the expectations towards the approach as indicated above. In addition, recent technological developments related to the subject are presented.

2.3.1 The Concept of Modularity

The idea of a modular simulation has been interpreted in various ways. In an effort to create a common understanding, Mazzarella et al. [74] described the goals of the approach as follows. A modular simulation

- improves the understanding of a system through its decomposition,

- allows for modification of the system composition according to needs,
- provides the possibility to extend a system with new modules
- and assures module re-use and access of modules in repositories.

The authors furthermore argue that different levels of modularity regarding the implementation of a tool must be considered. Subsequently, four levels of modularity were identified, as described in the following.

- Functional Layout Modularity: This level targets a user's interaction with the simulation. A modular simulation allows for selecting single modules that represent entities within a system. Theses entities can be added, removed or exchanged by a user. A challenge at this representational level is the connection of modules in addition to their selection. This can be automated or demanded from a user.
- Mathematical Models Modularity: The mathematical level of modular simulations relies on the encapsulation of a mathematical model in objects, a concept stemming from object-oriented programming. It aims at reducing mathematical interaction of subsystems to interaction points that receive, respectively send information. The extension with modules based on new mathematical models or the maintenance of existing modules is therefore facilitated. Changes of internal mathematical models need to maintain the integrity of the entire simulation, supposing that each module is still provided with the required information.
- Standardized Mathematical Models Modularity: Coupling of new modules to existing modules requires a common language for realization of information exchange. This not only implies standardized communication to ensure correct data transfer between modules, but also correctness on a numerical level. A standardized process for numerical solution of the sub-model as well as the overall model must be maintained throughout system changes.
- Code's Modularity: This level describes the ease to combine and re-use code parts. It aims at an object-oriented approach allowing for definition of classes with defined functionality and interfaces and, eventually, being based on inherited properties from parent classes.

Past efforts initially aimed at incorporating the concept of modular simulation on code level. Schuetze et al. [109] and Zimmermann [135] describe the implementation of the building simulator PSIGene. It consists of communicating objects that inherit the functionality of their parent class, e.g. equations for the thermal behavior of a wall. The idea follows the concept of object-oriented modeling, as described in [115]. Computation of the incorporated model is done in each instance individually. Variables, such as surface temperatures, are exchanged when necessary, following pre-defined communication patterns. A global equation system does not exist in this approach.

TRNSYS offers similar functionality for the simulation of HVAC systems. It enables the creation of FORTRAN procedures inside types. Instances of these types can be connected to form a greater system that results in a procedural code solved through iteration. A related concept is provided in Modelica. Physical processes can be modeled in classes and inherited to super-classes. Interfaces such as Dymola allow for connecting instances of these classes. In contrary to PSIGene, a global equation system is generated from the resulting overall model. This DAE is directly converted from the equation-based formulation of physical processes. It is adapted to the problem and has to be re-computed after changes. Special purpose tools, such as EnergyPlus or ESP-r, feature a fixed structure of the global DAE. The nature of objects allowed to contribute to the simulation is pre-defined. This degrades possibilities for extension with new object modules.

The mentioned examples congruently feature a range of characteristics for modular simulation, as described previously. However, meeting the goals for building simulation processes, as elaborated in section 2.2.1, requires the realization of modularity on a higher level. This is primarily due to the requirement of multidisciplinary users being able to cooperate in an open simulation environment. The examples show modular behavior on a local scale within a single tool. In order to extend simulations with modules from heterogeneous sources, a more global view of the modules is required. The coupling of models, i.e. modules, from different tools, as indicated in section 2.2.5, presents such a concept.

2.3.2 Co-Simulation

The process of coupling models from heterogeneous tools during run-time is referred to as co-simulation. Crucial to the concept is the realization of a communication process between tools. The communication does not only enable exchange of data but also synchronizes the simulations of the contributing tools. With regards to numerical stability and accuracy, different coupling strategies have been developed. Furthermore, different concepts for the decomposition of a model can be realized. A sound overview of these methods beyond the following descriptions can be found in [120].

Coupling Strategies

For realization of the coupling process two different strategies can be applied as illustrated in the following. Denomination follows the definitions by Struler et al. [25]:

- **Strong Coupling:** This method requires an iteration process between simulation time steps. Figure 12 indicates this through the subdivided enumeration. The first loop starts with the execution of a simulation (simulation A) until the end of a time step. The computed exchange value is provided to another simulation (simulation B) that subsequently updates the simulation progress towards the same time step. The required data point for simulation A is sent to the previous time step, initiating a re-computation of this step in simulation A. After repetition of the loop until convergence criteria are fulfilled, the co-simulation can proceed to the next time step. The iteration requires that simulations provide the capability to be re-set to a former state. It has to be noted that this method is also referred to as fully dynamic [132] or onion [49] coupling in the literature.

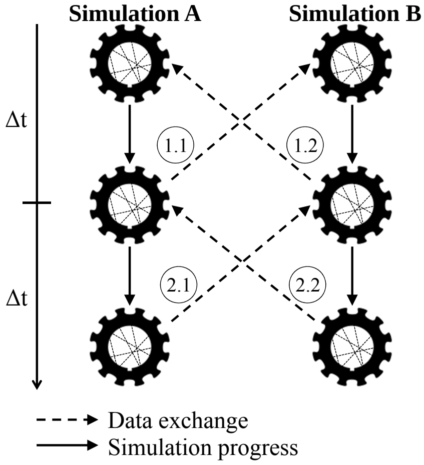


Figure 12. Strong coupling strategy for data exchange in co-simulation. (figure adapted from [120])

- **Loose Coupling:** This method does not rely on iterations and is therefore simpler in its implementation. Exchanged data is taken from previous time steps. In the scenario of sequential execution of the co-simulation participants, as shown in Figure 13, simulation B runs until a pre-defined time step is completed and provides data to simulation A. Based on this input, simulation A can be updated to the time step of simulation B and return the relevant result values. In a parallel execution of the simulations, as depicted in Figure 14, the data exchange in either direction is realized at the same time. Zhai [132] refers to this method as quasi-dynamic, Hensen [49] as ping-pong coupling.

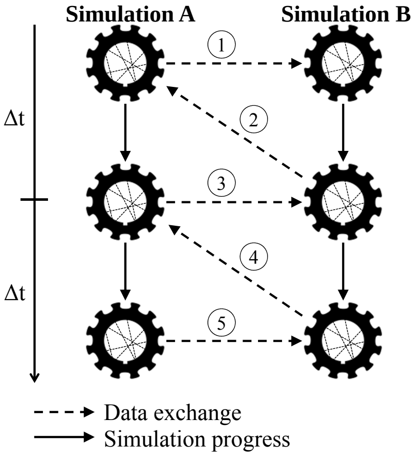


Figure 13. Loose coupling strategy for data exchange in co-simulation. In this case, the individual simulations are executed sequentially. (figure adapted from [120])

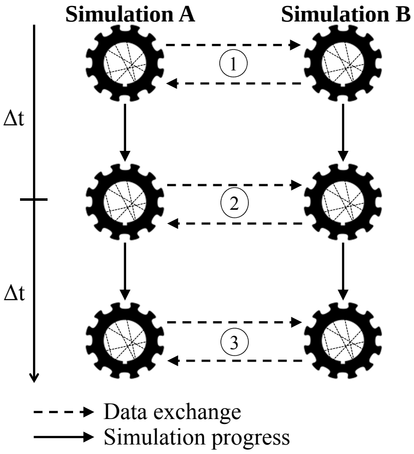


Figure 14. Loose coupling strategy for data exchange in co-simulation. In this case, the individual simulations are executed in parallel. (figure adapted from [120])

Decomposition Strategies

As described in [120], in order to decompose a system, two concepts can be distinguished: an intra-domain and an inter-domain approach. Trcka et al. [119] demonstrate an example study applying either concept. The former subdivides a system within one domain, e.g. the HVAC system. In this case, the interface was set to be the connection between heat supply and distribution system as shown in Figure 15. The latter approach realizes the decomposition at the functional boundary of domains, as exemplified in Figure 16 for the building envelope and the HVAC system. An assessment of the implications of either approach is provided in the following section.

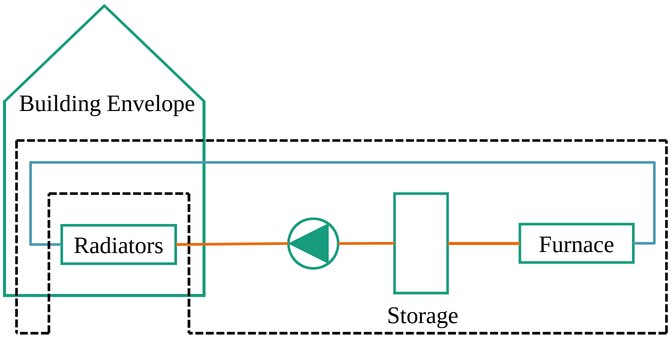


Figure 15. Intra-domain approach for decomposition of a building simulation with the interface in between heat distribution and supply system inside the HVAC domain. (figure adapted from [120])

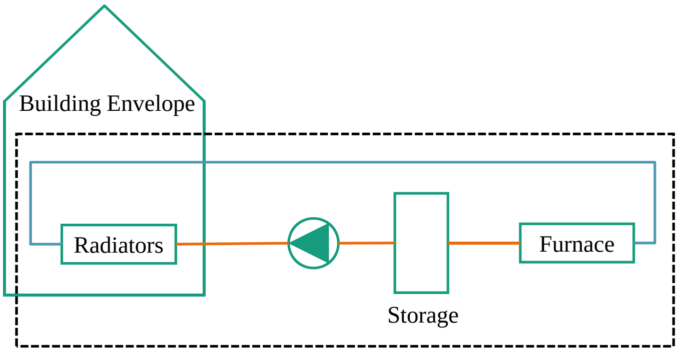


Figure 16. Inter-domain approach for decomposition of a building simulation between the domains building envelope and HVAC system. (figure adapted from [120])

Several contributions from the literature proof the feasibility and potential of the co-simulation approach. Janak [59] connected a daylight simulation in Radiance [36] to an energy simulation executed in ESP-r. In this setup, the latter provides information about sun position, direct and diffuse irradiance as well as zone and blind characteristics and invokes Radiance at each time step. The daylight simulation is executed with these boundary conditions and returns the indoor illuminance via a temporary result file that is read in ESP-r. Artificial lighting control in the energy simulation is set accordingly resulting in corresponding internal heat gains for the zone as well as electric power consumption during the next time step. The author showed that the combination of two specific tools leads to a significant improvement in predicting realistic lighting control and the associated implications for a thermal model.

In [31], Djunaedy deployed ESP-r and Fluent in a co-simulation. In a loose coupling ESP-r computed wall temperatures as well as heat sinks and sources inside a zone. The CFD model in Fluent was called after each time step and returned convective heat transfer coefficients from walls after they were computed in a steady state model. The data exchange was realized through an intermediary text file, requiring each tool to be adapted at source code level in order to read and write to the file. Exchanged variables must be mapped for each co-simulation setup manually in the realized setup.

Trcka et al. [119] realized the coupling of a building model in EnergyPlus and an HVAC model in TRNSYS. The two decomposition strategies corresponding to Figures 15 and 16 were applied. The exchanged variables in the intra-domain decomposition comply with conservation equations and are chosen to be mass flow, temperature and humidity ratio in either direction. The inter-domain decomposition exchanges heat flow rates in one direction and building temperatures in the other direction. Besides simulation partitioning, also coupling strategies were investigated. The communication process was realized through implementation of interfaces adapted to the architecture of either software on source code level. Provided with information about the connected variables, these components can be re-used for this tool combination. The various combinations of inter- and intra-domain decomposition as well as strong and loose coupling were compared among each other and additionally to a mono-simulation. It was shown that strong coupling with longer time steps achieved the same accuracy as loose coupling with shorter time steps. However, loose coupling performed faster due to the iteration loops in the strong coupling method and, provided with short time steps, generated results with the same accuracy as a mono-simulation. Regarding the ease of implementation, the authors recommended the loose coupling strategy with small time steps for future implementations. Inter-domain decomposition was accredited with shorter simulation time than intra-domain decomposition. However, accuracy of intra-domain decomposition is generally better, since the exchanged variables do not directly affect the inter-dependency between domains. Instead, this inter-dependency is modeled in a single environment and can be solved without the issue of a time offset. The authors conclude that the coupling leads to increased simulation possibilities that would not be possible in separated tools.

The mentioned examples show a common pattern when it comes to the orchestration of the communication process. In addition to providing values from its own simulation, one of the involved tools acts as the coordinator of the co-simulation, as depicted in Figure 17. Besides execution of the coupling strategy, the responsibilities of a master include initialization of the involved slaves as well as triggering steps in either simulation. Equipping a tool with such functionality requires additions on source code level.
Beausoleil [9] realized an altered approach using an intermediary "harmonizer" provided with master functionality. The harmonizer initializes the simulation in either tool and coordinates the data exchange and convergence check in the implemented strong coupling algorithm. The entity is realized as a compiled dll (dynamic link library). The method further relies on compiling the ESP-r and TRNSYS models to dlls, enabling the harmonizer to access subroutines inside these libraries. These subroutines are previously implemented functions of TRNSYS and ESP-r in the form of components or types. For ESP-r, the implementation is restricted to a few variables. In this case, sending and receiving modules for the exchange of hydronic and air-based coupling variables inside the plant network are created resulting in four additional components. This effort is due to the fixed structure of the global DAE in ESP-r, as mentioned previously. Integration of additional variables requires a selective manipulation, custom-made for each variable. Therefore, extension of the co-simulation functionality for other variables requires implementation of further components. In TRNSYS, implementation of one additional type enabling the connection to internal and external variables was enough. The connections allow for integration of the additional, external variables into the iterative solution procedure. The additions also inherit methods for stopping or re-setting the simulation to previous states as required by the harmonizer. Successful testing of the interface is reported in [68] with an intra-domain decomposition similar to the setup in Figure 15. The zone temperature computed in ESP-r serves as the input signal to the heating system controller in TRNSYS. In addition, fluid properties from the heat distribution system in TRNSYS are provided to ESP-r and returned after being processed in the ESP-r radiator model. Likewise, an air-based heating system was coupled. Beausoleil [9] evaluates the tool combination as an increase in functionality and accuracy when compared to the application of either tool on its own.

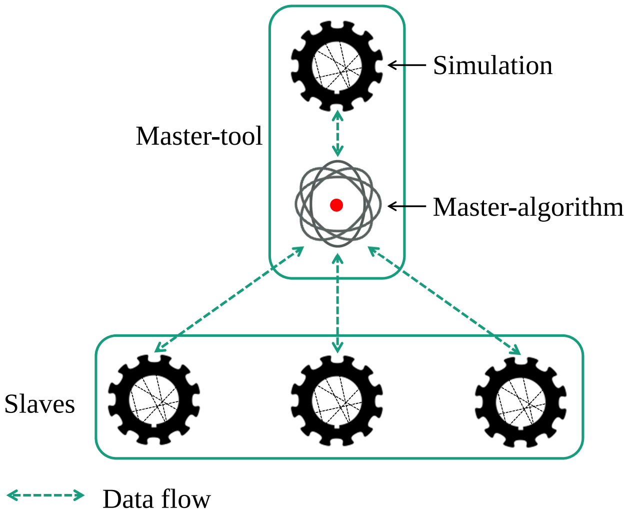


Figure 17. Data flow in co-simulation with a simulation tool acting as master. The slaves are directly addressed by a master algorithm inherited inside the tool in addition to the simulation model.

Besides the congruently emphasized improvements, the co-simulation of tools also shows disadvantages. Trcka [120] refers to the required time and expertise to establish a working connection between simulation models. Each of the illustrated examples was challenged with the investigation of an individual solution for the communication process. The most sustainable outcomes were produced in the latter efforts [9, 119] with the achievement of re-usability of such an interface for the targeted tool combination. As seen in the case of ESP-r, however, variation of exchanged variables often requires new implementations. Furthermore, each tool combination features particularities that need to be considered. This leads to the necessity of individual interfaces depending on the tool combination. An effort targeting this issue is the Building Controls Virtual Test Bed (BCVTB), as presented in the following.

Originally aimed at testing the performance of a controller in virtual buildings through a Hardware-in-the-Loop (HiL) concept [47], the BCVTB evolved to a co-simulation platform. It offers a collection of interfaces to a number of simulation tools and provides a C-library with functions to facilitate the implementation of interfaces to new tools. Similar to the harmonizer concept of Beausoleil [9], the platform acts as a master orchestrating the simulation slaves, as illustrated in Figure 18. Master functionality is adopted by the independent platform, reducing the required source code manipulations in the tools to data exchange functionality. A loose-coupling strategy is realized to manage the data exchange. In [129], Wetter and Haves show the successful coupling of EnergyPlus and Simulink enabled through the BCVTB. Outdoor and zone temperature information serve as input variables to a controller in Simulink, triggering window opening. A further example benefiting from the platform can be found in [107] with the coupling of a building model in EnergyPlus and multiple Matlab controllers for, e.g. ventilation, blinds, lighting etc. Simulation of a 9-zone building resulted in the exchange of 159 variables, a 28-zone building model in the exchange of 576 variables. The co-simulation was found to offer several advantages such as the use of a specialized tool for control algorithms and improved model maintenance and re-usability. Disadvantages concern the additional effort to define and implement the connections for data exchange as well as increased simulation time.

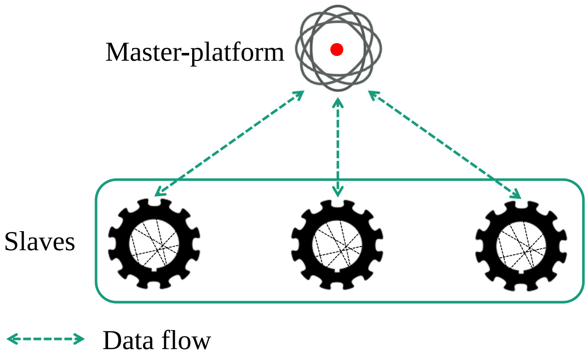


Figure 18. Data flow in co-simulation with an independent platform acting as master. The slaves are addressed by the master algorithm implemented in the platform. The functionality of each tool is reduced to the simulation.

Development of the BCVTB reduced the effort for co-simulation through offering implemented interfaces for a range of tools. However, users were still restricted to the supported tools and the need for local installations of the software. These aspects limited the usage of BCVTB to cases in which a researcher exploits the characteristics of one tool in order to complement the drawbacks of another tool as presented above. Collaboration among several parties was not promoted. This issue led to the development of a new technology, the Functional Mock-up Interface (FMI). The following section is devoted to this innovation and is aimed at illustrating its prospect for modular simulation techniques.

2.3.3 The Functional Mock-up Interface

The FMI is a tool-independent standard that allows dynamic simulation models to be exchanged or co-simulated. Its development was initiated in the MODELISAR [58] project with the objective to facilitate exchange of simulation models between Original Equipment Manufacturers (OEMs) and suppliers in the automobile industry. A tool supporting the FMI is capable of im- or exporting black-box simulation models called Functional Mock-up Units (FMUs). FMUs are zip-files that contain a descriptive *xml* (eXtensible Markup Language) file and C-functions, generally in a compiled *dll* file, as shown in Figure 19. The model description contains the definition of all model variables as well as their characteristics, e.g. input, output, parameter, unit, etc. Furthermore, FMU-specific capabilities are stated, such as relevant information for a co-simulation master algorithm to apply appropriate methods and settings. The C-functions contain the functionality of the simulation model. They can be attached in source form, however, in order to protect model know-how, compiled binaries for different platforms can be provided instead. This poses limits for the usage of FMUs since the simulation algorithm can not be changed any more. However, model parameters can be adjusted to individual needs.

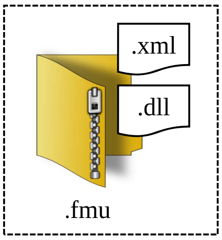


Figure 19. Content of a standard FMU zip-file: executable dll and descriptive xml file.

The first version of FMI was released in 2010 [11] and comprises two types of FMUs: Model exchange and co-simulation. The former is intended to serve as a possibility for distributing a simulation model in order to enable virtual testing through parameter changes. The latter enables standardized communication and functionality for co-simulation setups and avoids tool specific interface implementation as remarked in section 2.3.2. Two possibilities for co-simulation FMUs were projected, namely stand-alone and tool co-simulation. Stand-alone co-simulation FMUs can be executed independently from the source tool. The functionality of the simulation and its solution algorithm is entirely inherited in the FMU. Opposed to this, tool co-simulation FMUs require a local installation of the source tool. The FMU simply provides the standardized communication interface and runs the simulation within the tool in the background. Version 2.0 of the standard, as released in July 2014 in a Modelica Association project [80], dropped this option and merged the other two into one single FMU type where flags indicate the co-simulation or model exchange functionality. This version ensures complete independence of source tools when executing an FMU.
Additionally, the revision added the possibility for saving and restoring the FMU state during a simulation. This is especially relevant to co-simulation algorithms relying on iterations as necessary in a strong coupling strategy. Further improvements are described in [12].

In the following, several examples in the field of BPS are presented, which benefit from the possibilities of the standardized tool interface. With regards to the topic of modular simulation, the survey is focused on the co-simulation aspect of FMI.
Pazold et al. [100] applied the FMI to integrate several HVAC models from Modelica into the simulation software WUFI® Plus [35]. The HVAC models comprise, among others, a condensing gas boiler, a solar thermal collector, combined heat and power plants, storage tanks etc. The existing simulation software, focused on the thermal processes inside the building envelope, is extended with HVAC FMUs consisting of various combinations of these devices. This enables a user to chose among a range of pre-defined heat supply systems and individualize their parameters corresponding to his needs. An FMU adapter implemented in the software selects the corresponding FMU and initializes a loose coupling co-simulation.
In [93], Nouidui et al. document the implementation of the FMI into EnergyPlus. Two case

studies involving the coupling of a ventilation system and a shading controller as FMUs to a building model in EnergyPlus were conducted. Each example features EnergyPlus as the master of the co-simulation. The authors conclude that the standardized interface offers new possibilities for tool interaction and accredited the FMI with considerable potential on the path to an integrated building simulation process. Plessis et al. [103] implemented the FMI in the occupancy simulator SMACH. They exemplify the arising possibilities with a building FMU from Modelica coupled to a detailed occupancy behavior model. The combination allows for considering occupants' actions in the building model due to uncomfortable conditions. Aside from emphasizing the facilitated coupling process, the authors suggest research into auto-mapping of coupling variables based on input and output names and dimensions.

Similar to the early examples from section 2.3.2, these studies comply with the approach of inheriting the master functionality inside a simulation tool as depicted in Figure 17. Consequently, alike the BCVTB, also platform solutions for FMU co-simulation evolved.
In [23], Cremona et al. present FIDE, a development environment for FMI-based co-simulation. Once a co-simulation is set up via a GUI (Graphical User Interface), FIDE is able to compile the entire simulation to a stand-alone executable inheriting the master algorithm and each FMU's simulation functionality. The applied co-simulation algorithm features a strong coupling approach with adaptable time step. Discrete and continuous time-dynamics of each exchange variable are considered to extract the allowed maximum time step at each iteration. Both, the adaptable time step algorithm as well as the translation to C-code is expected to result in significant reduction of simulation time. References, however, are still missing in the literature. Galtier et al. introduced a further FMI co-simulation platform in [37]. Their solution, termed DACCOSIM, aims at large-scale simulations. Similar to the FIDE, it provides a variable time step in a strong coupling algorithm. However, single FMUs are handled by wrappers in different computation nodes. The wrappers identify the maximum time steps of the FMUs via e.g. Euler's method, which uses a comparison of computed and through previous derivatives predicted values. For large deviations, the FMU is set back and the time step is reduced. The wrappers report to a global algorithm, which sets the time step of each FMU to the required minimum. The advantage of this approach is the distribution of the simulation via multiple computation nodes to several cores on a personal computer or even a cluster. Performance comparisons for a co-simulation involving four FMUs showed slightly shorter computation time than a mono-simulation. Comparisons on a larger scale are not yet available. Further platform solutions can be found in [118] and [90]. Where the former allows for browser-interaction and high computational resources through cloud computing, the latter provides a number of advanced co-simulation algorithms including the Gauss-Seidel, Gauss-Jacobi and Newton approaches implemented with or without iteration, i.e. as strong or loose coupling strategy. It has to be noted that also the previously introduced BCVTB extended its support to the FMI. The implemented loose coupling algorithm can now also be applied to co-simulate FMUs.

Several efforts [16, 103, 120] have mentioned the necessity for managing the connections between co-simulation actors. This issue motivated a further Modelica Association project in 2014 called *System Structure and Parameterization (SSP)*. Its purpose is to complement the FMI with a second standard to describe system topologies and parameter sets in a homogeneous form. It arises from the need to transport information about a network of FMUs from one entity, e.g. a stakeholder or a simulation platform, to another. Intermediate results of the project have been presented in 2016 [64]. They indicate the structure of a *.ssp* (system structure package) zip file. This zip file contains XML schemes for

- the system structure definition (ssd),
- the system structure parameter values (ssv)
- and the system structure parameter mapping (ssm).

Despite the close bond with FMI, the SSP not only intends to describe co-simulation networks but also, e.g. Model-in-the-Loop (MiL) and HiL systems.

2.4 The Web Ontology Language

Recent developments in information technology led to a revolution of digital data processing and increased possibilities in data application. As suggested in [74], investigation of these technologies and their application in the field of BPS may trigger progress and result in an extension of the current capabilities and scope of BPS within the design process. Among these developments are Semantic Web Technologies (SWT) and, as an important component, the Web Ontology Language (OWL).

The World Wide Web Consortium (W3C) first standardized OWL in 2004 and in its revised form in 2009 [52]. The launch was motivated by the rapidly growing input to the www and the resulting difficulties to detect relevant content for a particular search. It was envisioned that an advancement to a *Semantic Web* can solve this issue. The idea of the Semantic Web lies in the assertion of machine-readable meaning to textual representation in the web. This is enabled through the association of content to ontologies.

Ontologies are description logics to characterize a certain area of interest in a structured way by representing knowledge about this subject in formalized network graphs. Components of ontologies can be classes, relations, restrictions, etc. Through connecting textual web content to these components, the web content receives context information. Hence, a machine is able to detect the relations to other components and is therefore able to recognize the meaning of the web content. This concept is depicted in Figure 20. It is distinguished between the Terminological box (Tbox) and the Assertion box (Abox). The Tbox contains the ontology representation, e.g. the taxonomy of classes including their relations and assertions as well as other components necessary to complement the description logic of a particular subject. The Abox contains individuals related to this ontology, i.e. distributed web content.

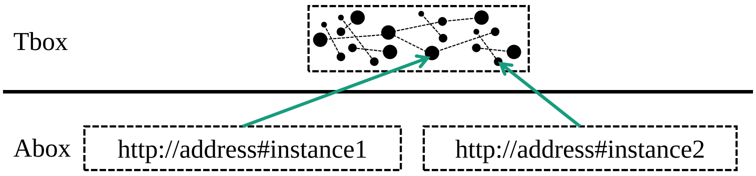


Figure 20. Partitioning of a knowledge base into Tbox and Abox. Abox content is related to the taxonomy in the Tbox.

The formulation of ontologies is realized with knowledge representation languages and as such, OWL plays an important role in the Semantic Web. In order to unambigiously define components of ontologies, OWL applies Unique Resource Identifiers (URI). A commonly known application of URIs are Uniform Resource Locators (URLs) for identifying the location of a website by means of a web address. A sound overview of the concept is provided in [51] and [125].

Besides providing context information to entities, a knowledge representation also enables the application of specialized algorithms, called reasoners. Examples for such implementations are the HermiT reasoner presented in [85] as well as the Pellet reasoner discussed in [110]. These algorithms allow for inferring new information from the existing knowledge representation. The concept relies on the *open world assumption* applied in Semantic Web languages. As an example, if a class *Cat* has three individuals - *lion*, *tiger* and *puma* - and *lion* is connected to *tiger* via a property called *isRelatedTo* and similarly *lion isRelatedTo puma*, then a reasoner would infer that also a *tiger isRelatedTo puma*. Opposed to this, a *closed world assumption* would by default assume a *False* statement when queried for the information if tigers are related to pumas.

A further possibility to create new information in a knowledge representation is the reasoning of rules. Rules can be formulated in the Semantic Web Rule Language (SWRL), as described in [53]. SWRL can be used to describe statements readable by a reasoner. These statements can be verified for a knowledge base and eventually lead to new inferred information about the knowledge base.

In addition to inferring new information from an ontology, the data graph can also be queried for present information. A utility to realize such a query is the SPARQL Protocol and RDF Query Language (SPARQL) [124]. It allows for formulating queries as triple patterns consisting of a subject, predicate and object. The language serves to traverse an information graph based on these triples in order to return the corresponding targets.

A number of successful examples for the application of OWL in the building sector can be found in the literature. Pauwels et al. [98] present an approach to automatically derive acoustic classification of a room. In [46], Han et al. describe a context-aware building energy management system which infers an energy waste context and provides suggestions for actions against it. An approach to automatically infer the root cause of a fault in building operation is described in [28]. Kofler et al. [63] implemented an ontology to enable energy analysis during the operation of buildings. Further examples include the translation of widely recognized formats in the field of BIM to OWL. As such, Pauwels and Terkaj [99] realized a conversion for the IFC data model. A similar effort is reported in [97] with the conversion of SimModel, orginally developed in [94] with the intention to capture relevant data for whole building energy simulation.

2.5 Appraisal of the State of the Art

The elaboration of developments and current activities in BPS was commenced with an introduction to model strategies for building simulation. It was noted that models can be formulated in steady state as well as in dynamic form. Furthermore, different model strategies can be applied, as illustrated for the building envelope domain. Their application depends on available information and targeted KPI. As such, they can be deployed at different stages in the design phase.

The following literature review identified the goals and challenges for application of building simulation. Increased complexity and numerous innovations, e.g. on the field of renewable energy systems, led to the necessity of a holistic building view. The integration of domains requires simulation models on a dynamic level. The resulting interactions can only be captured through high temporal resolution and consideration of non-steady phenomena. A main concern remains the integration of BPS into the design process. An adaptation in terms of an advancing simulation through the course of a project is demanded. Furthermore, stakeholders must directly be integrated into the simulation and receive immediate performance feedback for their anticipated design. In this sense, researchers remark the isolation of tools. Missing efforts in facilitating the communication among tools from different domains has led to a stagnation of simulation usage. An open simulation platform is demanded that allows for direct integration of models from several stakeholders into a simulation providing the opportunity to test their models under the influences of others.

The literature identifies four approaches that have emerged over time with the goal to meet these requirements. Starting with stand-alone tools and the accompanied shortcomings, more advanced approaches were developed targeting data and process models. Significant improvements concerning the communcation effort of input data could be realized through data model interoperation. Furthermore, extended tool functionality led to the possibility of holistic building assessment on a dynamic level. However, Clarke and Hensen [17] conclude in a recent review of BPS developments that the prospect on a solution meeting the full range of the above-mentioned requirements is low. Potential is congruently [16, 17, 48, 74, 120] seen in process model co-operation, i.e. a modular simulation concept as depicted in Figure 21. Such an approach provides the possibility to include models from several domains in a single simulation through connection at interaction points. Furthermore, it provides the required flexibility to adapt to the design progress. Single modules can be exchanged allowing for higher resolution when more information becomes available during the course of the design process. The modularity further offers several stakeholders to directly get involved with the simulation by providing own modules.

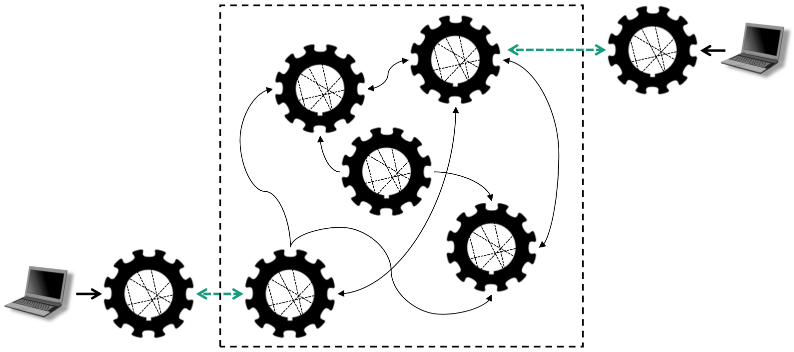


Figure 21. Modular simulation approach with several modules being connected in an environment open for several stakeholders, which are able to contribute and exchange simulation modules.

On a tool level, this modular concept has been used to facilitate user interaction. However, the collaboration aspect requires modularity above the tool level. Communication among tools was therefore moved into the focus of the research agenda. With the BCVTB, a platform emerged that collected knowledge about the individually explored and realized interfaces between tools. It allows for connecting a range of supported tools in co-simulation setups. Nevertheless, planners are still limited to the supported tools and the creation of additional interfaces remains a challenging task with individual particularities for each tool combination. This was solved with the launch of a standardized interface, the FMI. This tool-independent standard promotes the exchange and co-simulation of dynamic black-box models. Its development denotes significant improvements in the field of co-simulation.
With the initialization of the SSP project, the Modelica Association further aims at standardizing the transfer of information about network topologies as well as parameter sets of a co-simulation system. The format allows for exporting FMU co-simulation setups to platform solutions, as described in section 2.3.3, for efficient computation through intelligent co-simulation algorithms or distribution to multiple cores on local computers or clusters.
Despite the successful application of FMI in BPS for single use cases and the accredited advantages, a methodology capable of fully exploiting the modular nature of FMUs, is yet missing. Especially the issue of deriving connections between several FMUs has been an obstacle to many applications [16, 107, 120]. This thesis is aimed at developing a methodology, capable of enabling several users to contribute simulation modules from heterogeneous software to form a continuously refined simulation along the design process. The methodology must furthermore enable the integral assessment of buildings based on dynamic models and comply with the goals for modular simulation as mentioned in section 2.3.1, namely: system decomposition, model re-use, extendibility and system composition modification. The issue of setting up such a modular simulation by means of finding connections between modules is approached based on present information within the simulation modules. With the experiences in the www

discussed in section 2.4, which led to the semantic web revolution, a similar issue serves as example to ultimately develop an automated initialization of the simulation topology.
Figure 22 illustrates the alignment of this contribution with the current developments in the field of co-simulation. The FMI serves as the key technology to realize the generation of modules and their communication. The methodology developed in this thesis is meant to exploit these capabilities resulting in a modular building simulation meeting the mentioned requirements for BPS. In order to forward topological information about the simulation, a corresponding vehicle is provided with the SSP project. Functionality of simulation platforms regarding numerically optimized co-simulation algorithms and increased computational resources in the form of clusters, can then be accessed directly for execution of the modular BPS.

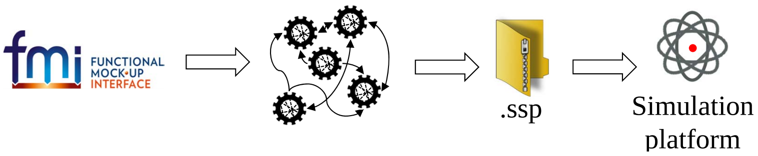


Figure 22. The path from FMI to co-simulation platforms enabled through the SSP project. The developed methodology in this thesis builds on the FMI standard and enables a scalable BPS based on modular components. This simulation can be forwarded to simulation platforms via the SSP format.

Chapter 3

Development of a Scalable Building Performance Simulation with Modular Components

This chapter describes the development of a modular BPS based on the FMI standard, starting with an introduction to the theoretical concept of the procedure. This concept is aimed at meeting the goals defined in chapter 2 and thereby clarifies, among others, the author's interpretation of scalability within the simulation. It is furthermore shown that the current content of FMUs needs to be extended in order to enable the realization of the procedure. The previously discussed innovation of OWL in the field of information technology serves as a remedy to overcome this issue. The remainder of the chapter treats the development process starting with the decomposition into single modules, i.e. FMUs. It is shown how OWL can contribute to the aggregation of these FMUs and to what extend information needs to be provided and stored to ensure re-usability and scalability of the modules. Concepts for realizing a modular single-zone, multi-zone and zonal airflow simulation are developed and presented.

3.1 Theoretical Concept

The process of model co-operation, as elaborated in section 2.2.5, is the foundation of this thesis. The approach allows for integrating models, in this case in the form of FMUs, from heterogeneous sources as depicted in Figure 23. Several stakeholders, such as planners from different disciplines, building owners or project managers as well as manufacturers must be able to contribute directly to the simulation with their own dynamic model translated to an FMU. The FMI standard guarantees the openness of the concept to each stakeholder through specification of an interface for co-simulation of the contributed models. These models may be based on individual tools and, corresponding to the approach for data model interoperation presented in section 2.2.3, associated with a common input data model enabled through BIM.

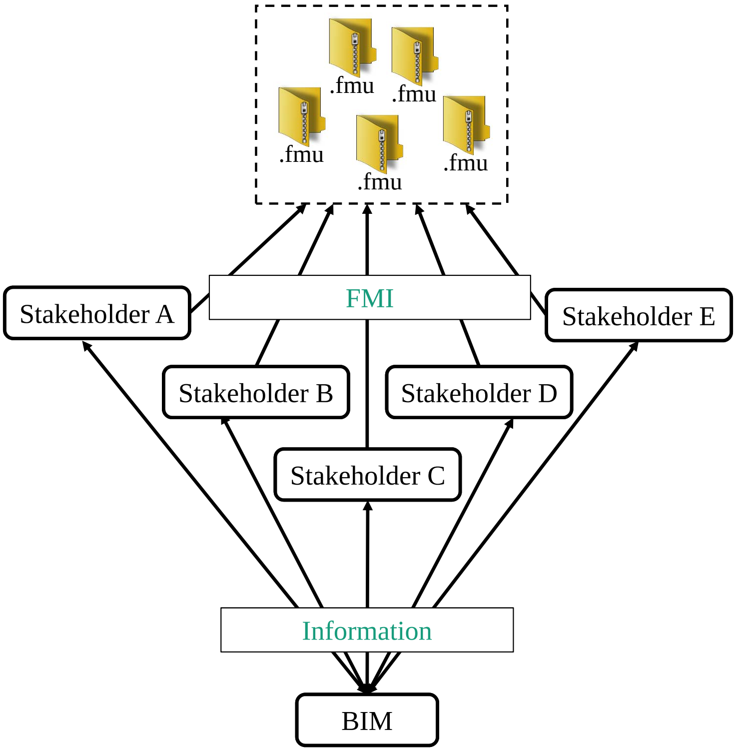


Figure 23. Process for FMI-based, modular simulation allowing for integration of several stakeholders.

The process must furthermore ensure the possibility to exchange single modules. Figure 24 exemplifies this aspect for different user groups that can be evaluated for a given building. Similarly, the opportunity to test various technical systems through the exchange of corresponding modules should be integrated. An exchange of modules must furthermore enable the application of different model LODs to ensure the continuation of the simulation along the design process. In this case, the in section 2.1 presented models, namely single-zone, multi-zone and zonal airflow models, are chosen to be integrated into the procedure. Due to the high degree of specialization and the limitation to particular problems, CFD models are not considered. Nevertheless, the selection allows to combine the assessment of two KPIs, i.e. energy and indoor thermal comfort.

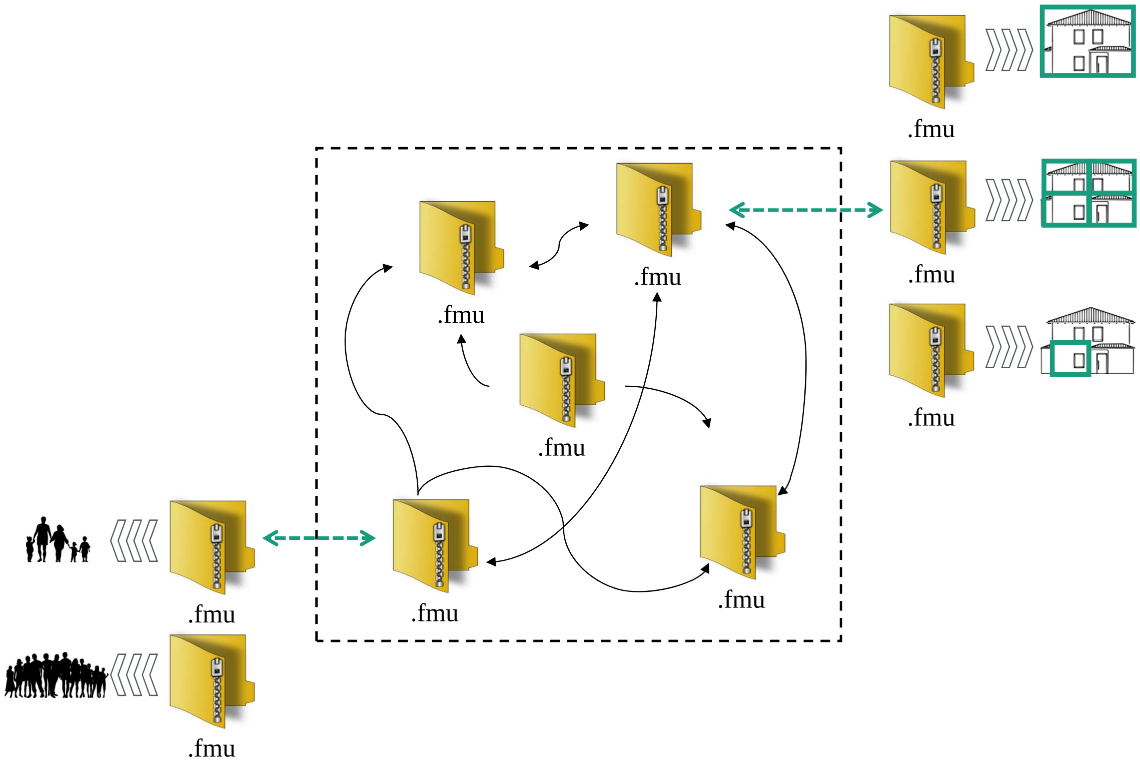


Figure 24. Concept for FMI-based, modular simulation allowing for exchange of single FMUs in order to vary system composition and LOD.

3.1.1 Simulation Decomposition

The development of a modular BPS starts with its decomposition into single modules. As mentioned in section 2.3.2, this can be achieved according to two alternative strategies: an inter-domain and an intra-domain decomposition. In this thesis, the in section 2.1 mentioned domains corresponding to Hensen [48] are applied and extended with an additional domain - building automation. In order to allow for clear distinction in the remainder of this thesis, each domain is briefly described in the following.

- Building envelope: This domain treats the thermal interaction of used materials and the enclosed air volume. Likewise, the behavior of opaque as well as transparent components of the building's skin and its internal constructions are included.
- Environment: External influences on a building arising from weather conditions are modeled in this domain. Apart from that, also passive information, such as the location of a building can be provided.
- HVAC: This domain involves the technical systems of a building that are required to achieve thermal comfort and a desired air quality. This comprises devices from energy generation to control, distribution, storage and energy exchange to the building.

- People: The people domain inherits the description of how people use and influence a building space. It comprises the user's attendance as well as their behavior and interaction with building components.
- Building equipment: This domain includes influences from devices and installations inside a building. Opposed to HVAC systems, these do not serve as conditioning elements but instead embody devices ensuring a building's purpose and functionality.
- Building automation: The building automation domain treats devices for the control of building components such as windows, shading etc.

The definitions allow to further address the choice between the two decomposition strategies. As depicted in Figure 25, the inter-domain decomposition leads to simulation modules with defined boundaries. The model scope of the single modules is fixed, thereby ensuring the compatibility of modules. Opposed to this, the intra-domain approach, shown in Figure 26, leads to modules with boundaries inside the domains. The functionality of these modules may comprise several domains. Furthermore, the scope of the covered content within these domains may vary resulting in floating boundaries. As mentioned in section 2.5, one of the goals of the developed methodology is module exchange and re-use. This can only be accomplished when the compatibility of contributing modules is ensured through fixed boundaries as is the case in an inter-domain decomposition.

On the other hand, previous work [119] found the intra-domain approach to yield more accurate results, as discussed in section 2.3.2. A definition of fixed boundaries is also possible for the intra-domain approach. However, the absence of a functional pattern complicates this task and poses a threat to a solution becoming a stable, widely recognized scheme. Instead, the strict module limits naturally implied in an inter-domain composition feature the appropriate characteristics for such a concept. This becomes even more apparent when considering the integration aspect of several stakeholders into the simulation. A clear definition of domains facilitates the allocation of responsibilities to single planners or other stakeholders. Additionally, the inter-domain approach has advantages regarding the source tools of the modules. As discussed in section 2.2, they are often intended to serve a specific purpose, i.e. providing a model of a specific domain. Hence, the functionality of the tools does not always allow for arbitrary model boundaries within the scope of another domain.

In conclusion of the discussed aspects, the modular simulation developed in this thesis is built upon an inter-domain decomposition of a BPS into the mentioned domains.

The chosen inter-domain decomposition allows for a categorization of FMUs, as illustrated in Figure 27. Within those categories, FMUs can be provided as ready-to-use simulation models corresponding to the scope of a domain. Such libraries can serve as repositories beyond single projects. An exception in this sense is the building envelope which is generally unique in the building sector. In contrary to the repetitive patterns or technical devices in other categories, a model of the building envelope generally requires an individual generation. The integration of functionality to directly create a building envelope simulation model within

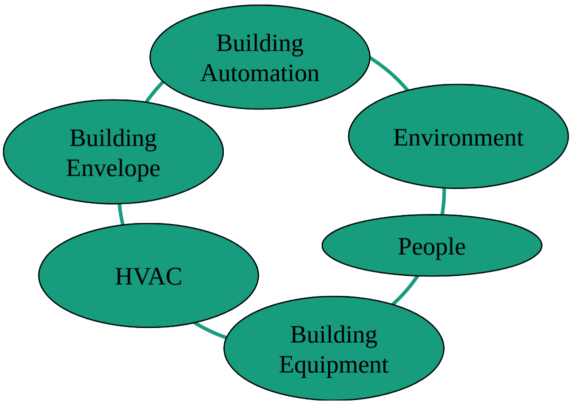


Figure 25. Decomposition of BPS using an inter-domain approach. Boundaries between modules are clearly defined.

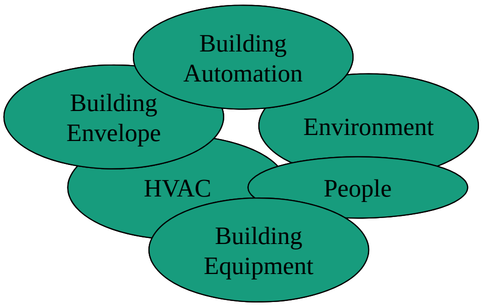


Figure 26. Decomposition of BPS using an intra-domain approach. The content of modules can not be assigned to a single domain. Instead, overlapping occurs at undefined and eventually flexible boundaries.

design tools, such as realized in [41] for the combination of SketchUp and EnergyPlus, present useful solutions for this task. Nevertheless, the effort to generate a simulation of the entire building system is greatly reduced through the provided catalog FMUs. It has to be noted that FMUs within those categories are not required to cover all aspects of the corresponding domain. However, the scope, i.e. the boundaries, of the domain must be respected. This facet is illustrated in the case studies presented in chapter 4. In the following, the issue of relating a set of chosen FMUs to each other and deriving the right connections among them is discussed.

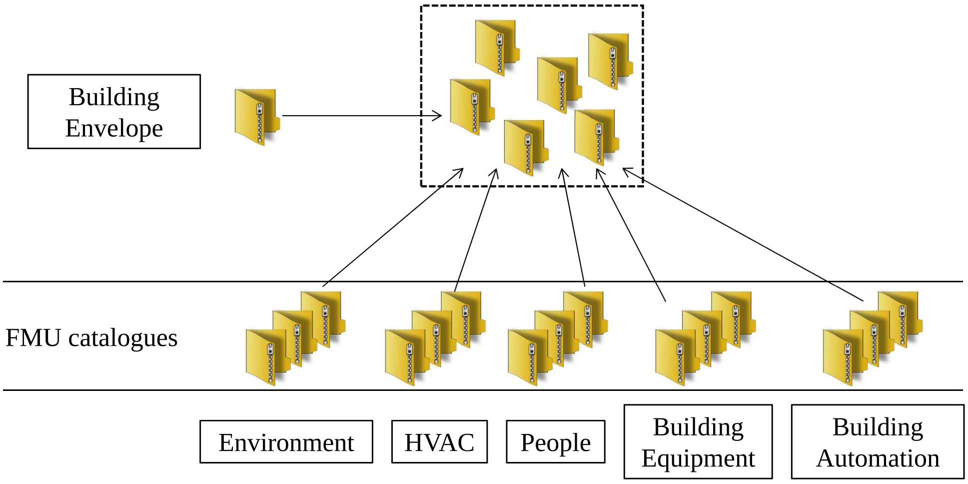


Figure 27. The resulting categorization of FMUs enabled through the realized inter-domain decomposition of a BPS. Apart from the building envelope module, libraries can provide a range of FMUs representing different types in each domain category.

Varying characteristics of modules in terms of the covered physics as well as their LOD pose a challenge to the aggregation process. As stated in section 2.3.3, with only a few modules included in the simulation, several authors [16, 103, 120] have found this task to be a decisive obstacle when treated manually. The need for an automated connection process becomes even more apparent when considering a simulation consisting of a higher number of FMUs or exchanged variables. The following analysis of the number of possible connections demonstrates the magnitude of this issue.

The output vector of a black-box simulation unit, such as an FMU, can mathematically be described as an enclosed function of the input vector

$$\underline{O} = f(\underline{I}) \tag{3.1}$$

with the output vector of length l

$$\underline{O} = \begin{pmatrix} o_1 \\ o_2 \\ \vdots \\ o_l \end{pmatrix} \tag{3.2}$$

and the input vector of length m.

$$\underline{I} = \begin{pmatrix} i_1 \\ i_2 \\ \vdots \\ i_m \end{pmatrix} \tag{3.3}$$

With the number of contributing FMUs in a modular simulation being n and assuming that inputs of a simulation unit do not receive values of their own outputs, the number of possible connections to the inputs of a single FMU amounts to:

$$n_{con,j} = m_j \cdot (\sum_{i=1}^{n} l_i - l_j) \tag{3.4}$$

An integration over each simulation unit leads to the total number of possible connections for the modular simulation.

$$n_{con,tot} = \sum_{j=1}^{n} n_{con,j} = \sum_{j=1}^{n} (m_j \cdot (\sum_{i=1}^{n} l_i - l_j)) \tag{3.5}$$

For a simulation consisting of five FMUs, with each having five input and output values, this results in 500 possible connections. Automating the derivation of the simulation topology minimizes the effort for detection of the right connections among these possibilities. Furthermore, as demanded in [89], in order to enable the application of optimization algorithms on the simulation, the integrity of the simulation must be preserved without manual interferrence after an exchange of modules. This ensures an extension from parametric optimization to an optimization at system composition level, i.e. from steady to non-steady objective functions. Such an example is shown in Mitterhofer et al. [76] for a case study of window and shading systems and their effect on indoor climate. Several library FMUs incorporating various window types and shading controls are coupled with an indoor climate model and tested in a batch setup for their effect on thermal comfort in summer. The main requirement for this optimization study was the automated derivation of connected variables after the exchange of a window FMU. In this case, the connected variables were of the same type throughout the entire optimization study leading to reduced effort for inferring the connections. However, such a setup is typically not the case in a modularized BPS where various types of connections can occur depending on the entity the modules are describing.

The matching of inputs and outputs can be compared with issues targeted in the field of Service-Oriented Architecture (SOA), a paradigm for software engineering. The following section therefore provides a short overview of the concept in order to facilitate the understanding of the developed methodology for automated derivation of simulation topologies.

3.1.2 Service-Oriented Architecture

The development of SOA was motivated with the dynamic changes in today's business processes. Traditionally, software concepts are based on a closed structure providing an implemented, limited amount of functionality to a fixed partner requiring this functionality. Especially innovations such as the www increased the need for more flexible software solutions capable of interacting with actors in a distributed environment. As illustrated in Figure 28, the concept of SOA promotes a segregation of this structure into entities that provide a service and entities that request a service. Entities providing a service inform a global registry unit about the kind of service they can offer to the system. If an entity in the system needs a service, it poses a request to the registry unit with a description of the corresponding service. The registry unit processes the description and determines the entity that provides the correct service. The reduction of entities to the mere functionality they can provide to the overall system, facilitates the maintenance of such a software and facilitates the extension with new entities. A more profound description of the SOA-concept as well as areas of application can be found in [134].

The modular simulation developed in this thesis adopts the basic principles of SOA. It treats simulation units as entities in a segregated model. FMUs can provide and request information, i.e. provide output and request input values. Similar to a registry unit, the methodology must inherit a concept to match this information. Newly integrated or changed FMUs can then instantly be provided with the required connections and contribute with other, eventually missing services. In order to create such a registry unit, information about the nature of the provided and requested variables must be appended in the FMUs. The *.xml* model description file is intended to inherit such meta-data. However, the arising possibilities for determining the meaning of exchange variables, i.e. services, are limited.

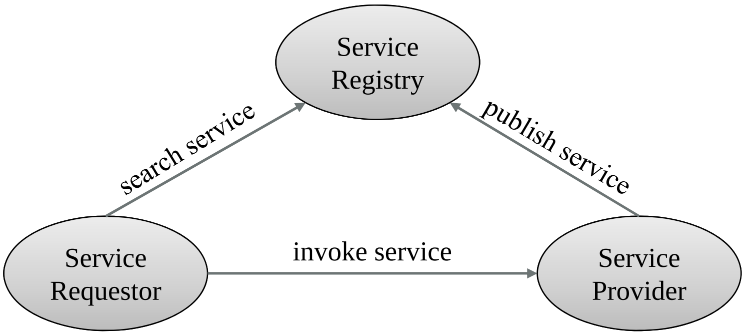


Figure 28. The principles of Service-Oriented Architecture [134].

3.1.3 The FMI Model Description

Information about FMU specific characteristics, such as version, origin, simulation relevant capabilities etc. are provided in the *.xml* model description file. Additionally, a description of parameters as well as input and output variables is contained. However, exported information differs widely due to differentiation between mandatory and optional entries.

As an example, two code snippets are presented in Listing 1 and 2 exported from Dymola 2015 [24] and EnergyPlus [22], respectively. Both variables describe heating rates of a building, one as input, one as output variable.

In Listing 1 meta-information for interpreting the variable *Q_Room* is included. It is described as the *RealOutput* type from the Modelica Standard Library and its unit is specified. Usable information reduces to the unit, which can be paired with other unit descriptions via string comparisons. However, solely the information about units is not enough to uniquely distinguish between variables as the comparison between a convective and a radiation heating flow rate demonstrates. Even a description with the type *Modelica.SIunits.Power* would require a variable from a different FMU to be described with a type termed in such a way that consensus can be derived regarding the type. In order to identify the meaning of the variable in a system, the description entry allows for human-readable textual descriptions. However, this information is not formalized and hence it is hardly possible to derive information through machine-based interpretation to support an automated process.

Listing 1. Excerpt of an FMU model description file created with Dymola 2015.

```
<UnitDefinitions>
 <Unit name="W">
  <BaseUnit kg="1"
   m="2"
    s="-3"/>
 </Unit>
</UnitDefinitions>
<TypeDefinitions>
```

```
 <SimpleType
  name="Modelica.Blocks.Interfaces.RealOutput">
   <Real/>
 </SimpleType>
</TypeDefinitions>
<ModelVariables>
 <ScalarVariable
  name="Q_Room"
  valueReference="335544325"
  description="heating power"
  causality="output">
   <Real declaredType="Modelica.Blocks.Interfaces.RealOutput"
     unit="W"/>
 </ScalarVariable>
</ModelVariables>
```

Fewer information is exported from EnergyPlus as presented in Listing 2. The annotation describes the variable as a continuous input variable. No information on quantity, medium or unit is provided. Instead, a naming convention is introduced.

Listing 2. Excerpt of an FMU model description file created with EnergyPlus 8.5.0.

```
<ModelVariables>
 <ScalarVariable
  name="Q_radRoom_W"
  valueReference="7"
  variability="continuous"
  causality="input"
  description="IDFline2811">
   <Real  start="0"/>
 </ScalarVariable>
</ModelVariables>
```

In conclusion, an automated collocation process can rely on the comparison of character strings in *.xml* files, as shown in Mitterhofer et al. [78], which follow a convention for the naming of quantities, variables, units etc. However, as mentioned by the authors, naming conventions often lead to long variable names, low flexibility and are generally not feasible in a heterogeneous environment involving multiple parties. The methodology developed in this thesis is therefore built upon the principles of linked data enabled through the usage of OWL, which allows for a structured connection of information.

3.2 An Overarching Knowledge Framework for Modular Simulation

The following section describes the developed integration of FMUs in a whole system simulation built upon the principles of SWT explained in section 2.4. An overarching ontology called *FMUont* provides a description logic for an FMU-based modular BPS. As illustrated in

Figure 29, the elements in this Tbox serve as anchors for information contained in the individual FMUs. Through relating the information inside the single simulation units to a common data model, they can be associated with the same context. Hence, when selecting a range of FMUs for simulation, a description of the provided and requested services of these FMUs, i.e. exchange variables, is gathered in the knowledge base. Subsequently, this description can be used by a reasoner to infer the required connections among the input and output variables of the FMUs.

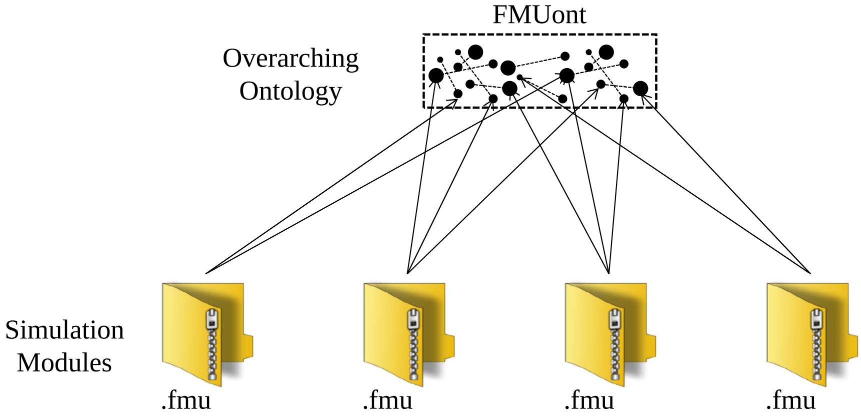


Figure 29. The framework for the knowledge-based integration of simulation modules. Information contained in FMUs is related to an overarching ontology. Information contained in several FMUs can therefore be connected with each other.

3.2.1 FMUont - An Ontology for the Description of FMUs

Figure 30 shows the overarching ontology *FMUont*, designed to accommodate information about FMUs and their variables in order to allow for inferring the simulation topology. The structure is adopted from the FMI model description definition discussed in section 3.1.3. The concept consists of a class *Simulation* that can be composed of several FMUs, which inherit model functionality of a *SimulatedSystem* corresponding to the defined domains and LODs. The class *FMU* can have *Input* and *Output* variables as well as *Parameters*. These classes are associated via the object properties *hasInputVariable*, *hasOutputVariable* and *hasParameter* respectively. The classes *InputVariable*, *OutputVariable* and *Parameter* are all subclasses of *AnnotatedElement*. An instance of this class can be described through several attributes. These are intended to describe the nature of a variable and determine its provided or requested information to a simulation. Accordingly, the object property *hasMedium* relates an *AnnotatedElement* to instances of a class *Medium*. Similarly, the classes *Quantity* and *Unit* can be used via their corresponding property associations. At this stage, established ontologies from other fields can be integrated.

So far, the descriptive attributes are base properties allowing for a general description of FMUs independent from their area of application. In order to include BPS-specific information, a *SemanticType*, as introduced in [29], is included. In contrary to the properties above,

this class does not feature pre-determined elements as part of an established framework, like physical quantities or units. Instead, it inherits meaning within a specified context, in this case BPS. This allows to assign a variable to a certain aspect inside a BPS and enables its classification within this context.
In order to relate a variable to its descriptive properties, instances serving as anchors in the Tbox must be defined. The definition of these instances is treated in the following.

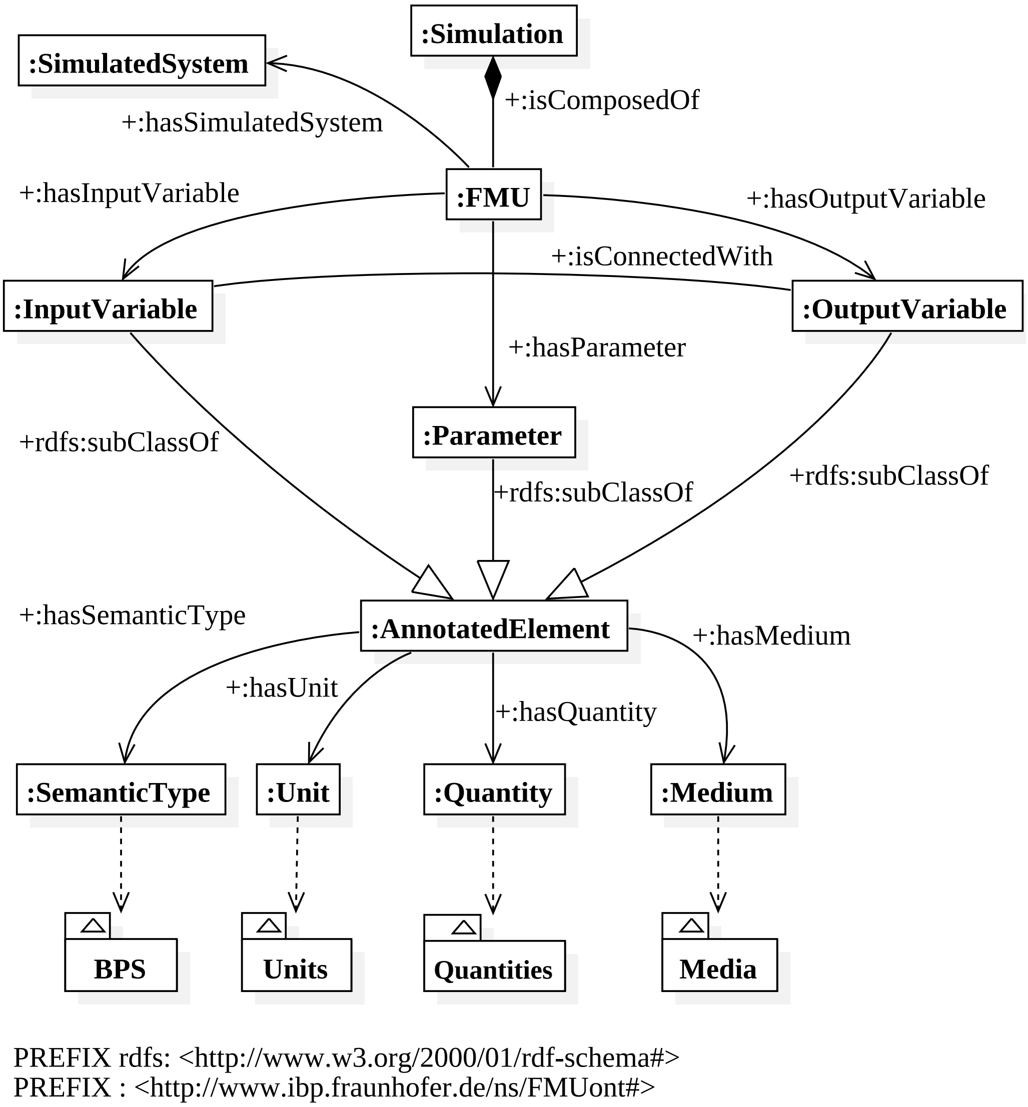


Figure 30. The taxonomy of FMUont. The ontology is used as an overarching data model in order to integrate distributed information in single FMUs into a common context. The prefixes indicate the definition of unique identifiers in different ontologies.

Quantities and Units in FMUont

For the definition of quantities and units, the concepts from the *Ontology of units of Measure (OM)* [105] are applied. This ontology provides descriptions of physical base quantities and decomposes them into sub-quantities. Each of these quantities can have several units, which are similarly defined through the combination of base units. The ontology provides a sound description of interrelations between dimensions and measures and offers a common database for numerous applications.

In BPS, certain quantities may occur outside the pattern of generally acknowledged physical quantities. Instead, these quantities were established based on studies suggesting the practicability of such a quantity for the purpose of interpretation and evaluation in the field of BPS. An example for such a quantity is the Predicted Mean Vote (PMV). The PMV is used to quantify the expected comfort at certain indoor climate conditions. It is based on statistical analysis of empirical studies and presents a common measure for evaluation of thermal comfort in buildings. Due to its empirical origin, no classification within the structure of the mentioned OM ontology is possible. Similarly, no direct classification for the concept of an operative temperature can be found. This measure describes a weighted average of air and surrounding wall surface temperatures and is frequently used for comfort studies. In contrary to the PMV, this quantity can be associated with a physical unit, however, due to its empirically motivated definition it is not included in a listing of physical quantities as the one above.

For the purpose of the procedure developed in this thesis, the definition of basic quantities, such as *Temperature* or *Power*, from [105] is extended with additional quantities, such as the mentioned *PMV* and *Operative Temperature*. A complementing list of quantities and units determined to be used in the remainder of this thesis can be found in Appendix A.1.

Media in FMUont

For the definition of media, several common instances from the building domain are chosen to serve as anchors for the variable description. Among others, they comprise, *Water*, *Brine*, *Air* and *Carbon Dioxide*. In addition to fluids also media describing solid texture, such as *Ground*, are defined. A complementing list of instances is provided in appendix A.2.

Semantic Types in FMUont

The semantic type instances are intended to complement the description of a service a variable can provide or request within the simulation. In order to allow for a seamless exchange of simulation modules, this description must universally be applicable to every module. Since the single modules feature the domain boundaries defined in section 3.1.1, the semantic types must provide a general description of the interaction points between the domains. An association to concrete systems can not be included in this description. This ensures the general applicability of semantic types to all modules, regardless of the module's represented entity within a domain.

Figure 31 shows the derived semantic type instances. The environment domain is able to provide information about a building's *Location* and corresponding *Weather Data*.

As elaborated in section 2.1, the building envelope model serves as a passive response model to the simulation, embodying the thermal reaction of materials to various influences. As such, it is able to provide information about the current conditions in a building, i.e. when considering a spatially resolved building also in different spaces of a building. Hence, the semantic type instance *Space Condition* is determined for information describing the status of building spaces.

HVAC modules simulate the behavior of active components inside a building responsible to condition the air according to given setpoints. MacDowall [75] determines several processes that are responsible to achieve such an impact. The reduction of semantic types to these processes allows for system-independent description of influences from HVAC systems. Accordingly, the semantic types *Heating*, *Cooling*, *Ventilation*, *Humidification* and *Dehumidification* are defined.

In [70], Mahdavi sub-divides the people domain into active and passive influences on the performance of a building. The latter present the mere *Occupancy* of people resulting in internal thermal loads as well as others, such as humidity, carbon dioxide etc. He further describes the active influences as actions directly targeting elements of a building. These elements comprise *Shades and Blinds*, *Windows*, *Setpoints* of various nature, such as air temperature or carbon dioxide level, as well as *Lighting*. The semantic type instances for the active and passive effects from people within a simulation are adopted from these elements.

Devices within the domain of building automation are able to provide the same set of active influences as *People*. The resulting conformity underlines the service-oriented paradigm of the description. With regards to active effects, modules from the people and building automation domain can not be distinguished as they ultimately result in the same service provided to the simulation.

Ultimately, in addition to the passive influences from people, *Equipment* can yield further internal loads within a building. An overall summary of the defined semantic types is provided in appendix A.3.

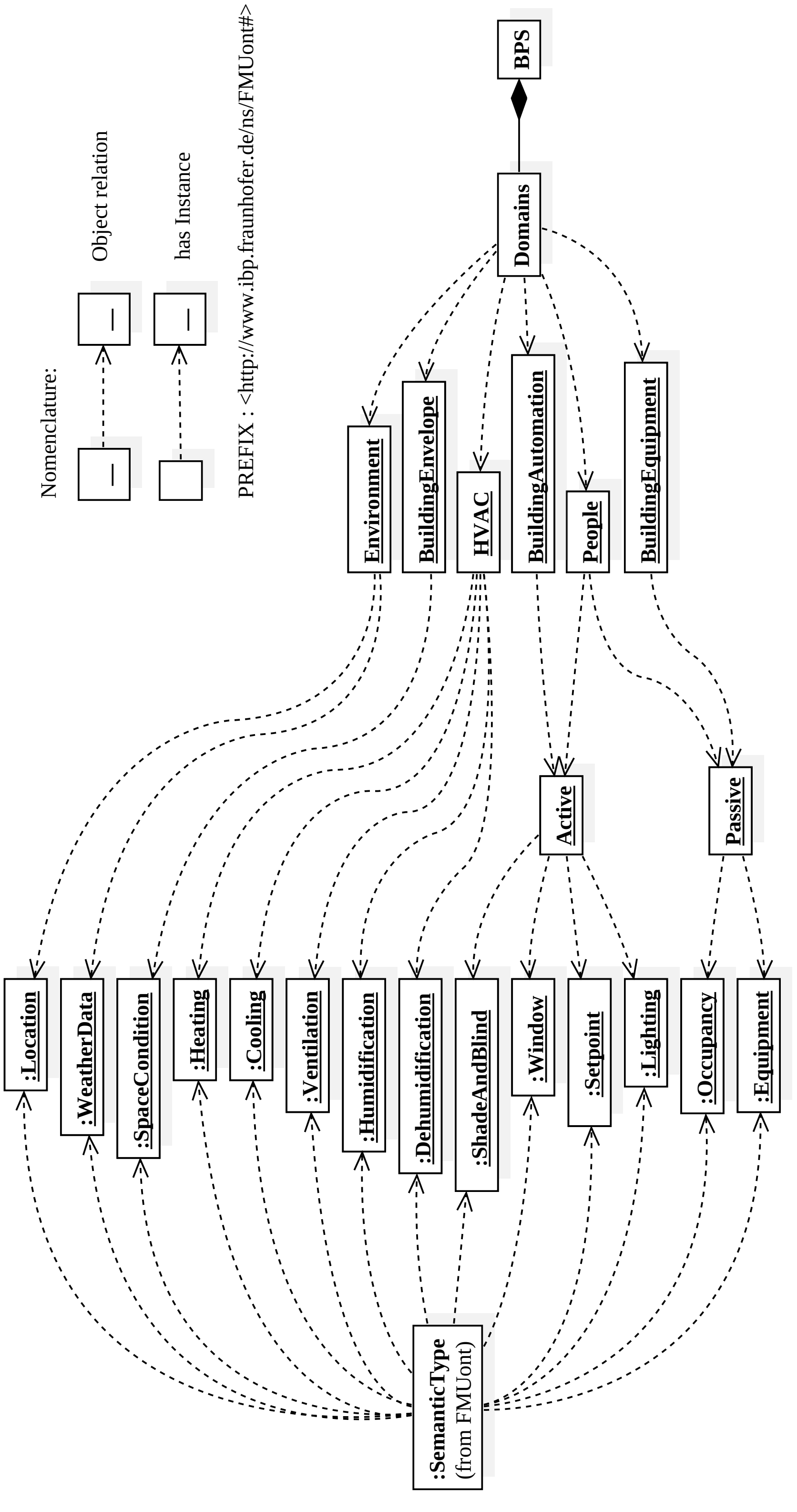


Figure 31. The derivation of semantic types in FMUont. For each domain a range of semantic types are defined.

3.2.2 FMU Extension

The presented instances for the description of variables as well as the overarching structure of *FMUont* can now serve as a reference for the description of individual simulation units. In order to accommodate this description in FMUs, the original composition of the FMU zip-file needs to be extended. In addition to the standard content, i.e. the executable *dll* file and the *xml* model description file, an *owl* file is appended. This extension is illustrated in Figure 32. The responsibility of the former files primarily lie in providing the FMU's simulation functionality. In addition to information relevant to the execution of the simulation, the model description file also provides a description of variables, as discussed in section 3.1.3. The *owl* file extends this description with attributes being associated to the overarching ontology *FMUont*.

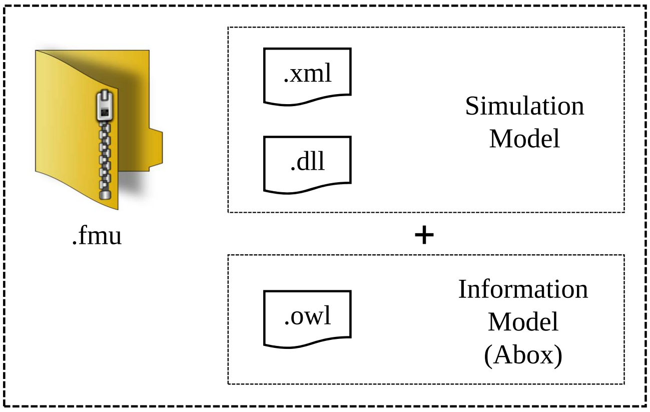


Figure 32. Extension of the standard FMU content with an owl file. The content represents the Abox of a knowledge description with associations to the overarching ontology, i.e. the Tbox.

The content of the *owl* file relies on a definition of a resource identifier for the FMU-specific Abox, which is based on the name of the model inherited in the FMU. Using this identifier, instances of input and output variables are defined. These instances are associated with the concepts from *FMUOnt*, i.e. the previously described classes and individuals are associated with the Abox instances. An example for a resulting variable description is provided in Table 1. The FMU instance *OfficeBuilding* and the variable *T_air* as FMU-specific components are defined inside the Abox through the FMU-specific resource identifier. Opposed to this, the remaining object properties and attribute instances are merely referenced through their URIs defined in the overarching ontology *FMUont*.

Table 1. Example of an annotated variable inside the Abox of a building envelope FMU.

FMU	OfficeBuilding
OutputVariable	T_air
hasMedium	Air
hasUnit	C
hasQuantity	Temperature
hasSemanticType	Space Condition

3.3 Single-Zone Simulation

Section 2.1.1 illuminated the principles of a single-zone simulation. As described, such a model is characterized through the reduction of a building envelope to one thermal inertia which responds to influences by its surroundings. The remaining domains from section 3.1.1 can act as these influences. As such, they provide models inheriting the influence on the entirety of a building and correspondingly rely on the respond of the building envelope as a whole. These interactions are uniquely identified through the previously discussed descriptions. Similarly to a registry unit in SOA, an entity is required to match the input and output variables embodying these interactions.

As mentioned in section 2.4, reasoners are generic algorithms able to generate new information from an existing knowledge representation. In the context of the developed methodology for modular BPS, reasoners are used in the sense of a registry unit. The formulation of tailored rules in SWRL enables a reasoner to infer the desired information about the connections of variables from selected FMUs.

In the case of a single-zone simulation, the rule presented in Listing 3 is applied on the knowledge representation of simulation modules. It compares the object properties *Medium*, *Quantity* and *SemanticType* of all input and output variables in order to infer the property *isConnectedWith* for the corresponding variable combination.

Listing 3. SWRL rule to infer the connections between variables for a single-zone simulation.

```
hasInputVariable(?FMUinput,?InputVariable),
hasOutputVariable(?FMUoutput,?OutputVariable),
hasQuantity(?InputVariable,?QuantityInput),
hasMedium(?InputVariable,?MediumInput),
hasSemanticType(?InputVariable,?SemanticTypeInput),
hasQuantity(?OutputVariable,?QuantityOutput),
hasMedium(?OutputVariable,?MediumOutput),
hasSemanticType(?OutputVariable,?SemanticTypeOutput),
SameAs(?QuantityInput,?QuantityOutput),
SameAs(?MediumInput,?MediumOutput),
SameAs(?SemanticTypeInput,?SemanticTypeOutput)
-> isConnectedWith(?InputVariable,?OutputVariable)
```

Considering the nature of meta-data required to derive the connections for a single-zone simulation, it can be concluded that two layers of information, as depicted in Figure 33, are required. First, basic information about the quantity and medium is necessary. Furthermore, information about the unit of the variable is required in order to perform conversions during the coupling process. Placeholders for these base properties are already provided in the model description file, as discussed in section 3.1.3. In addition to that, domain specific information providing meaning to a variable within a defined context is necessary. In this case, semantic types corresponding to the chosen decomposition strategy of a BPS inherit this information.

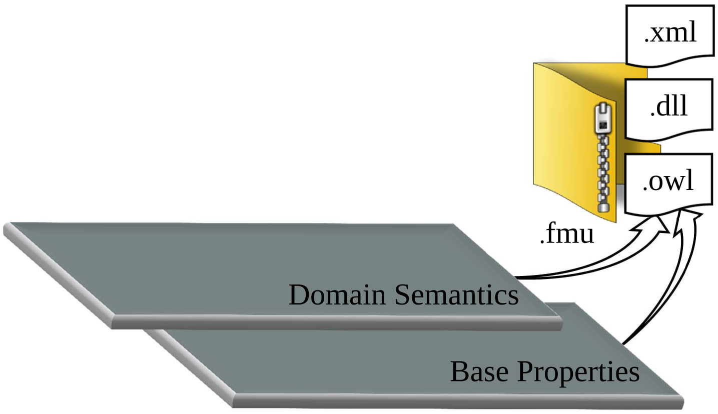


Figure 33. The required information layers to connect simulation modules in a single-zone simulation. The relevant FMUs should contain information about base properties of variables as well as domain context information.

3.4 Multi-Zone Simulation

In contrary to above, a multi-zone simulation subdivides a building into several thermal zones. Each zone can feature different influences or boundary conditions from other domains. Hence, also models from other domains can be individualized for each thermal zone, reacting to the conditions of the particular zone.

Table 2 shows the annotation of two variables in a building envelope FMU that inherits a multi-zone model. The two variables *T_Office1* and *T_Office2* represent air temperatures of two different thermal zones. When considering the queried information in the single-zone case, i.e. quantity, medium and semantic type, it becomes apparent that no differentiation between the variables is possible. The same issue occurs for every zone-related variable. Another example are variables representing heat flows from several individual radiators in an HVAC FMU. While featuring an identical description they are all meant to provide heat to different zones depending on their location in the building. Therefore, for a multi-zone simulation spatial information is required to automatically infer the connections between input and output variables.

Table 2. Example of two annotated variables inside the Abox of a building envelope FMU representing a multi-zone model.

FMU	OfficeBuilding	
OutputVariable	T_Office1	T_Office2
hasMedium	Air	Air
hasUnit	C	C
hasQuantity	Temperature	Temperature
hasSemanticType	Space Condition	Space Condition

In order to accommodate spatial information in the overarching ontology, an extension, realized through to the red-marked additions in Figure 34, is required. The focus of these changes is the introduction of a class *Space*. *Thermal Zone* is a subclass of *Space* allowing for the instantiation of thermal zones within the simulation. Each FMU can provide these instances through the object property *hasSpace*. Similarly, each variable can have an affiliation to instances of this class through the *hasZoneAffiliationTo* attribute.

The information, to which thermal zone a variable is associated to, can not be provided by distributed stakeholders. At the stage when they contribute with individual modules to the simulation, knowledge about other modules, e.g. number, name, scope etc. of the modeled thermal zones is not available to the separated actors. As discussed in section 2.2.3, in this case a digital representation of a building corresponding to the paradigms of BIM can serve as an information pool to each stakeholder. Section 2.4 further indicated two efforts [97, 99], which successfully converted widely accepted BIM standards to a formulation in OWL. This allows for connecting a central building data model to the overarching data framework, serving as a mediator to provide the required information for instantiation of a multi-zone simulation. Variables can be attached to a topology element in the building data model via the object property *hasAffiliationTo*. This association is not intended to directly connect a variable to a spatial object, i.e. a room or a thermal zone. Instead, variables are attached to objects that directly reflect their modeled entity, e.g. a variable providing information about heat flow from a radiator to a thermal zone is directly attached to the corresponding radiator object in the digital representation of the building. The resulting association to a space can be retrieved via queries in the building data model. This ensures flexibility regarding the change of the determined spatial enclosure of a room or thermal zone. In such a case, updating each variable description with the new spatial information can be avoided. Instead a repetition of the query automatically derives the new spatial association.

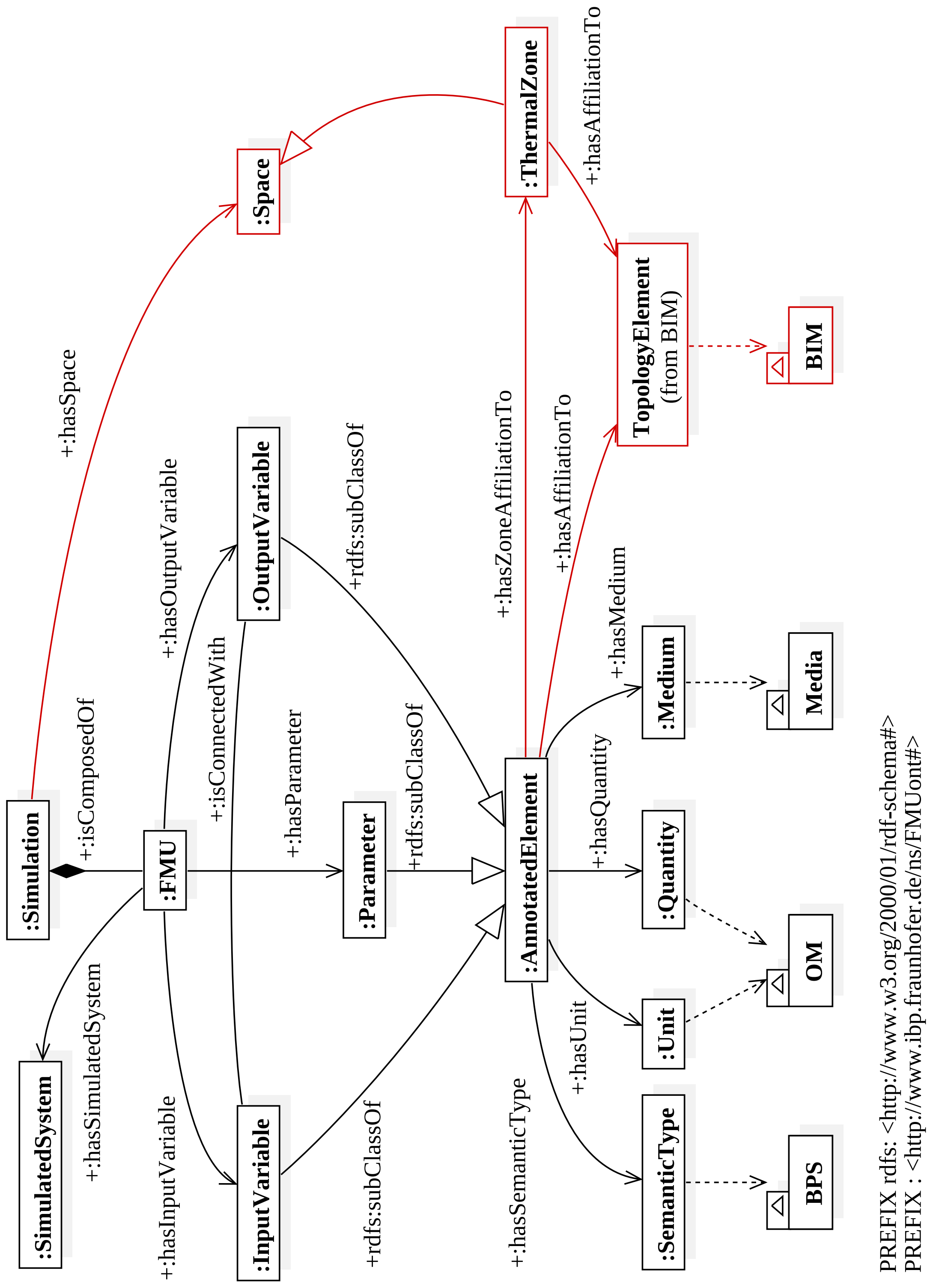


Figure 34. The taxonomy of FMUont with the integration of zone-related information. The red-marked changes present additions made in order to accommodate associations to different thermal zones in the overarching ontology.

Figure 35 shows an excerpt of *FMUont* integrated with BIM. For the purpose of demonstration, elements from the SimModel format are used. Prefixes indicate the origin of the information and allow for differentiation of data contained in *FMUont* or the building data model. A building envelope FMU is described with its thermal zone instances. These instances are connected with their corresponding zones in the building data model, all of which being instances of the same class in SimModel. Additionally, a variable originating from an HVAC FMU describing heat flow from a convector is defined and connected with its representing object in the building data model. This object is an instance of the SimModel class for convectors. Corresponding to the ontology of SimModel, it features relations to several other classes in order to describe the building. Accordingly, the convector instance is connected to several instances of these classes. The resulting network in the building data model also includes a path to the associated thermal zone of the convector instance, in this case called *Office_Heart*. Through the previously added connection of this object to a thermal zone in *FMUont*, the variable's relevant spatial association for the simulation can be derived. In Figure 35, this path is indicated in red.

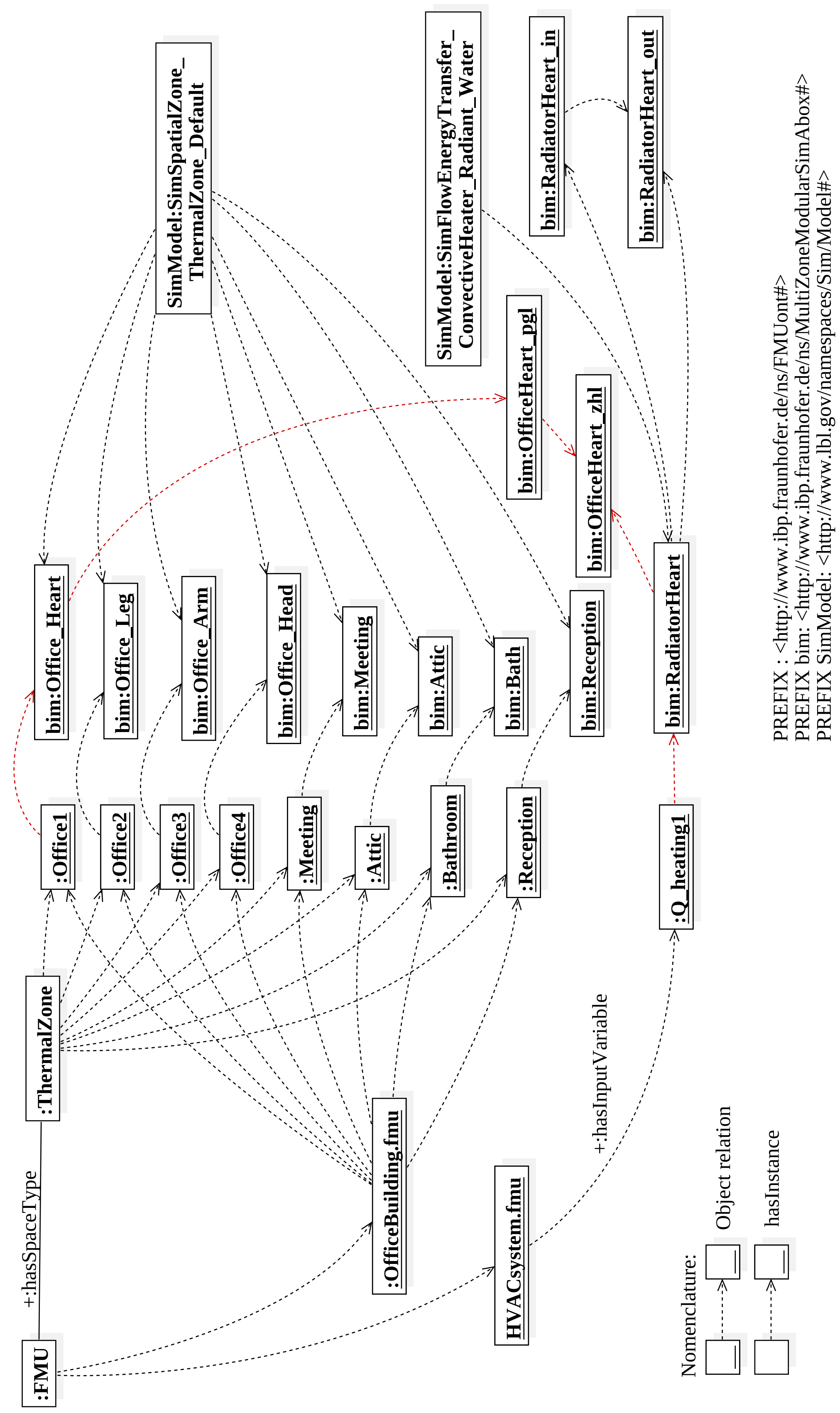


Figure 35. The integration of BIM data with FMUont. The red-marked connections illustrate the path for deriving a zone association of a variable.

Regarding the automated retrieval of the zone information, queries formulated with SPARQL, as elaborated in section 2.4, can be formulated. This requires a definition of the path, described through object properties and classes, from the variable in question to the *Thermal Zone* class in *FMUont*. Since numerous different objects of a building data model may be associated with simulation variables, this requires the pre-definition of numerous paths for the query. Depending on the class of the associated BIM object, the corresponding query can be selected and executed.

In order to reduce this effort, a more generic solution, applicable for each BIM object class, is favored. The realized approach is based on algorithms in the field of graph theory targeting the challenge to identify shortest paths in complex networks. An example for such work is the Dijkstra-algorithm as originaly described in [30]. The method computes the shortest path between nodes while considering the distance between nodes. This algorithm is in the following applied to the search of the correct thermal zone a variable needs to be associated with. A pseudo-code representation of the implementation is presented in Listing 4.

The algorithm starts with the associated instance of a variable in a building data model. In a first step, it determines the neighboring instances. This can be any instance associated to the variable via a connecting object property. This requires a search for all object properties that include this variable and determine the corresponding partner instance. The found neighbors are checked for instances of the class *thermal zone*. If successful, the found thermal zone is returned, otherwise the neighbors are appended to search paths that describe the hitherto tested network routes. The procedure for determining the neighbors of instances, as well as their check for a *thermal zone* object, is repeated for the last elements in the evolving search paths. If an instance is found that has already been under consideration in other paths, this path is no longer followed. A successful check for a *thermal zone* instance ends the process and returns the corresponding object representing the targeted zone.

It has to be noted that this process also allows for instances to have connections via several objects and properties to other thermal zones. However, the correct thermal zone must feature the shortest connection in terms of number of objects in between. A solution for situations which do not correspond to this pre-requisite is the definition of path elements that are not to be followed within the algorithm.

Listing 4. Dijkstra-algorithm to detect thermal zones associated to a variable.

```
getNeighbors(BIMobject)
checkNeighbors(neighbors)
if thermal zone found
    associate variable with thermal zone
else
        for neighbor in neighbors and path in paths
        append neighbor to path
    while zone not found
        updateNeighbors(paths)
        for path in paths
            append last element to newNeighbors
```

```
        checkNeighbors(newNeighbors)
        if thermal zone found
            associate variable with thermal zone

function getNeighbors(node)
    for instance in instances
        for property in properties(instance)
            if connected instance is node
                append instance to neighbors
    for property in properties(node)
        append connected instance to neighbors
    return neighbors

function updateNeighbors(paths)
    for path in paths
        getNeighbors(last instance in path)
            for neighbor in neighbors
                if neighbor not in paths
                    append neighbor to path
    return paths

function checkNeighbors(neighbors)
    get thermal zones from ontology
    for neighbor in neighbors
        for thermal zone in thermal zones
            if neighbor is thermal zone
                return thermal zone
```

In order to infer the connections between variables through reasoning, the SWRL rule from section 3.3 for a single-zone simulation must be extended. In addition to the hitherto used variable properties, the spatial information must be considered. This is achieved through integration of the *hasZoneAffiliationTo* property into the SWRL rule as formulated in Listing 5. However, not each variable can be associated with a thermal zone, since some are of a more global nature. These variables originate from the environment domain and can be identified through their corresponding semantic types. Hence, two rules targeting the weather and location data are additionally formulated ignoring zone-related information.

Listing 5. SWRL rule to infer the connections between zone-related variables in a multi-zone simulation.

```
hasInputVariable(?FMUinput,?InputVariable),
hasOutputVariable(?FMUoutput,?OutputVariable),
hasQuantity(?InputVariable,?QuantityInput),
hasMedium(?InputVariable,?MediumInput),
hasSemanticType(?InputVariable,?SemanticTypeInput),
hasZoneAffiliationTo(?InputVariable,?ZoneAffiliationInput),
hasQuantity(?OutputVariable,?QuantityOutput),
```

```
hasMedium(?OutputVariable,?MediumOutput),
hasSemanticType(?OutputVariable,?SemanticTypeOutput),
hasZoneAffiliationTo(?OutputVariable,?ZoneAffiliationOutput),
SameAs(?QuantityInput,?QuantityOutput),
SameAs(?MediumInput,?MediumOutput),
SameAs(?SemanticTypeInput,?SemanticTypeOutput),
SameAs(?ZoneAffiliationInput,?ZoneAffiliationOutput)
-> isConnectedWith(?InputVariable,?OutputVariable)
```

Listing 6. SWRL rule to infer the connections between variables describing weather data in a multi-zone simulation.

```
hasInputVariable(?FMUinput,?InputVariable),
hasOutputVariable(?FMUoutput,?OutputVariable),
hasQuantity(?InputVariable,?QuantityInput),
hasMedium(?InputVariable,?MediumInput),
hasSemanticType(?InputVariable,?WeatherData),
hasQuantity(?OutputVariable,?QuantityOutput),
hasMedium(?OutputVariable,?MediumOutput),
hasSemanticType(?OutputVariable,?WeatherData),
SameAs(?QuantityInput,?QuantityOutput),
SameAs(?MediumInput,?MediumOutput)
-> isConnectedWith(?InputVariable,?OutputVariable)
```

Listing 7. SWRL rule to infer the connections between variables describing location data in a multi-zone simulation.

```
hasInputVariable(?FMUinput,?InputVariable),
hasOutputVariable(?FMUoutput,?OutputVariable),
hasQuantity(?InputVariable,?QuantityInput),
hasMedium(?InputVariable,?MediumInput),
hasSemanticType(?InputVariable,?Location),
hasQuantity(?OutputVariable,?QuantityOutput),
hasMedium(?OutputVariable,?MediumOutput),
hasSemanticType(?OutputVariable,?Location),
SameAs(?QuantityInput,?QuantityOutput),
SameAs(?MediumInput,?MediumOutput)
-> isConnectedWith(?InputVariable,?OutputVariable)
```

As discussed in this section, the information requirement to derive the simulation topology in a multi-zone simulation includes the spatial association of variables. This information is of project-specific nature, possibly even varying during a single design process. Consequently, in contrary to the information levels identified for a single-zone simulation, this information can not be carried in the individual FMUs in order to enable their application beyond the scope of a single project. Instead, for each project, individualized information must be provided to the FMUs via BIM. This leads to an additional information layer as depicted in Figure 36. With regards to the re-usability of simulation modules beyond single projects, information from this layer can not be added to the Abox contained inside FMUs.

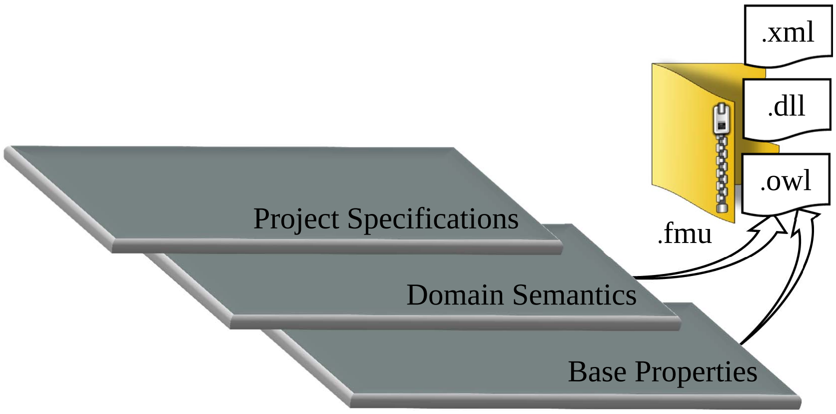


Figure 36. The required information layers to connect simulation modules in a multi-zone simulation. Information about base properties and semantics within a defined context needs to be extended with project specific information.

3.5 Zonal Airflow Simulation

An assessment of indoor climate, especially when considerable differences of thermal conditions at different locations in an air volume occur, requires the computation of air movement within this enclosed space. In contrary to the multi-zone simulation, a zonal airflow computation is able to predict the temperature distribution within an air volume, as described in section 2.1.3. The application of a zonal airflow model to the entire building envelope is technically feasible, however, due to high implementation and computation time, it is generally only applied to single parts of a building. Hence, the integration of such a model into the modular simulation does not lead to an exchange of the building envelope module, as realized previously for the single- and multi-zone model. Instead, it serves to assess one or more rooms of the building envelope with greater detail. The isolation of the remaining building parts requires the co-existence of a single- or multi-zone model in order to capture the dynamics and interactions with other spaces as well as their influence on the common supply systems. Simultaneously to the higher model resolution of the building envelope, models from the remaining domains need to increase their LOD as well. As presented in sections 3.3 and 3.4, the increased spatial resolution allows to distribute variables to single zones and distinguish between them. With the subdivision of a thermal zone or a room of a thermal zone into several air-cells, variables can now be associated to a specific location in a room. Hence, for example several radiators or people in a room can be considered individually if the corresponding functionality is provided by their modules. This functionality requires that e.g. aggregated internal loads of a thermal zone are partitioned into single sources that may or may not reside in the room under consideration enabling a local association of internal loads with positions within the room.

The in section 3.4 applied concept to enable a differentiation of variables in a multi-zone simulation benefits from the capability of BIM formats to subdivide buildings into several spaces and the association of building elements to these spaces. A similar concept is applied for the inclusion of a zonal airflow model into the simulation, as highlighted in the adapted overarching ontology *FMUont* in Figure 37. Thermal zones may consist of several rooms. In order to distinguish between these rooms a class *Room* is introduced. This class is a subclass of *Space* and can be related to an *Annotated Element*, an *FMU* and an object from a digital building representation similar to the *Thermal Zone* class, as described in section 3.4. Additionally, the relation *belongsTo* allows for associating rooms to thermal zones within the simulation ontology.

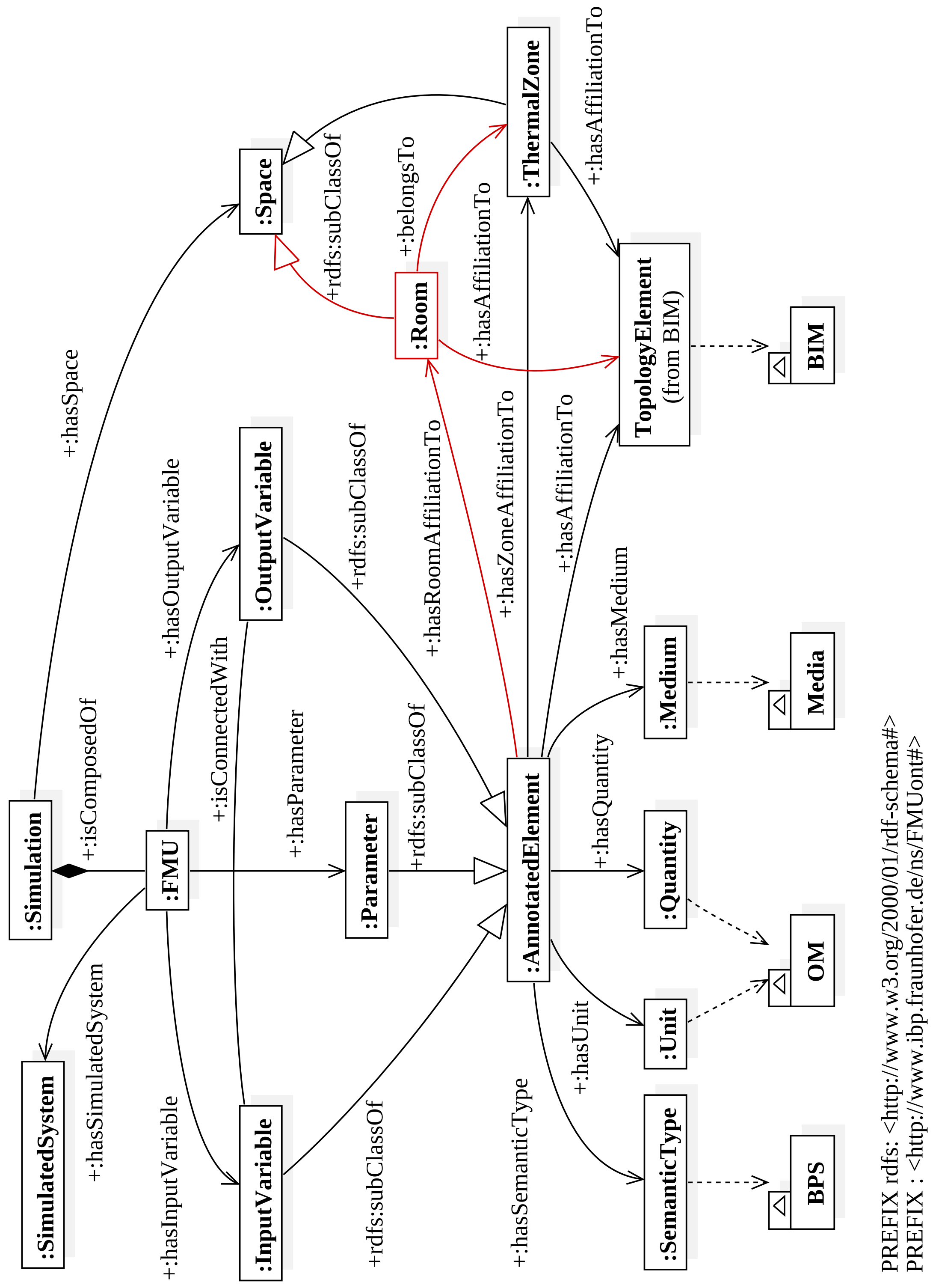


Figure 37. The taxonomy of FMUont with the integration of room-related information. The red-marked changes present additions made in order to accommodate associations to different rooms in the overarching ontology.

Similarly to the multi-zone concept, associations of variables with a room need to be derived from the knowledge base. The approach follows the principles of the procedure for thermal zones, as explained in section 3.4. Variables from a zonal airflow module are connected to objects from a BIM. This allows for detecting a path to a room instance, as shown in Figure 35. For this purpose, the application of the Dijkstra algorithm can be repeated with an altered goal definition, i.e. instances of type *Room*.

Since the location of objects inside a room can be considered in a zonal airflow model, the association of variables to a room is not enough to distinguish between them. A further differentiation is necessary, for example to differ between several radiators in a room. The previously realized direct association of variables with their corresponding objects in a digital representation of a building enables this differentiation. Hence, for single rooms, computed as zonal airflow models, this information must be included in the reasoning process. Simultaneously, the co-existence of a model for the entire building envelope requires the derivation of variable connections for the remaining models as realized above. Hence, in addition to the SWRL rules formulated in section 3.4, the reasoning process includes two further rules, shown in Listings 8 and 9, which are tailored to the inference of connections for zonal airflow models of rooms. The first rule is aimed at the input variables of a zonal airflow model. It generates connections with other variables through comparison of base properties, semantic type and the associated BIM object. For the sake of a complete simulation ontology, matching output variables are additionally associated with the corresponding room. Similarly, the second rule infers connections for output variables of a zonal airflow model and attaches the room to the matching input variable.

Listing 8. SWRL rule to infer the connections between variables in the modular simulation with a zonal airflow model. This rule derives connections for inputs of the zonal airflow model and associates the connected outputs to the modeled room.

```
hasInputVariable(?FMUinput,?InputVariable),
hasOutputVariable(?FMUoutput,?OutputVariable),
hasQuantity(?InputVariable,?QuantityInput),
hasMedium(?InputVariable,?MediumInput),
hasSemanticType(?InputVariable,?SemanticTypeInput),
hasRoomAffiliationTo(?InputVariable,?RoomInput),
hasAffiliationTo(?InputVariable,?AffiliationInput),
hasQuantity(?OutputVariable,?QuantityOutput),
hasMedium(?OutputVariable,?MediumOutput),
hasSemanticType(?OutputVariable,?SemanticTypeOutput),
hasAffiliationTo(?OutputVariable,?AffiliationOutput),
SameAs(?QuantityInput,?QuantityOutput),
SameAs(?MediumInput,?MediumOutput),
SameAs(?SemanticTypeInput,?SemanticTypeOutput),
SameAs(?AffiliationInput,?AffiliationOutput)
-> isConnectedWith(?InputVariable,?OutputVariable),
hasRoomAffiliationTo(?OutputVariable,?RoomInput)
```

Listing 9. SWRL rule to infer the connections between variables in the modular simulation with a zonal airflow model. This rule derives connections for outputs of the zonal airflow model and associates the connected inputs to the modeled room.

```
hasInputVariable(?FMUinput,?InputVariable),
hasOutputVariable(?FMUoutput,?OutputVariable),
hasQuantity(?InputVariable,?QuantityInput),
hasMedium(?InputVariable,?MediumInput),
hasSemanticType(?InputVariable,?SemanticTypeInput),
hasAffiliationTo(?InputVariable,?AffiliationInput),
hasQuantity(?OutputVariable,?QuantityOutput),
hasMedium(?OutputVariable,?MediumOutput),
hasSemanticType(?OutputVariable,?SemanticTypeOutput),
hasRoomAffiliationTo(?OutputVariable,?RoomOutput),
hasAffiliationTo(?OutputVariable,?AffiliationOutput),
SameAs(?QuantityInput,?QuantityOutput),
SameAs(?MediumInput,?MediumOutput),
SameAs(?SemanticTypeInput,?SemanticTypeOutput),
SameAs(?AffiliationInput,?AffiliationOutput)
-> isConnectedWith(?InputVariable,?OutputVariable),
hasRoomAffiliationTo(?InputVariable,?RoomOutput)
```

The resulting set of rules derives the connections for all modules in the simulation, when a zonal airflow model is present in addition to the multi-zone model. However, the duplicated modeling of partial building space may lead to duplicated variable connections among the simulation modules. Reasoning based on SWRL is limited to infer new information from an ontology. The modification of existing information is not supported. Hence, the implementation, as is described in the following, prioritizes the connection with a higher LOD, i.e. the variables from the zonal airflow model.

With regards to the necessary information levels, it can be concluded that the introduction of a zonal airflow model can be realized based on the same information levels as previously defined for the multi-zone simulation. However, in order to consider effects on a room basis, the formulation of simulation models must ensure the required resolution to differ between influences at different locations in a room. Furthermore, the resulting differentiation of variables must be considered in the reasoning process.

3.6 Implementation

The implementation of the presented methodology for a scalable simulation based on modular components is realized using the Python programming language. It benefits from the capabilities of the *PyFMI* package [83] for operating FMUs as well as the *Owlready* package [66] for loading, creating and editing ontologies in OWL.
The implementation follows the scheme presented in Figure 38. It relies on several sources that provide FMUs within the defined context. These sources can be of different nature.

FMUs may be generated from own models, taken from libraries or obtained directly from manufacturers. Either FMU must be provided with an additional file inheriting the model description corresponding to the overarching data model. This extension is facilitated through a GUI called the *FMUOntologyXtender*, as depicted in Figure 39 and 40. The GUI allows one to load an FMU and annotate its variables as well as provide general information about the author of the model, the simulated system etc. The annotation is only necessary if the required information is not present inside the FMUs, as is the case for re-used library FMUs.

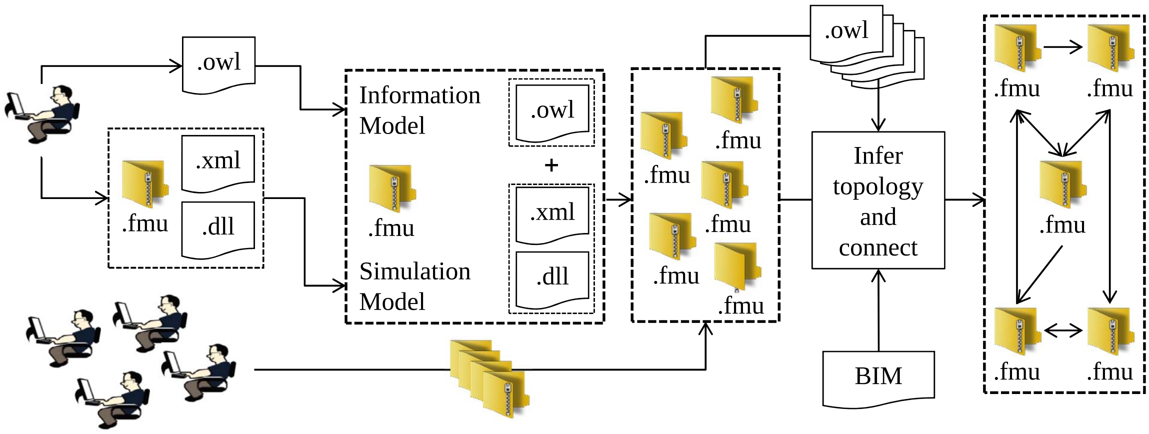


Figure 38. The implemented process to realize the modular simulation.

After a selection of FMUs is determined to form the simulation, an algorithm aggregates the Abox files in a simulation ontology. Duplicates of identifiers need to be avoided during this step. Hence, in order to allow for multiple instantiation of an FMU, a progressing identification number is attached to the loaded FMUs. At this stage, information about base properties and domain semantics is summarized in the simulation ontology. Subsequently, simulation, FMU and variable classes are instantiated in the implementation based on the Abox information of each FMU. A following query of the simulation ontology detects the LOD of the simulation, i.e. the presence of single-zone, multi-zone or zonal airflow models. If information on project level is required for the derivation of the simulation topology, a connection to objects in a digital representation of the building needs to be established at this point. Therefore, the digital building representation is loaded and each exchange variable can be mapped to its corresponding BIM object. This allows for applying the in section 3.4 presented form of the Dijkstra algorithm to each variable in order to retrieve their association to thermal zones or rooms respectively. Hence, an SWRL rule corresponding to the determined LOD, is attached to the simulation ontology completing its basic setup.

A reasoner is now able to infer the connections between variables. Therefore, the *HermiT* reasoner [85] is invoked and set up through Java. The reasoning returns the inferred *isConnectedWith* object properties and updates the simulation ontology. This information is subsequently processed by the algorithm to derive the connections for the simulation. At this stage, the connections are evaluated regarding duplicates and adjusted corresponding to their LOD, as mentioned above. Optionally, parameters of the contributing FMUs can be set before initializing the simulation.

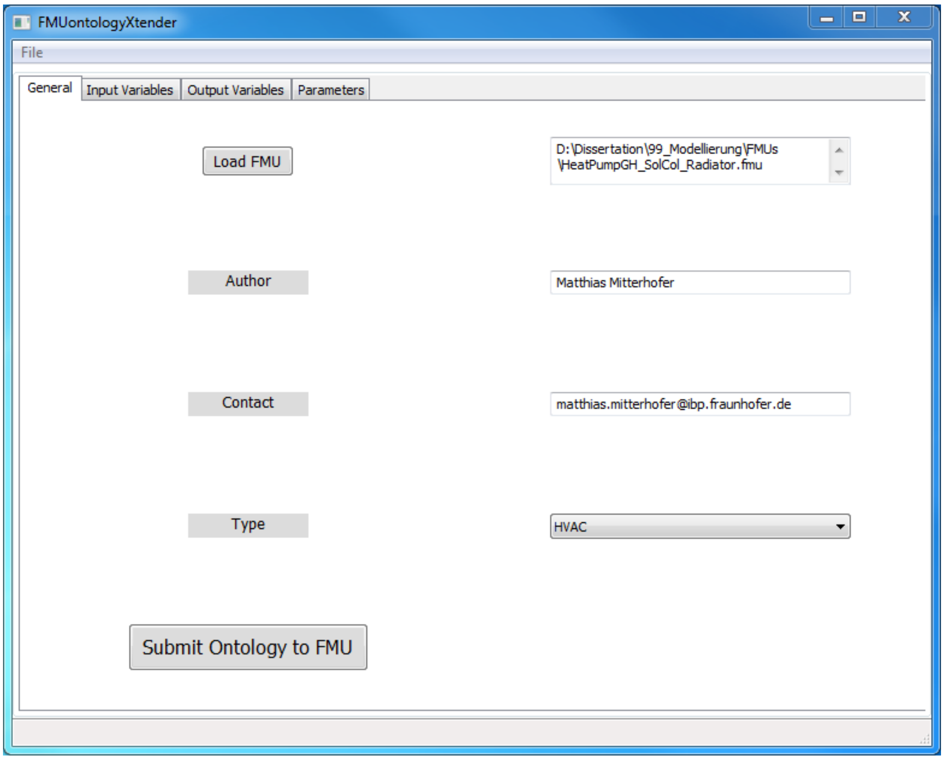
The co-simulation is realized with the loose coupling algorithm in parallel execution, as explained in section 2.3.2. In accordance with the recommendations from [119] and due to the focus of the implementation being on the feasibility of the methodology, its advantages regarding faster run-time and ease of implementation were prioritized. At each time step, current variable values are exchanged according to the inferred connections and the single FMU states progress with the updated inputs. In the case of multiple outputs connected to a single input variable, an aggregation is realized. This aspect is discussed in more detail in sections 4.1 and 5.1.3.

Listing 10 provides a pseudo-code representation of the described implementation. A representation of the implemented classes and methods in the Unified Modeling Language (UML) is provided in appendix A.

Listing 10. Pseudo-code representation of the implemented algorithm to realize the modular simulation.

```
for each FMU in simulation
        import Abox description into simulation ontology
        generate ID

if number of thermal zones is one
        LOD is single-zone
elif number of thermal zones is greater one
        LOD is multi-zone
        for each FMU
                if simulated system is zonal airflow
                        zonal airflow is true

if LOD is multi-zone
        import BIM ontology
        for each FMU
                for each variable in FMU
                        connect variable to BIM object
                        execute Dijkstra

if LOD is single-zone
        import single zone rule
elif LOD is multi-zone
        if zonal airflow is true
                import zonal airflow rules
        else
                import multi-zone rules

execute reasoner

```

```
for each FMU in simulation ontology
        for each input variable in FMU
                get connected output variables
                if output variables from zone and room connected
                        delete connection to variable from zone

while timestep in simulation time
        if timestep is one
                set input variables to start values
        else
                for each connection
                        if multiple variables connected
                                perform aggregation
                        perform unit conversion
                        append value to input variable
                        set input variable to current value

        for each FMU
                simulate FMU timestep
                for each output variable in FMU
                        get current value of output variable
        advance timestep
```



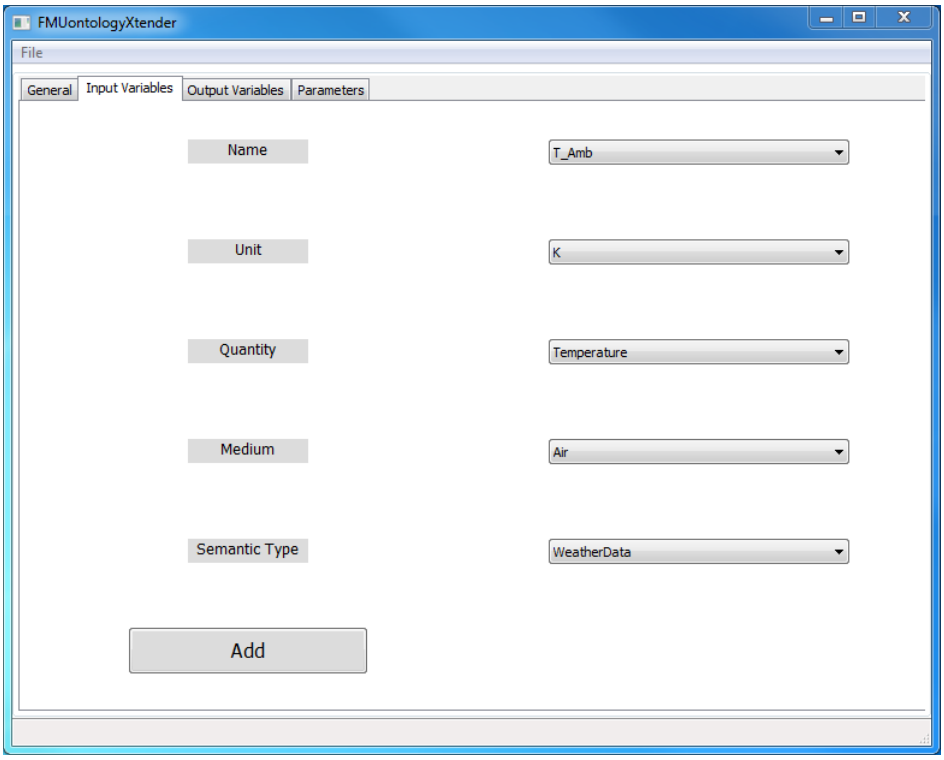
D:\Dissertation\99_Modellierung\FMUs\HeatPumpGH_SolCol_Radiator.fmu

Author: Matthias Mitterhofer

Contact: matthias.mitterhofer@ibp.fraunhofer.de

Type: HVAC

Figure 39. The FMUOntologyXtender for annotation of FMUs. In the first tab general information is provided, including the author of the model, the type of the simulated system etc.



Name: T_Amb

Unit: K

Quantity: Temperature

Medium: Air

Semantic Type: WeatherData

Figure 40. The FMUOntologyXtender for annotation of FMUs. In the latter tabs variables can be described corresponding to the overarching ontology.

Chapter 4

Case Studies

Application of the developed methodology is in the following illustrated in two case studies. The first example features a building on the test site of the Fraunhofer IBP in Holzkirchen, Germany. This study aims at demonstrating the characteristics and advantages of the modular simulation regarding seamless module exchange and scalability throughout the design process, while maintaining an integral view in the simulation. The supporting digital representation of the building is generated using the SimModel format. As mentioned in section 2.4, its formulation can be realized in the required OWL. In addition to this, the second example deploys a building's description formulated in the IFC data format. It benefits from the in [99] developed procedure to convert an IFC scheme to OWL. The example building in this study is taken from [60], where a collection of building representations in the IFC format is provided by the Karlsruher Institute of Technology. The goal of the second case study is to demonstrate the independence of the developed methodology from the chosen format for the digital representation of a building and specifically proofs compliance with the IFC data scheme.

In order to reference the executed simulations in the case studies, a numbering system, embedded in the corresponding section titles, is introduced for both case studies. The in Listing 10 described loose coupling algorithm is applied for each simulation at a time step of two minutes. Illustrations of the resulting knowledge representation graphs are generated with Protegé [112], an application for editing and viewing ontologies.

4.1 Example 1 - The Twin House

The test facilities of the Fraunhofer IBP comprise, among others, two identical buildings, called the Twin Houses. In the following case study, one of these two houses, as depicted in Figure 41, serves as the example building for the application of the developed methodology for a scalable simulation. In the study, its usage for office purposes and the selection of a corresponding heating system is investigated.

The building consists of three floors, each in the dimension of ten by ten meters, as shown in the floor plans in Figures 42 to 44. In the example study, the projected purpose of the basement is the accommodation of technical systems, e.g. HVAC components, such as thermal storage. The first floor, corresponding to its distributed layout, includes offices and common areas, such as bathroom and lounge. The second floor is determined to provide space for a meeting room.

Figure 41. The Twin Houses at the test site of the Fraunhofer IBP in Holzkirchen, Germany.

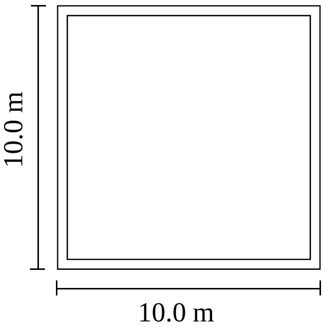


Figure 42. Basement of the Twin House.

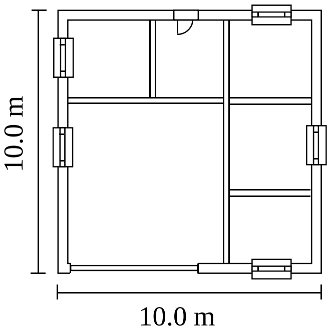


Figure 43. First floor of the Twin House.

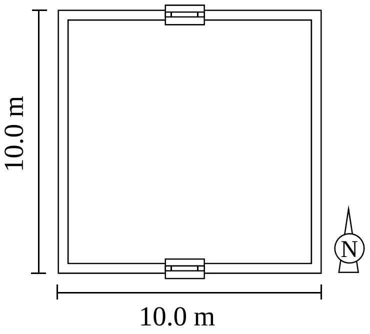


Figure 44. Second floor of the Twin House.

4.1.1 Single-Zone Simulation

To perform a first estimation of the building's heating energy demand, a single-zone model of the building envelope is developed. The generation of building envelope models within this thesis is exclusively realized with the in [42] presented plugin for the 3D modeling software SketchUp [121], which allows to directly create EnergyPlus models from CAD. As mentioned in section 3.1.1, opposed to the other domains, building envelope FMUs might have to be regenerated several times during the design process since parameterization can not be applied as flexible as for the other domain modules. In order to facilitate this task, a further plugin was implemented that enables the generation of EnergyPlus FMUs within the CAD environment provided by SketchUp. FMI support of Energy Plus is restricted to the tool co-simulation of version 1.0. Hence, a local installation of Energy Plus is required to run these FMUs. Due to a further particularity, the weather data must be included in the EnergyPlus FMU, i.e. the weather module is incorporated in the building envelope module. Provided that domain boundaries are still respected, the methodology is able to accommodate this particularity. With the exception of the building envelope FMUs, the FMUs applied in the remainder of this thesis correspond to the specifications of FMI version 2.0, generated and exported in Dymola.

Run 1.1 - Ideal Heating with Single-Zone Model

In a first simulation, a model representing an ideal heating system is coupled to the single-zone FMU. This model reacts to a given setpoint and measurement signal and provides a corresponding heat flow rate to the building. A third simulation module represents the people inside the building and as such provides internal loads through their presence, e.g. heat and CO_2. Additionally, a time-varying temperature setpoint for working and non-working hours is included.

Using the FMUontologyXtender presented in section 3.6, each FMU is extended with an Abox containing the description of input and output variables corresponding to the overarching ontology *FMUont*. As an example, Figure 45 illustrates the Abox of the ideal heating FMU. Each Abox has an instance of the class *FMU*, in this case the *IdealHeating* object. With this instance, the simulated system *HVAC* is associated. Furthermore, the FMU instance has the input variables *T_set* and *T_measured* as well as an output variable *Q_heating*. The corresponding object properties are applied to associate the variables with instances of unit, medium, quantity and semantic type, which are defined in the overarching *FMUont*. As an example, the variable *T_set* is connected to the quantity instance *Temperature* via the object property *hasQuantity*. The variable *T_measured* has the same quantity. Hence, the object property *hasQuantity* of this variable points to the same quantity instance, i.e. *Temperature*. A difference between these two variables poses the associated semantic type. While *T_set* is associated with the semantic type *Setpoint*, *T_measured* is associated with *Space Condition*.

Following the steps listed in the algorithm in Listing 10, the Abox of each FMU is imported into a single simulation ontology. Figure 46 shows an excerpt of the resulting data scheme. The class *FMU* now features three instances. These comprise an FMU incorporating the building envelope of the Twin House as a single-zone model, an FMU representing the people domain and the previously described FMU incorporating an ideal heating system. In order to allow for using an FMU, and hence importing its Abox several times during a single simulation, elements of each imported Abox receive an identification number. The remainder of this case study provides further clarification regarding this aspect.

Each simulation module has input and output variables. In order to maintain readability, only a selection of variables is shown in the figure. The building envelope FMU is expanded to show the output variable *T_Air_ TwinHouse*. From the ideal heating FMU the input variables *T_measured* and *T_set* are expanded. Similarly to above, the description

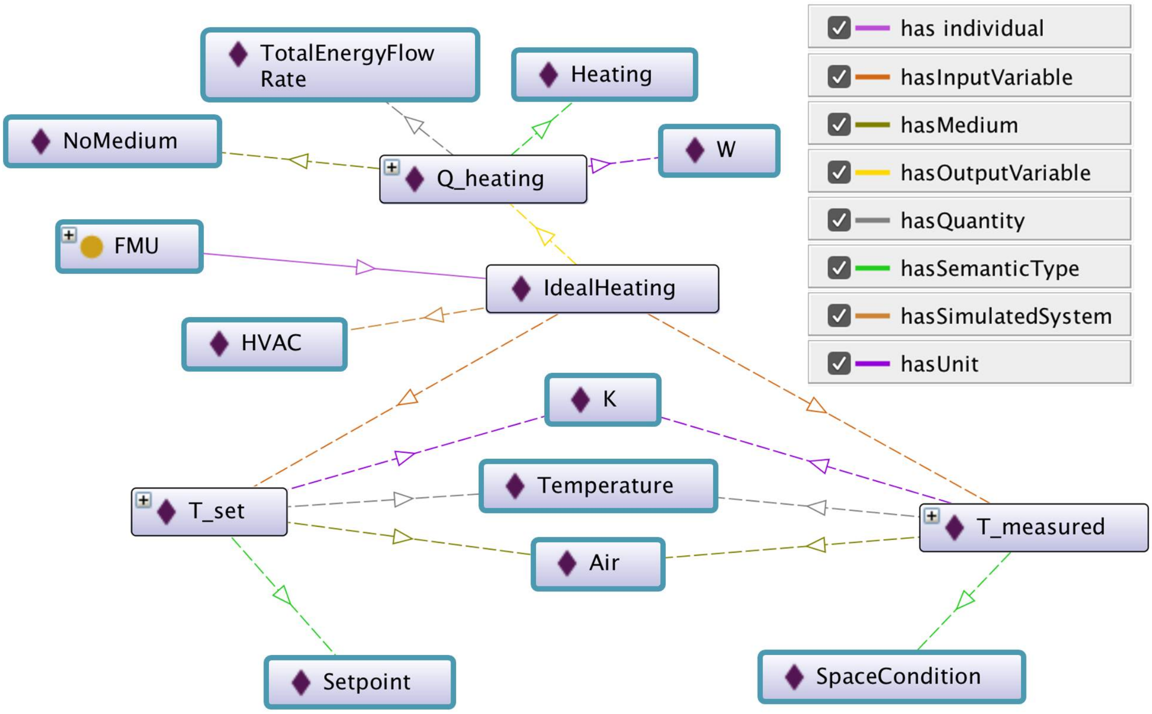


Figure 45. Individuals in the Abox of the ideal heating FMU in context with FMUont. The instances and classes defined in the overarching ontology FMUont are framed in blue. The remaining instances are locally defined in the Abox using the classes from FMUont. Their description is complemented through object property associations to the instances in FMUont. (yellow circles indicate classes; purple diamonds indicate individuals)

of these three variables is shown. It can be seen that the object properties of *T_measured* and *T_Air_TwinHouse* refer to the same instances of type medium, quantity and semantic type from the overarching ontology. *T_set* has a similar description, however, it differs in the associated semantic type instance.

At this stage, a reasoner is able to benefit from the resulting object relations in the simulation ontology. Corresponding to the SWRL rule presented in Listing 3, the reasoner screens every FMU for input and output variables, as defined in the first two lines of the code snippet. For each variable the associated quantity, medium and semantic type are retrieved corresponding to lines 3-10. These properties are compared for every variable pair. In the case of identical properties, the variables are connected through the *isConnectedWith* statement from *FMUont*. Hence, for the simulation ontology partly shown in Figure 46, the reasoner detects a connection between the variable *T_Air_TwinHouse* and *T_measured*. In total, among 199 possible connections corresponding to equation 3.5, five connections between the three FMUs are inferred.

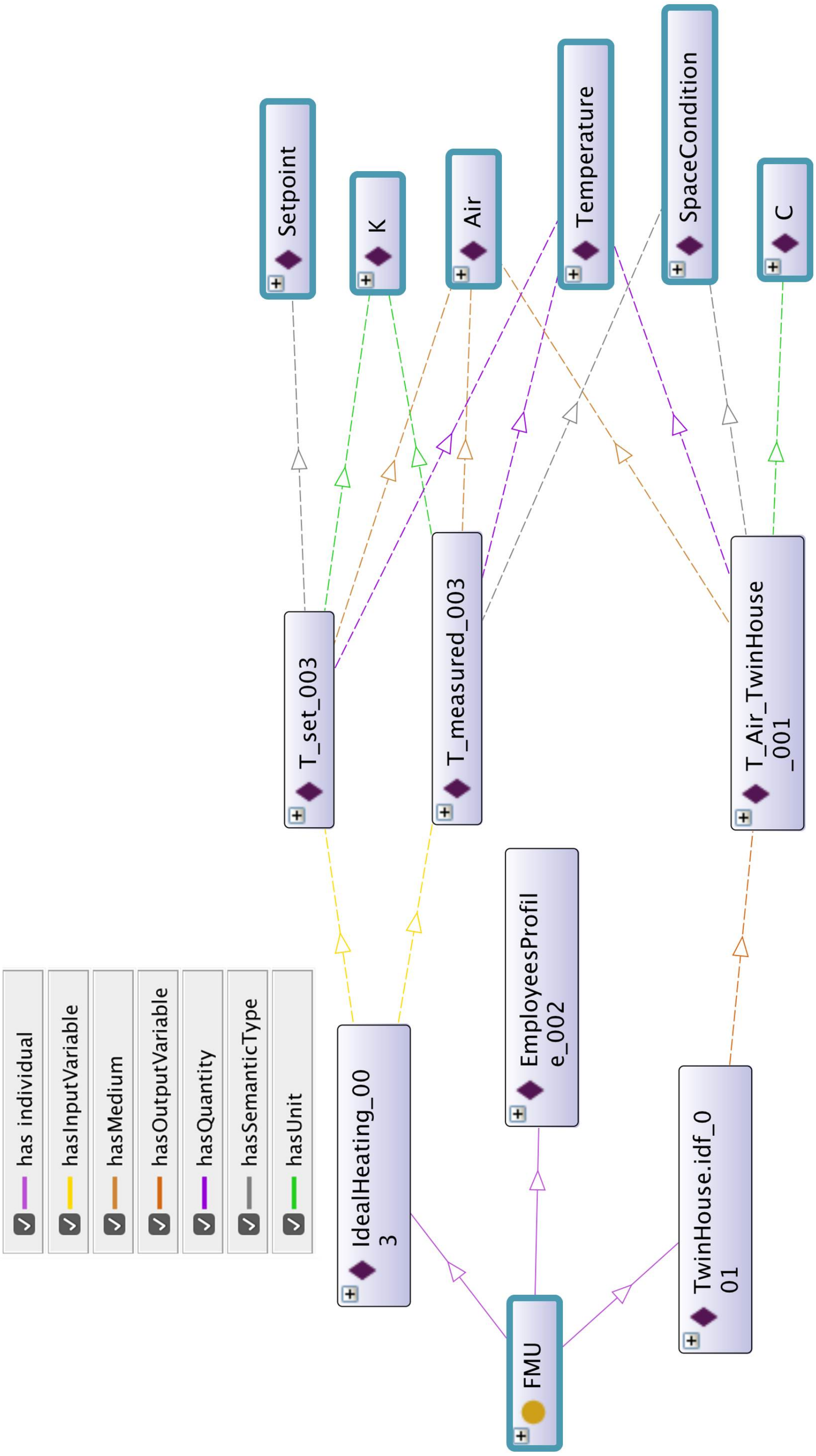


Figure 46. An excerpt of the simulation ontology in run 1.1 after the import of Aboxes from all three FMUs. The instances and classes defined in FMUont are framed in blue. (yellow circles indicate classes; purple diamonds indicate individuals)

Figure 47 shows the simulation result for two winter weeks. It depicts the setpoint and simulated temperature inside the building along with the required heating rate from the ideal heating system. It demonstrates that the heat load peaks in the morning when the setpoint temperature rises in order to provide a sufficient temperature level during working hours. The maximum heat load amounts to 7,285 W.

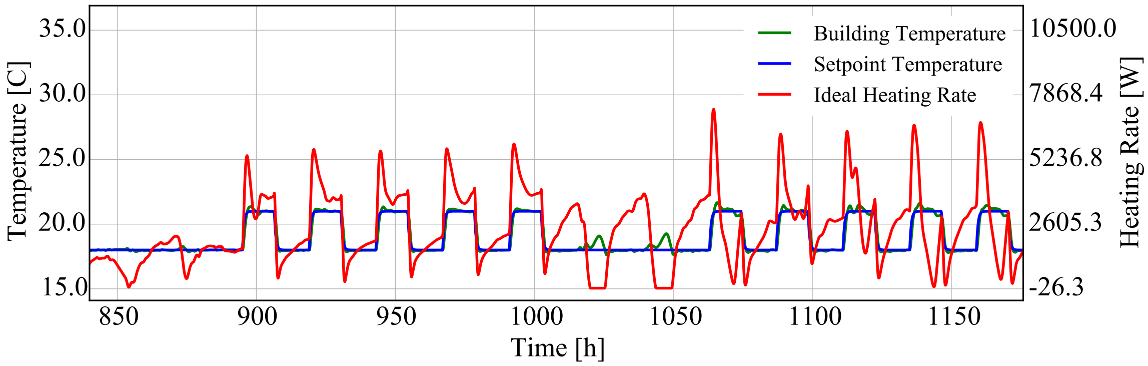


Figure 47. Air temperature and setpoint in simulation run 1.1 for two winter weeks in the building at the ideal heating rate.

Run 1.2 - Ground Heat Pump and Solar Collector with Single-Zone Model

In a next simulation run, the FMU representing the ideal heating system is exchanged with a library model that incorporates a ground heat pump as primary and a solar collector as a supporting secondary heat source. Both provide a storage unit with thermal energy in order to maintain a temperature of 55 °C in the tank. Both, heat pump and solar collector are generic models that can be specified with product specific properties taken from product data sheets. Table 3 shows a range of the applied parameters. In order to comply with the maximum heating load computed above, a heat pump with a nominal heating power of 7,350 W is selected.

Activation of the heat pump is controlled with a hysteresis, which starts the heat pump at a temperature of 5 °C below the tank design temperature and ends operation after reaching a temperature 5 °C above the tank design temperature. Additionally, three daily off-periods in the morning, midday and evening are defined in order to relieve the electrical grid in these high-consumption periods. Compressor power of the heat pump is set to generate a 5 °C rise in the condenser. The solar loop is activated when the fluid temperature in the collector exceeds tank temperature by 10 °C until a maximum of 70 °C is reached in the tank. Heat transfer to the building is realized with radiators, which are controlled with thermostats.

The simulation modules representing the people and building envelope domain from run 1.1 remain identical. Nevertheless, the exchange of the HVAC FMU increases the number of possible connections to 1,476. Besides new input variables in the HVAC FMU like ambient temperature and solar irradiation, this is due to each radiator providing a heating rate and

Table 3. Applied performance data for heat pump and solar collector.

Heat Pump	Nominal COP	4.5
	Nominal Heating Rate	7,350 W
Solar Collector	Maximum Efficiency C_0	0.8
	First Order Coefficient C_1	3.5 $\frac{W}{m^2K}$
	Second Order Coefficient C_2	0.015 $\frac{W}{m^2K^2}$
	Absorption Area	6.6 m^2

requiring information about setpoint and actual indoor air temperature. 68 connections are derived by the reasoner.

Figure 49 shows an excerpt of the inferred connections focusing on the the heating rate provided to the building. The heat flow output of each radiator from the HVAC FMU is connected to the same input variable of the building FMU requiring an aggregation during the variable exchange. This issue, arising from the many-to-one cardinality, is further discussed in section 5.1.3. Regarding the required setpoint and current air temperature for each radiator, the situation is reversed. In a one-to-many connection, the setpoint as well as the indoor air temperature are provided to each radiator within the HVAC FMU.

The resulting indoor air temperatures with the mentioned heating system for the same two winter weeks as above are shown in Figure 48. While the temperature setpoint can be reached during the course of the day, the inertia of the system is too high to provide enough heat to follow the increase of the setpoint at the beginning of the day. This results in a significant gap between desired and actual temperature in the building during early working hours.

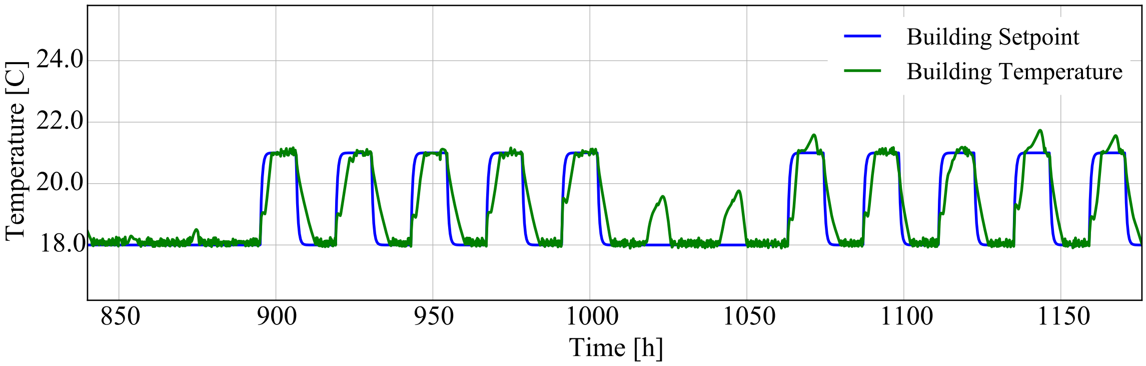


Figure 48. Air temperature and setpoint in simulation run 1.2 for two winter weeks in the building.

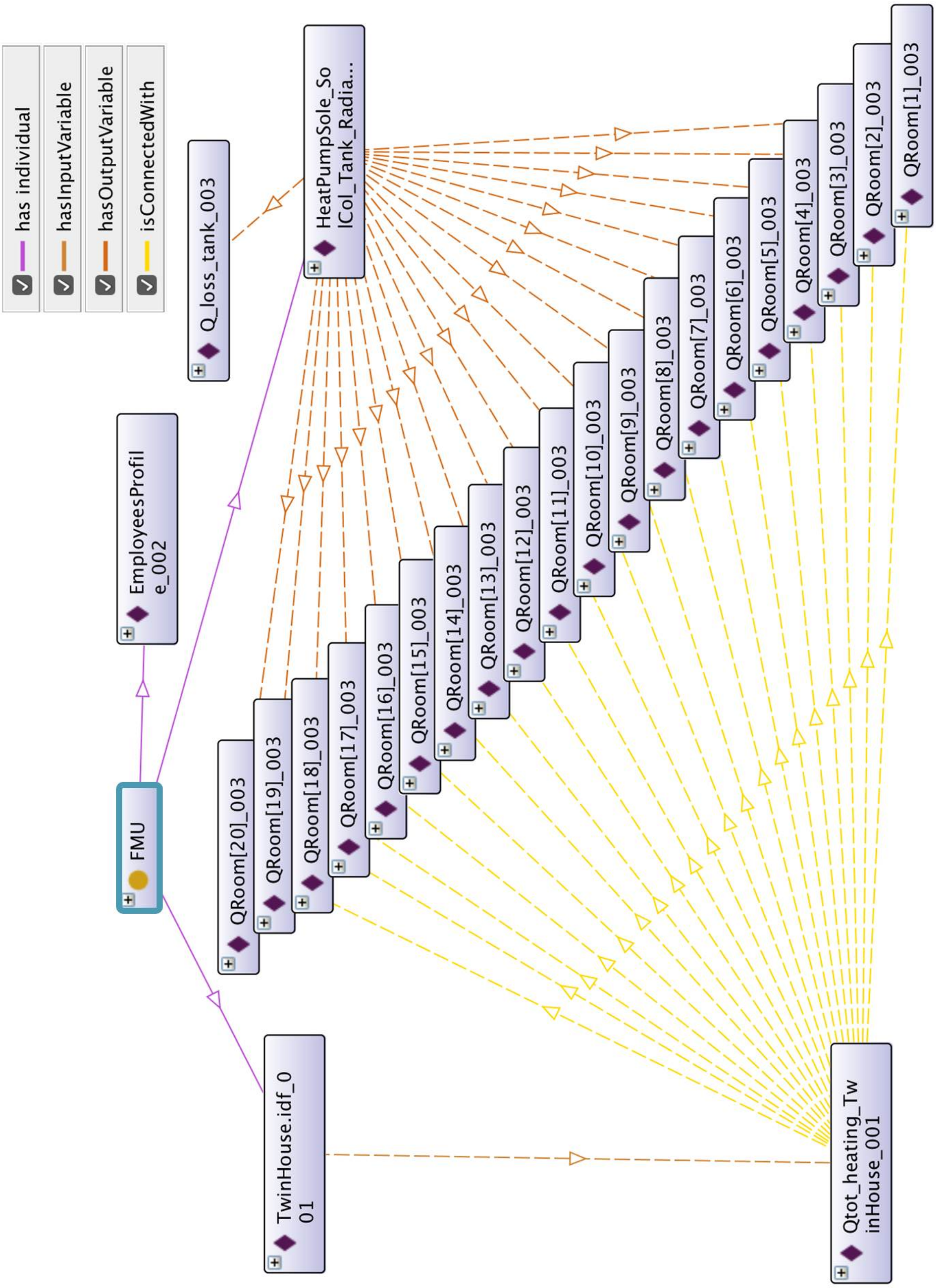


Figure 49. An excerpt of the simulation ontology in run 1.2 after executing the reasoner. Connections between the heating rates provided by each radiator within the HVAC FMU and the heating rate input to the building are shown. The in FMUont defined FMU class is framed in blue. (yellow circles indicate classes; purple diamonds indicate individuals)

4.1.2 Multi-Zone Simulation

To increase the simulation LOD, the building envelope is in the following modeled in a multi-zone representation. The change is realized within SketchUp. The implemented plugin allows for immediate regeneration of the building envelope FMU with now six thermal zones. Figures 50 to 52 show the resulting subdivision of the building envelope model. The Basement is represented in a single, unheated zone. The first floor is subdivided into Entrance, Bathroom, Offices and Lounge, while the second floor solely consists of the zone Meeting.

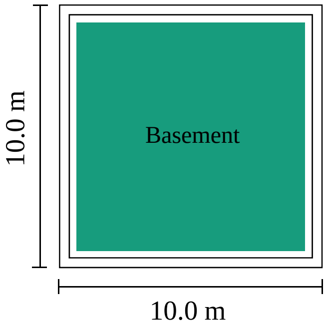


Figure 50. Zoning of the basement of the Twin House beginning with simulation run 1.3.

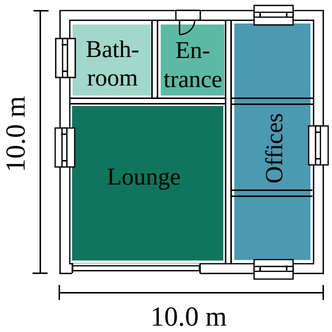


Figure 51. Zoning of the first floor of the Twin House beginning with simulation run 1.3.

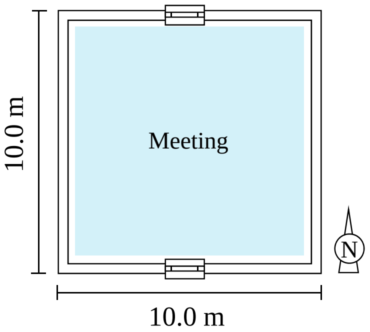


Figure 52. Zoning of the second floor of the Twin House beginning with simulation run 1.3.

Run 1.3 - Ground Heat Pump and Solar Collector with Multi-Zone Model

For the following simulation, the HVAC module remains identical to simulation run 1.2. The FMU representing the people domain is now applied to each occupied zone individually. Therefore, this FMU is instantiated multiple times. Each instance can be parameterized individually and therefore consider e.g. the distribution of people among the zones or different setpoint temperatures. In this case, the addition of identification numbers within the simulation ontology becomes crucial since each occupancy FMU instance features the same Abox. The provided identification number during import into the simulation ontology allows for unique differentiation between these instances. The excerpt of the resulting simulation ontology shown Figure 54 illustrates this aspect.
In contrary to FMUs from other domains, the Abox of the building envelope FMU contains, besides the variable descriptions, information about the thermal zones in the model. Figure 53 illustrates the content of the Abox without expanding the variable descriptions. Corresponding to the multi-zone formulation of the model, each zone is represented as an individual within the FMU specific Abox description.

For detection of variable connections in a multi-zone simulation, it is necessary to include project specific information in the form of a digital building model. In this case, a representation in the SimModel format was realized. Its specification was specifically aimed at

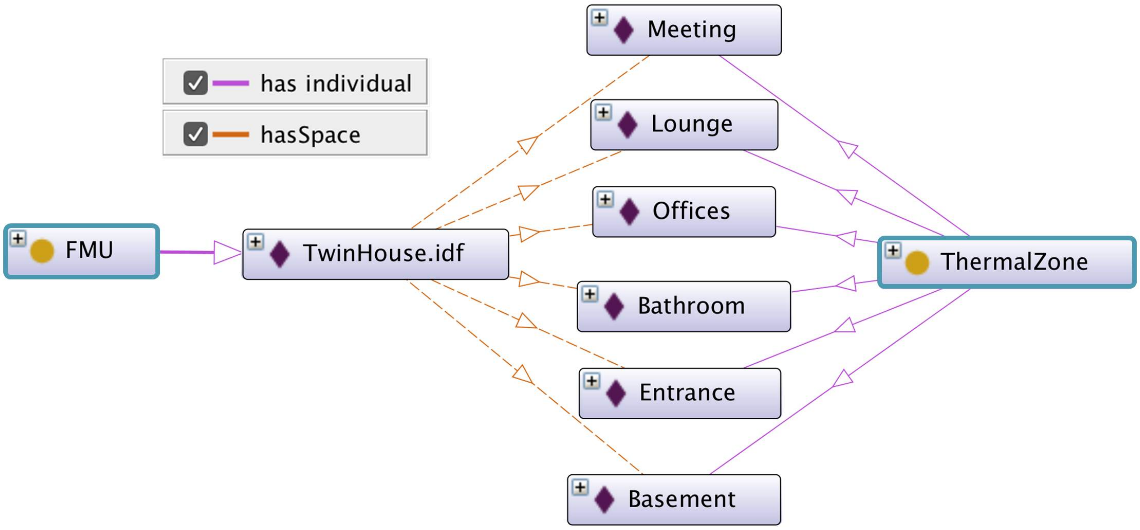


Figure 53. Thermal zone individuals in the Abox of the Multi-Zone FMU. In addition to variable descriptions, the Abox of the building envelope FMU contains information about the thermal zones in the model. The classes defined in FMUont are framed in blue. (yellow circles indicate classes; purple diamonds indicate individuals)

accommodating information of buildings relevant for building energy simulation. As such, the format supports the concept of thermal zones.

After import of each Abox into the simulation ontology, connections to the building data model are realized. These connections are required for variables as well as the thermal zone instances contained in the Abox of the building envelope module. Subsequently, the Dijkstra algorithm for derivation of the associated thermal zone of a variable can be executed. Figure 54 shows the resulting data structure. Among the FMU instances, the building envelope FMU features the property *hasSpace*, which relates the thermal zone instance *Lounge* to this module. In order to associate this simulation specific individual to a space in the building, it is connected via the *hasAffiliationTo* property to a zone within the SimModel representation. In this case, the individual *CommonRoom* is the corresponding counterpart in the building data model. Similarly, simulation variables are connected to their representing objects. As an example, the association of a variable representing a heat flow rate from a radiator within the HVAC module, namely *QRoom[1]*, is shown. This variable is connected to an instance of a radiator object within the building data model called *LoungeRadiator1*. Through the internal taxonomy of SimModel this instance is ultimately connected to the spatial zone instance *CommonRoom*. The Dijkstra algorithm is able to detect this path and return the corresponding thermal zone within the simulation, in this case the thermal zone *Lounge*.

The process leads to the association of each variable with a thermal zone. This enables the reasoner to integrate this property in the inference process and consider it according to the rule formulated in Listing 5. Figure 55 shows the derived connections for the heat flow rates originating from the radiators in the HVAC module. Opposed to the connections in the single-zone simulation shown in Figure 49, multiple zones within the building envelope FMU now receive heat flows. The correct allocation of heat flow rates to these zones is enabled through the above described relation of variables to their corresponding objects in SimModel and ultimately their corresponding thermal zone. In total, among 10,161 possible connections, 76 could be inferred for the contributing seven FMUs.

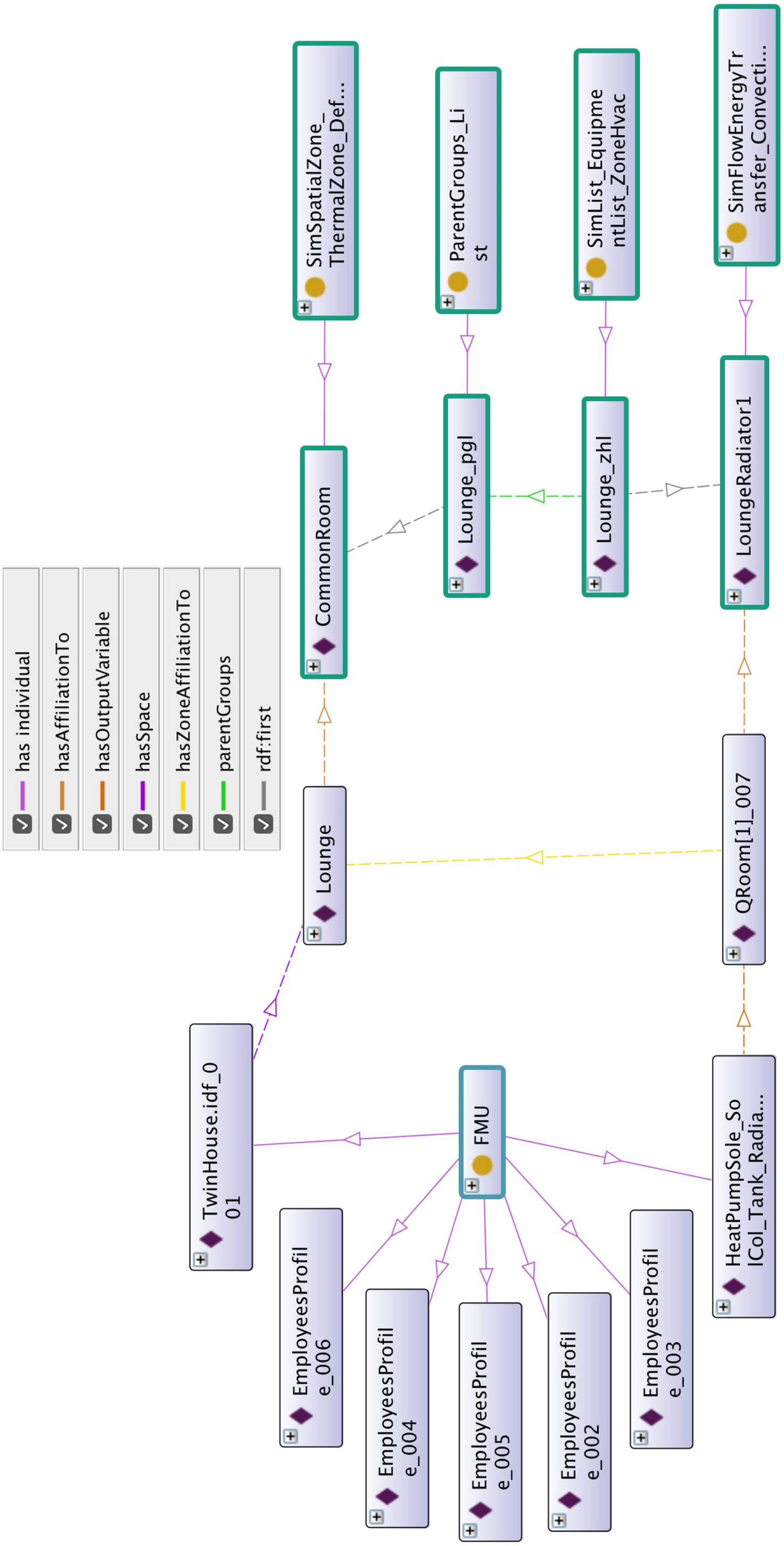


Figure 54. An excerpt of the simulation ontology in run 1.3 after the association of variables with objects from BIM and the execution of the Dijkstra algorithm. The instances and classes from SimModel are framed in green, from FMUont in blue. (yellow circles indicate classes; purple diamonds indicate individuals)

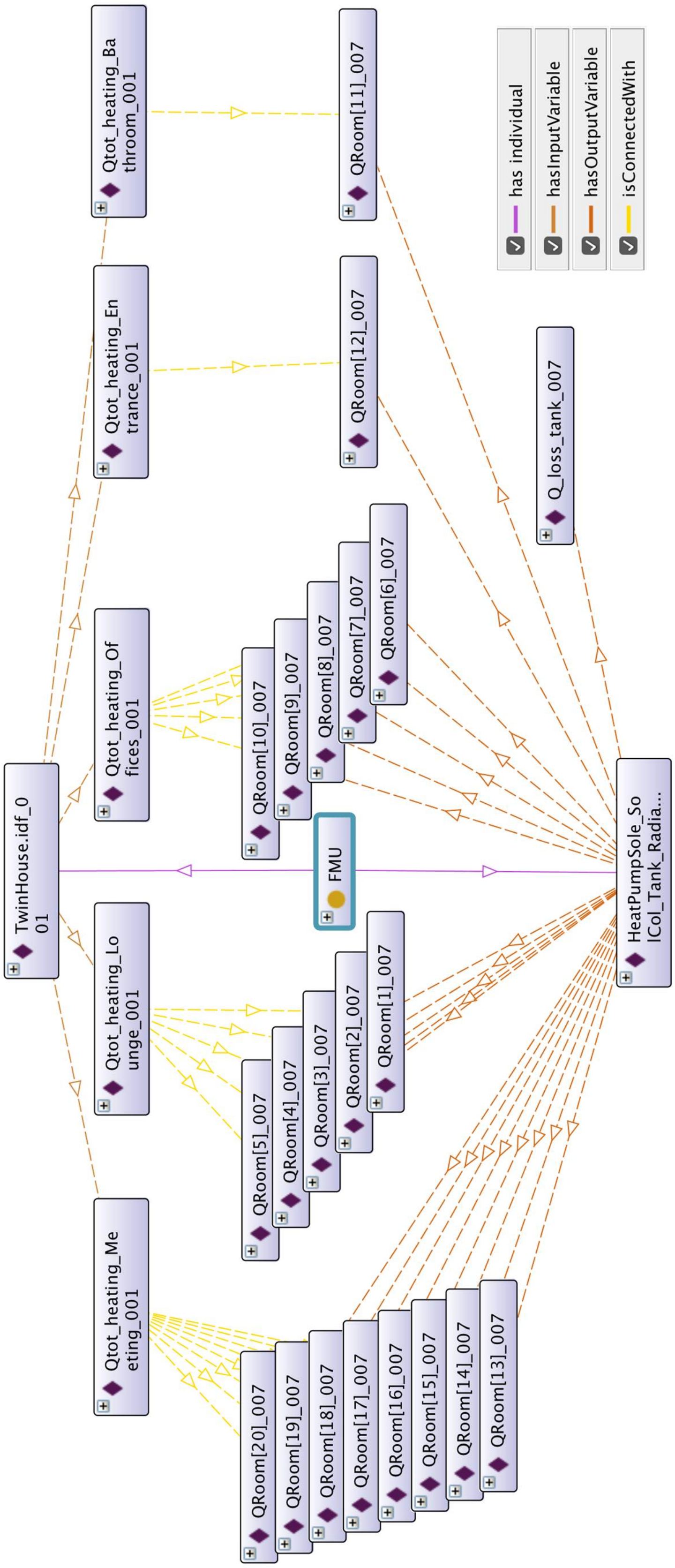


Figure 55. An excerpt of the simulation ontology in run 1.3 after executing the reasoner. Connections between the heating rates provided by each radiator within the HVAC FMU and the heating rate input to the building are shown. The connections reflect the zone association of each radiator.

Figure 56 depicts the resulting temperatures in three thermal zones compared to their setpoint for the same two winter weeks as in Figure 48 for simulation run 1.2. The temperature in the unheated basement remains at a moderate level below the temperature of the heated zones. The temperature of the Lounge and Offices can be held close to the setpoint at all times of the day. In comparison to run 1.2, the offset in the morning hours can be decreased significantly. This is due to the consideration of the basement as an unheated space. While in the single-zone model, the heating capacity of the HVAC system is distributed equally in the entire building, the multi-zone model accounts for different heating rates in each zone. Hence, the isolation of the basement from the overall heating rate leads to a higher available heating power in the remaining zones. Figure 57 illustrates this effect based on the degree hours below the temperature setpoint. In comparison to run 1.2, the situation could be improved significantly in the Lounge and Office zones.

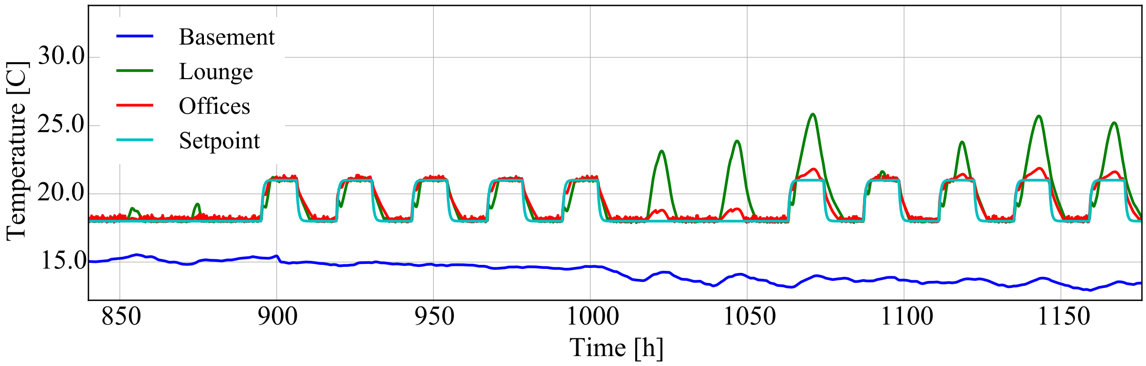


Figure 56. Air temperatures and setpoint in simulation run 1.3 for two winter weeks in the thermal zones Basement, Lounge and Offices.

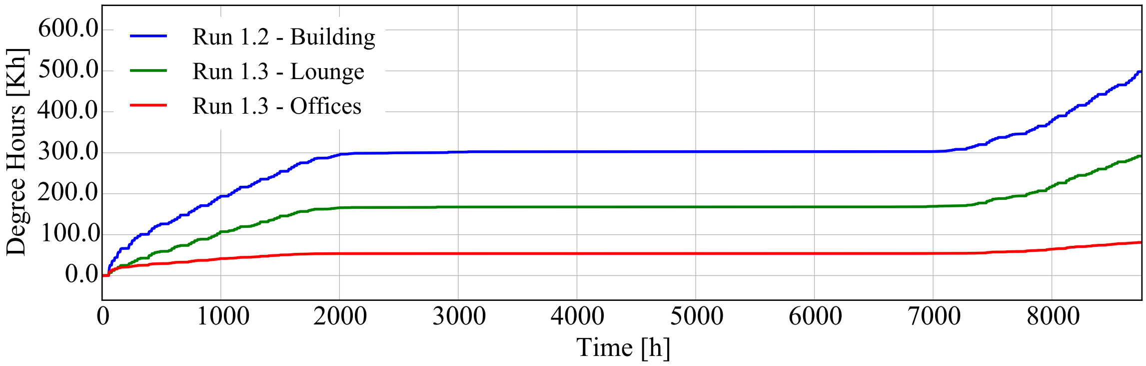


Figure 57. Comparison of simulation 1.2 and 1.3 regarding degree hours below the temperature setpoint.

Run 1.4 - Ground Heat Pump with Multi-Zone Model

In order to investigate the possibility of a heating system solely relying on a heat pump, the HVAC FMU is in the following exchanged with a generic module incorporating the above described system without the supporting solar collector. The parameter specifications remain equal. The change results in 10,029 possible connections among which 74 are realized to perform the co-simulation.

The simulation results in Figure 58 show the temperatures of the heated zones. The setpoint can be approached with good agreement at equal quality as in simulation 1.3. Hence, the heat pump can serve as a reliable system to heat the building.
While the temperature in the zones can still be maintained close to the setpoint, the removal of the solar collector requires the heat pump to increase its operation time and therefore its electricity consumption. Figure 59 illustrates the compressor power of the heat pump during a winter day with high solar radiation. It shows that especially during the day, the solar collector reduces the number of heat pump operation cycles and therefore the electricity consumption through the compressor. Figure 60 compares the consumed energy of the compressor over the course of a year. In total, the consumption grows by 6% of the original compressor energy used in run 1.3.

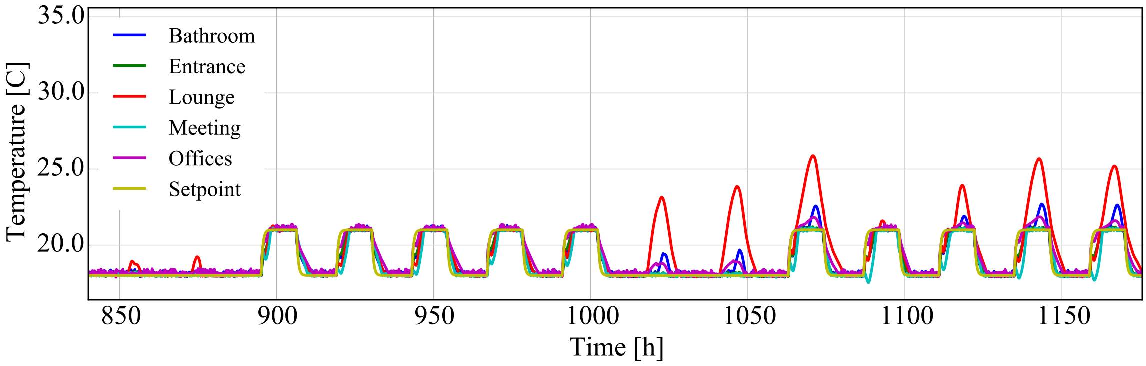


Figure 58. Temperatures of heated zones for two winter weeks in simulation 1.4 compared to the temperature setpoint.

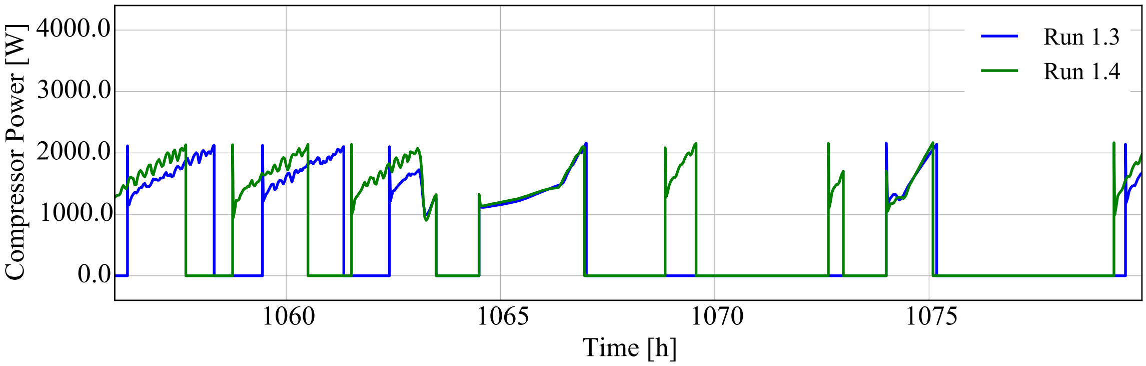


Figure 59. Comparison of simulation 1.3 and 1.4 regarding the consumed power of the heat pump during a winter day with high solar radiation.

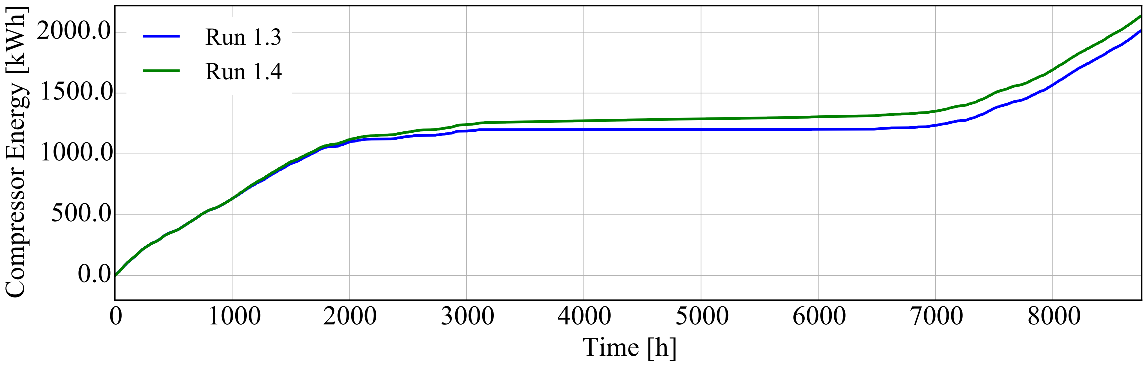


Figure 60. Comparison of simulation 1.3 and 1.4 regarding the consumed electric energy over the course of a year.

Run 1.5 - Calibrated Ground Heat Pump with Multi-Zone Model

At this stage, the models representing the HVAC domain are based on generic library models which are parameterized corresponding to product data specifications. In the following, the HVAC module will be exchanged for a product-specific simulation model calibrated to the performance of a commercially available heat pump as well as a thermal storage unit. The validation and calibration of the model is described in appendix B. The new FMU does not show differences in the exchanged input and output variables, however, parameters, such as listed in Table 3, can not be set any more, since the model is fixed to the characteristics of a specific heat pump as well as thermal storage. Besides a higher quality regarding the prediction of performance, this change also overcomes simplifications made regarding control.

As mentioned above, the generic model implies a 5 °C temperature rise in the condenser. The calibrated model considers the internal control of the compressor and adapts its power to the boundary conditions as implied by the control algorithm of the product. This effect is depicted in Figure 61. While in simulation run 1.4 a constant difference of 5 °C is realized between condenser inlet and outlet, this difference is dynamically adapted in run 1.5 depending on the product-specific behavior implied by the manufacturer. Hence, the module change minimizes assumptions about real-world behavior and therefore increases the quality of the simulation.

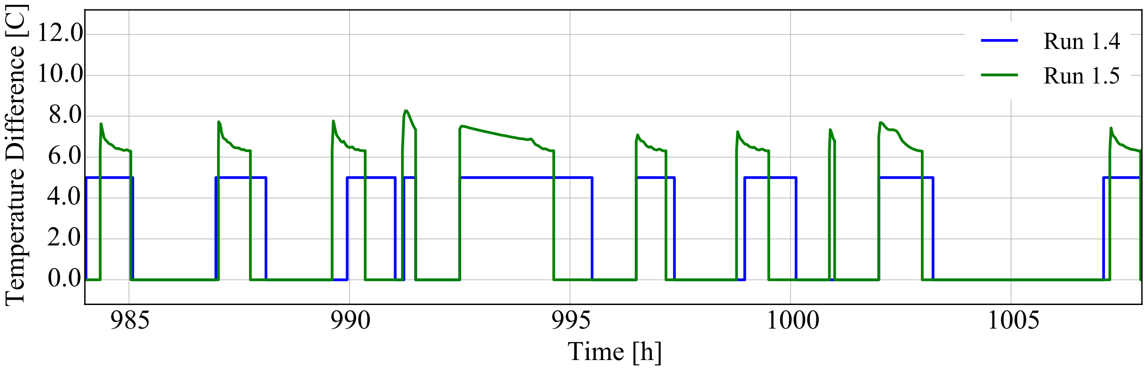


Figure 61. Comparison of simulation 1.4 and 1.5 regarding the temperature difference in the condenser of the heat pump.

Run 1.6 - Calibrated Ground Heat Pump with Multi-Zone Model at reduced Tank Design Temperature

In the current simulation setting, the tank design temperature is aimed at meeting the nominal temperature of the radiators. The following simulation investigates a modification of this temperature in order to improve the performance of the heat pump by means of its time-averaged COP. The COP of a heat pump is the ratio between generated heat and required power, as formulated in the following:

$$COP = \frac{Q_H}{P_{el}} \tag{4.1}$$

When assuming a maximum efficiency of the heat pump, i.e. the Carnot efficiency, the equation can be formulated as

$$COP = \frac{T_H}{T_H - T_C} \tag{4.2}$$

with T_H representing the temperature of the hot and T_C the cold reservoir. Hence, a reduction of the temperature at the hot side of the thermodynamic cycle leads to an improvement of the COP. The simulation therefore tests a reduction of the tank design temperature by 10 °C to 45 °C. Besides changes of the heat pump performance, the capability of the radiators to maintain enough heating to the building at these conditions is assessed. Opposed to the previous simulation runs, this does not require an exchange of a simulation module. Instead, a parameter within the HVAC FMU prescribing the tank design temperature is changed accordingly.

Figure 62 shows the resulting indoor air temperatures in the heated zones for two winter weeks.

Similarly to run 1.5, each room temperature follows the setpoint with acceptable accuracy demonstrating that the radiators are still able to provide enough heating power to the building. The effect of the decreased tank design temperature on the heat pump is depicted in Figure 63 showing the COP of the heat pump over the course of a winter day. It demonstrates that the COP can generally be increased through the reduced tank design temperature. Over the course of the year, the time-averaged COP grows from 4.92 to 5.51. The efficiency gain results in overall electricity savings of 15.2% compared to simulation run 1.5. Figure 64 shows the corresponding use of electric energy in both simulations with run 1.6 being on a lower level than run 1.5 with the elevated tank design temperature.

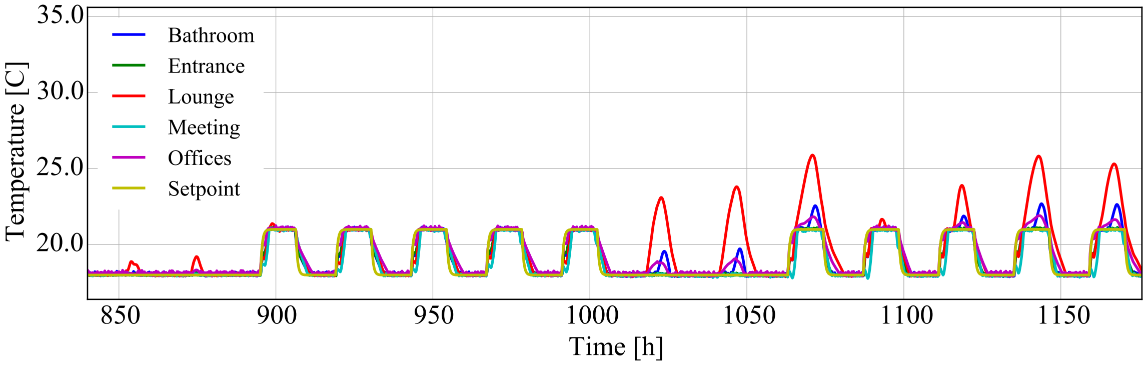


Figure 62. Air temperatures of heated zones for two winter weeks in simulation 1.6 compared to the temperature setpoint.

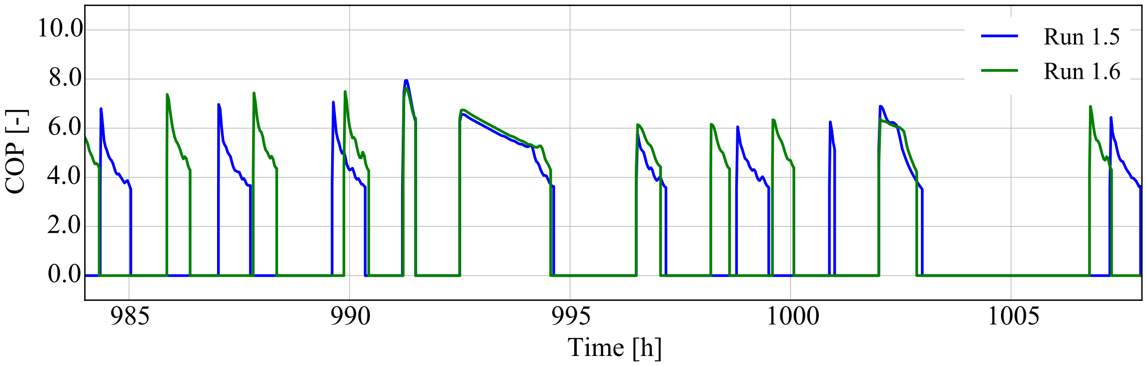


Figure 63. Comparison of the COP of the heat pump in simulation run 1.5 and 1.6.

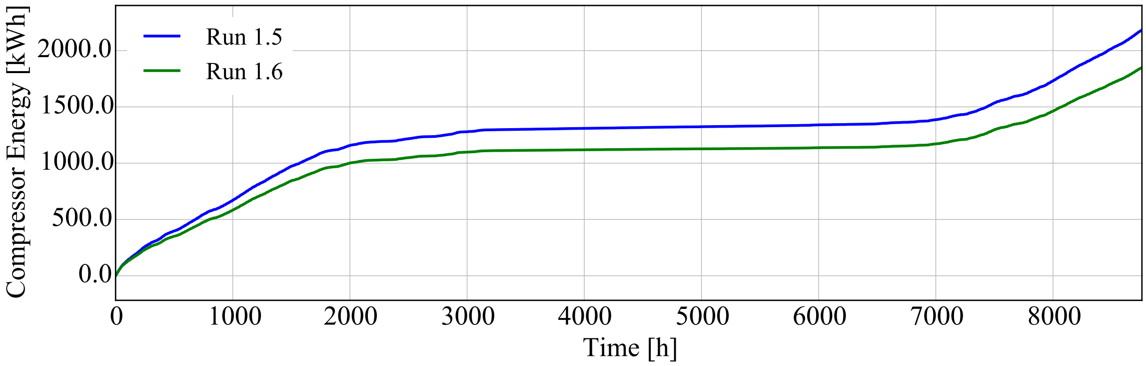


Figure 64. Comparison of electric energy used in simulation run 1.5 and 1.6.

Run 1.7 - Calibrated Ground Heat Pump with Multi-Zone Model and Occupancy Behavior

The next simulation step is aimed at integrating the behavior of people inside the building. Therefore, the Office zone is subdivided into three individual spaces, each designed to become a working space for a specific employee. Figures 65 to 67 show the resulting distribution of zones for the multi-zone model. Since the structure of the model is changed through the refinement of zones, the building envelope FMU needs to be regenerated via the SketchUp plugin.

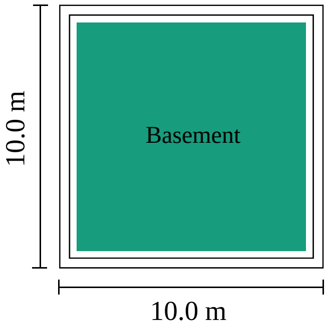


Figure 65. Zoning of the basement of the Twin House beginning with simulation run 1.7.

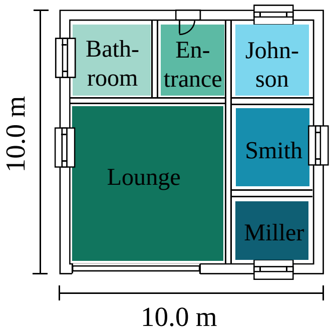


Figure 66. Zoning of the first floor of the Twin House beginning with simulation run 1.7.

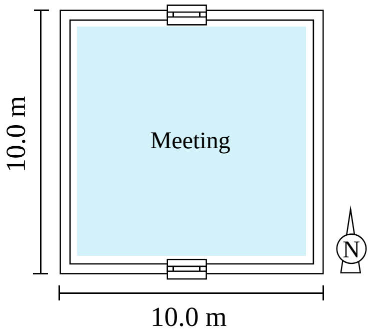


Figure 67. Zoning of the second floor of the Twin House beginning with simulation run 1.7.

Each of the office zones, receives an individual occupancy profile that characterizes the corresponding employee. Furthermore, behavior models are integrated that represent the employees' interactions with the building. Therefore, each occupant is associated with its own behavior model. In this case, a model triggering window opening [44] and a model triggering blind activation [131] are included as separate modules. Either model is based on experimental studies for occupant behavior in office buildings and calculates a probability for the corresponding action depending on boundary conditions which were determined to highly correlate with this action. In the case of the window model, this independent variable is the indoor air temperature of the zone, while for the blind activation, the current indoor illuminance is decisive. The computation of these probabilities furthermore differentiates between occupancy states, i.e. arrival, on-going presence and leaving. In order to allow for integration of these stochastic models into the simulation, the resulting action must unambiguously be indicated. This is achieved through comparing the computed probability with a random number between zero and one. If the probability for an action exceeds this number, the action is reported to the simulation with a signal variable of value one, as recommended in [67].

In this run, the simulation consists of 17 FMUs. Figure 68 shows the resulting simulation ontology after importing each Abox and connecting the variables to the populated SimModel. Each of the behavioral models is initiated four times. In addition to the offices, also the occupancy behavior in the Lounge is considered. Similar to simulation run 1.3, the necessity of unique identification numbers becomes apparent. Despite the multiple instantiation of a single FMU, differentiation of FMU and variable names must still be ensured.

The output variables of two of the blind models are expanded. Both variables are associated with an occupant in the digital representation of the building. The internal associations in SimModel allow for deriving the path to spaces in this data model and ultimately their corresponding thermal zones in the simulation model. It can be seen that either variable is connected to a different zone, namely the offices of Miller and Johnson. The remaining variables are treated correspondingly and ultimately the reasoner is able to detect the connections between the simulation modules. For the 17 contributing FMUs 21,259 connections are possible. 118 connections are detected by the reasoner.

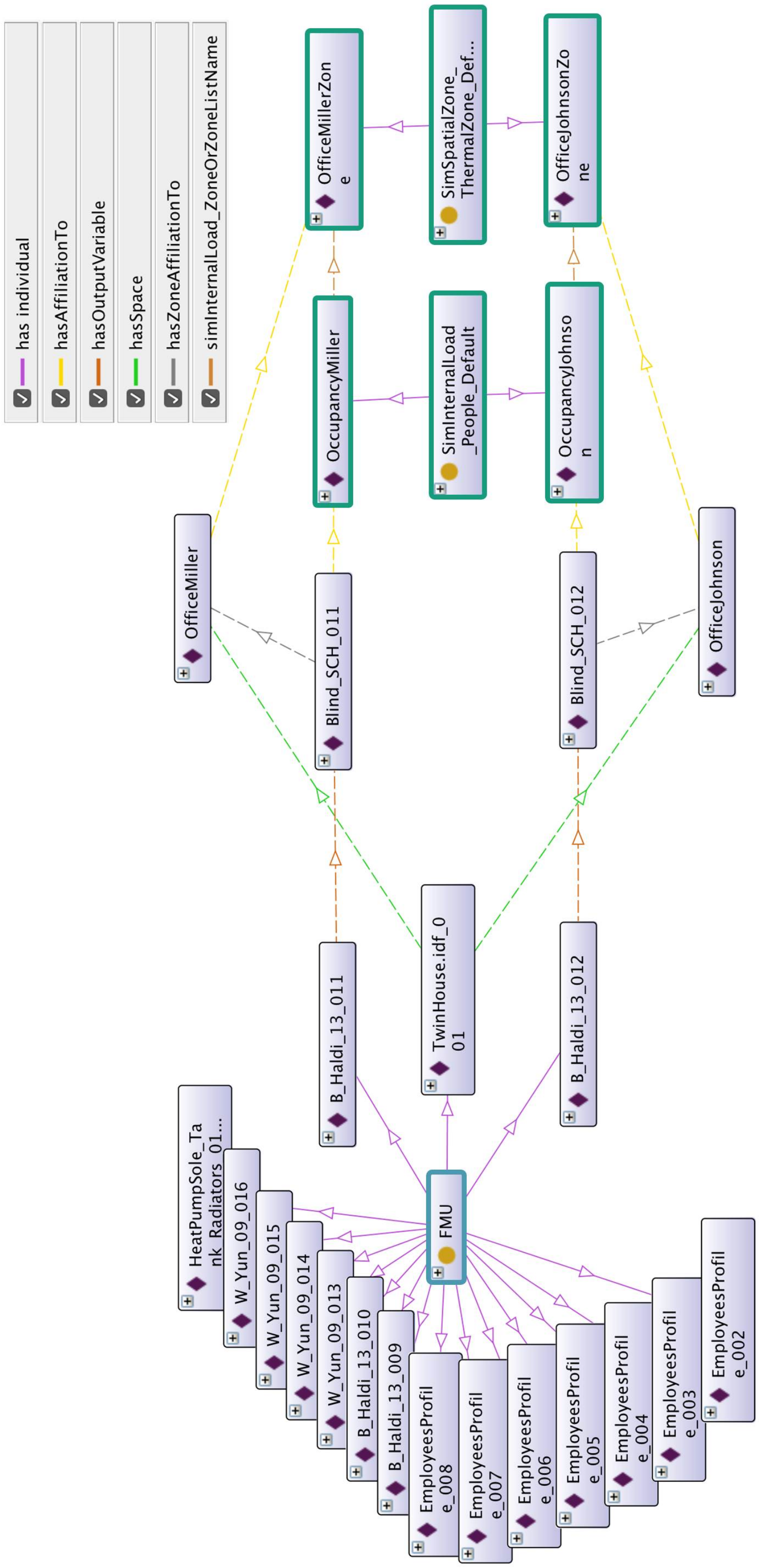


Figure 68. An excerpt of the simulation ontology in run 1.7 after connecting variables to objects in a building data model. Two signal variables from the behavior FMUs are expanded. Through the internal association in SimModel, they can be associated to thermal zones. The instances and classes from SimModel are framed in green, from FMUont in blue. (yellow circles indicate classes; purple diamonds indicate individuals)

The multiple instantiation of an FMU allows for applying the same module to different situations within a single simulation. This results in different boundary conditions for the individual instances and therefore in different results. In the present case, this allows for assessing the behavior of each occupant in the considered zones individually. Figure 69 shows the resulting behavior regarding blind activation in the offices of Miller and Johnson for one day. The high illuminance in the morning induces actions of both. The closed blinds immediately result in a decreased illuminance in both rooms. While the north-facing window in the office of Johnson provides a lower amount of daylight to the room, the illuminance in the south-facing room of Miller is still, despite the closed blinds, highly influenced by the course of the sun. During the day, decreasing illuminance levels lead to the opening of the blinds. Due to the lower illuminance level in the north-facing office, this action is earlier executed by Johnson than Miller.

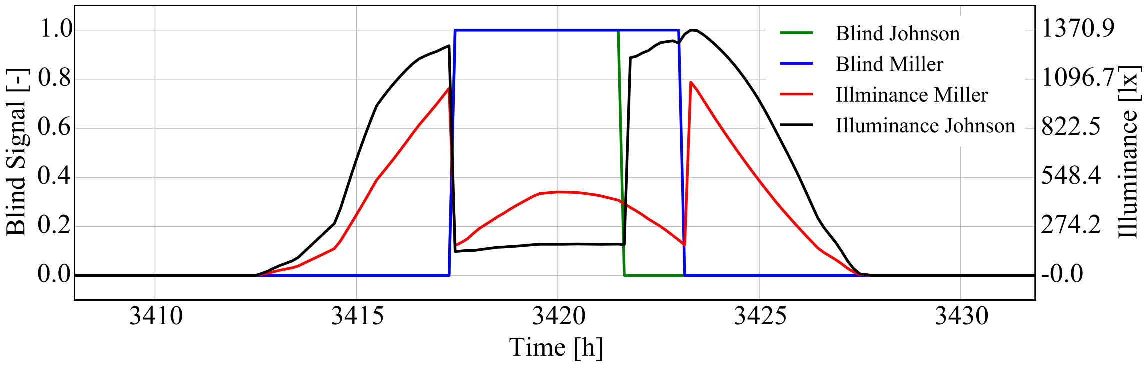


Figure 69. Blind opening behavior in two offices in run 1.7 and the resulting effect on indoor illuminance.

The behavior regarding window opening is exemplified in Figure 70 for a single day. Opposed to the situation above, the window states in the rooms differ at the beginning of the day. While the windows in the office of Miller are open during the night, the windows in Johnson's office are closed. This results in reversed actions at the beginning of the day with Miller closing the window and Johnson opening the window for a short period. During the day, several repetitions of window opening periods occur. The higher temperature level in the south-facing office of Miller leads to more and longer opening periods than in the office of Johnson. At the end of the day, the high temperature level in Miller's office causes its occupant to leave the window open before departure. Opposed to this, the window is closed in the office of Johnson.

While having secondary effects during the winter, the indroduced behavior models mainly affect the indoor air temperature in the summer. As an example, Figure 71 shows the temperatures in the two discussed offices compared to the Office zone in simulation run 1.6. Either office, north- and south-facing, shows an improvement regarding high indoor air temperatures. Hence, the peoples' behavior regarding blind activation and window ventilation is able to improve the thermal comfort during the summer.

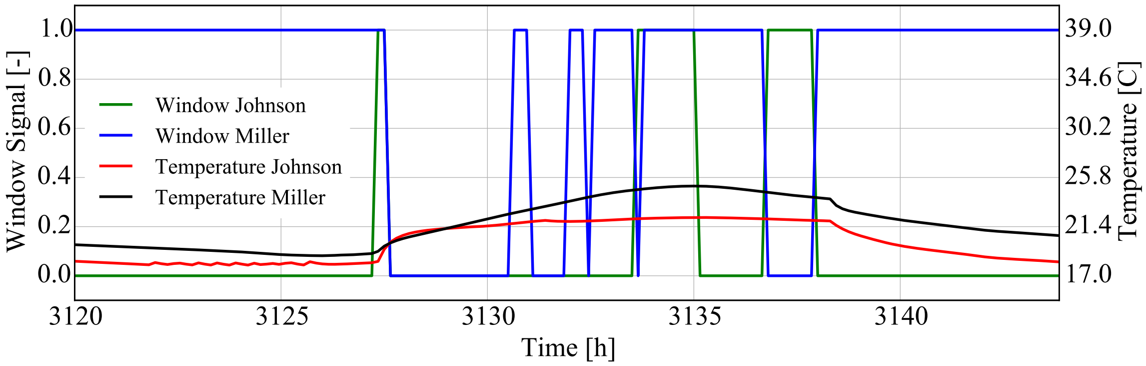


Figure 70. Window opening behavior in two offices in run 1.7 and their indoor air temperature.

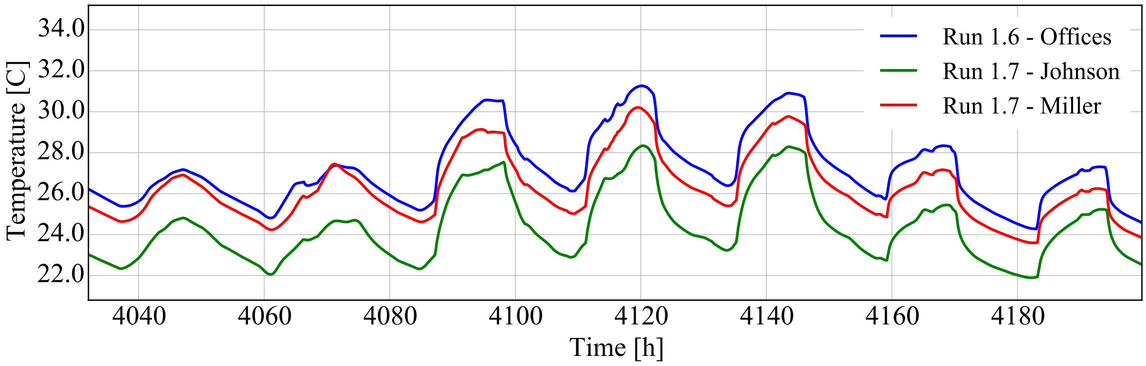


Figure 71. Comparison of indoor air temperatures in run 1.6 and 1.7 for a summer week.

4.1.3 Zonal Airflow Simulation

With each room being modeled as a thermal zone, the multi-zone model can not be increased in its spatial resolution. In order to assess the thermal climate within the single rooms in more detail, the enclosed air volume must be subdivided into several smaller volumes while considering the airflow between these volumes. As elaborated in section 2.1.3, the zonal airflow model is built upon this concept.

In the following section the indoor climate in the Lounge is assessed in more detail using such a zonal airflow model. Figure 72 depicts the location of the room within the building and its particularity. Compared to the remaining rooms, its window area is considerably higher. Especially the double-glazed window front facing south can lead to undesired effects. The following simulations are intended to investigate these effects for a cold winter day. Especially the positioning of radiators within these rooms is targeted. Since the zonal airflow model considerably increases the computation time, the following simulations are executed for a single winter week.

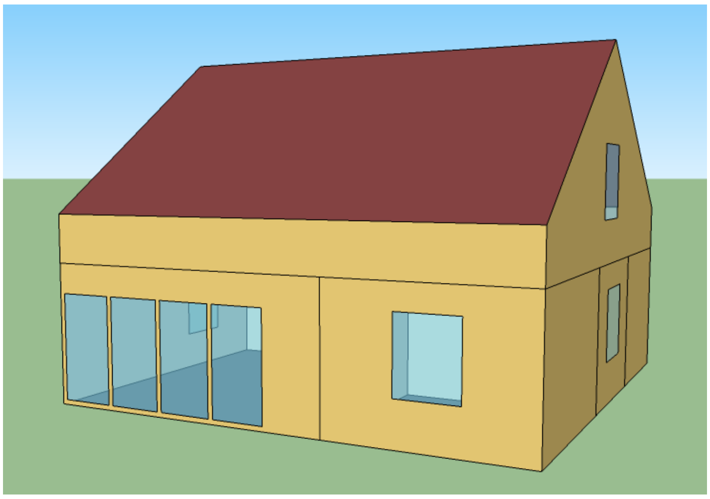


Figure 72. View of the Twin House with the Lounge located in the bottom left corner.

Implementation of the zonal airflow model is realized in Modelica and follows the methodology developed in [91]. The subsequently added Abox contains, similar to the Aboxes in the building envelope FMUs, an instance of class *Space*. In this case, however, the instance is of subclass *Room* instead of *Thermal Zone*. Figure 73 shows an excerpt of the corresponding Abox. In addition to the room represented in the zonal airflow model, also input and output variables as well as the simulated system are shown. As a part of the building, the room model requires the temperatures from neighboring zones as boundary conditions. Similarly, weather data, such as ambient temperature and solar irradiation is necessary. Besides these thermal influences, also occupancy behavior is considered in the room model via the corresponding input signals for blind activation and window opening. This requires the resulting effect, i.e. transmitted solar irradiation and exchanged airflow rate, to be modeled within the room model. Further inputs are internal heat loads from people as well as heat flow rates from radiators, which are distributed to their corresponding locations, as described in the following. Output variables describe the air temperatures at these locations as well as the resulting operative temperature in the vicinity of the window front.

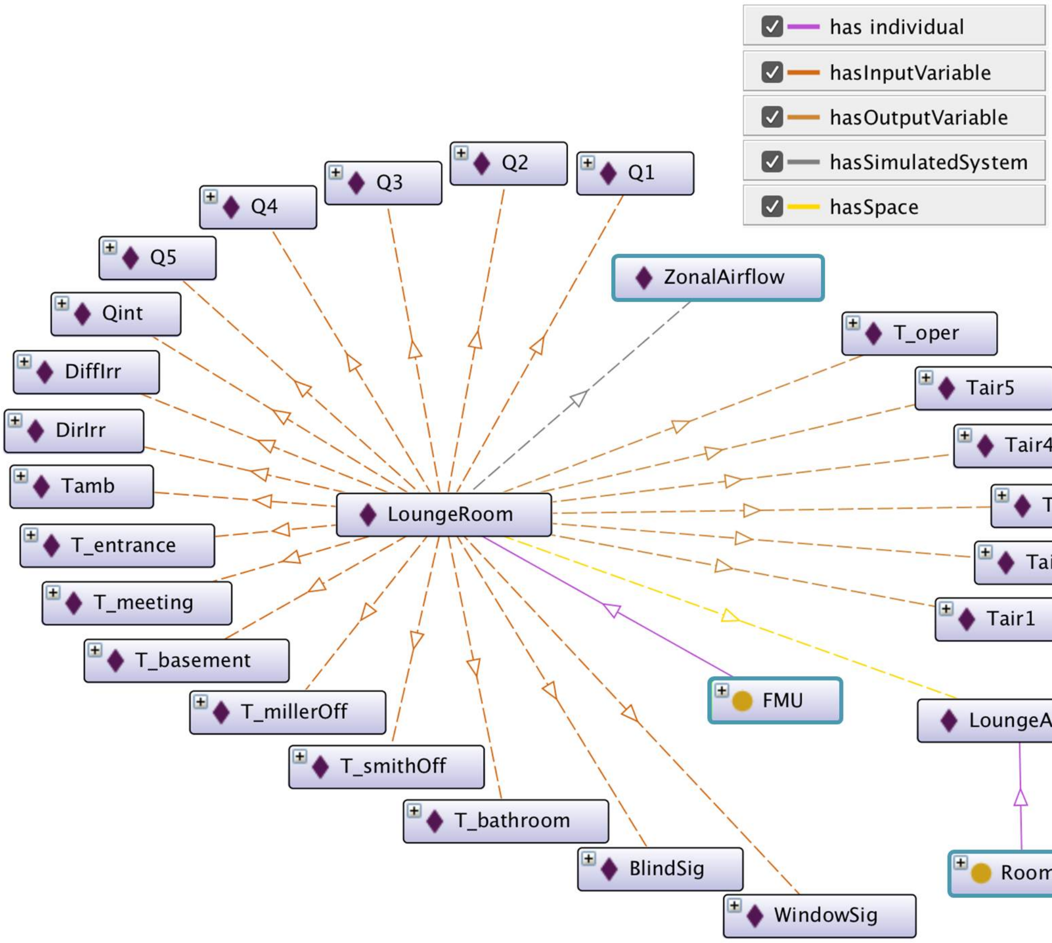


Figure 73. An excerpt of the Abox description in a zonal airflow model representing a single room within a building shows the contained room instance. In addition, input and output variables of the model and the simulated system are depicted. Instances and classes from FMUont are framed in blue. (yellow circles indicate classes; purple diamonds indicate individuals)

Run 1.8 - Calibrated Ground Heat Pump with Multi-Zone and Zonal Airflow Model

In a first setup the radiators are placed as depicted in Figures 74 and 75. One is located at the west-wall, next to a second one beneath the west-facing window. The remaining radiators are placed at the rear sides of the room bordering to internal walls in order to keep the window front free from visual obstacles.

After the integration of the zonal airflow module into the simulation, the Abox information of each FMU is integrated into the simulation ontology. Subsequently, variables are associated with their corresponding object in the digital building model. Through applying the set of rules described in section 3.5, the reasoner detects the correct connections between the FMUs to perform the simulation with the increased LOD. In total, the simulation consists of 18 FMUs with 24,832 possible connections among which 135 are realized.

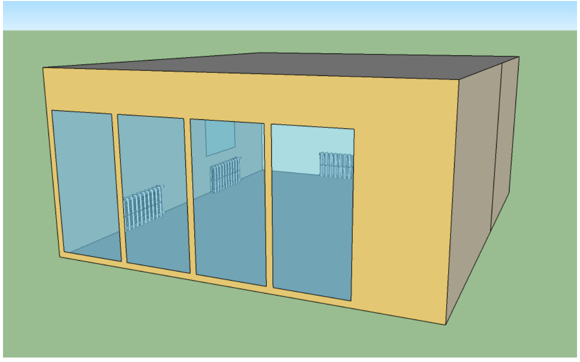


Figure 74. View of radiators in the Lounge for the zonal airflow simulation in run 1.8.

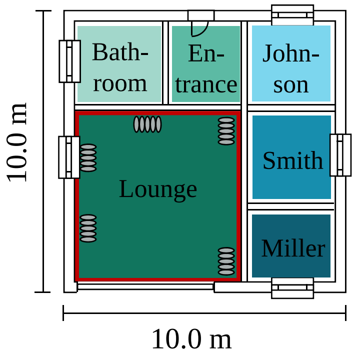


Figure 75. Layout of radiators in the Lounge for the zonal airflow simulation in run 1.8. The spatial boundaries of the Lounge are marked in red.

Figure 76 shows an excerpt of the resulting simulation ontology. The figure depicts the included FMU instances with the building envelope, the zonal airflow and the HVAC module being expanded. The variables illustrate the resulting relations between the thermal zones of the building and the detailed room model. The room module receives five heating rates corresponding to the number of radiators in the room. Each heating rate is guided to the location of the corresponding radiator within the model. As an example, *Q1* is illustrated in the ontology scheme. In order to adjust the thermostats at each radiator individually, the current air temperature at the radiator locations is provided to the HVAC module as an output; in Figure 76 exemplified with *Tair1*. Both variables are associated with the same radiator object in the building data model. Since their in base properties identical counterparts in the HVAC FMU are associated to the same object, connections between these variables can be derived and a room instance, i.e. *LoungeArea*, can also be associated with the corresponding partners. Hence, the variables from the HVAC module associated with the object *LoungeRadiator1* receive, in addition to their association to a thermal zone, the information about the room the radiator is located in. Due to the parallel existence of a multi-zone model for the entire building envelope and a model for a single room within this building, the heat flow originating from a radiator within this room must be connected to both. Therefore, the heat flow from the HVAC module *QRoom[1]* is also connected to the receiver in the multi-zone representation of the building, based on the comparison of identical thermal zones, as previously described in section 4.1.2. In order to enable the thermostat control of the indoor air temperature, the implementation prefers the provided temperature of the module with greater LOD. Hence, instead of the temperature representing the entire thermal zone computed in the multi-zone model, the above-mentioned air temperature provided by the zonal airflow model *Tair1* is connected to *T_zone[1]* from the HVAC module.

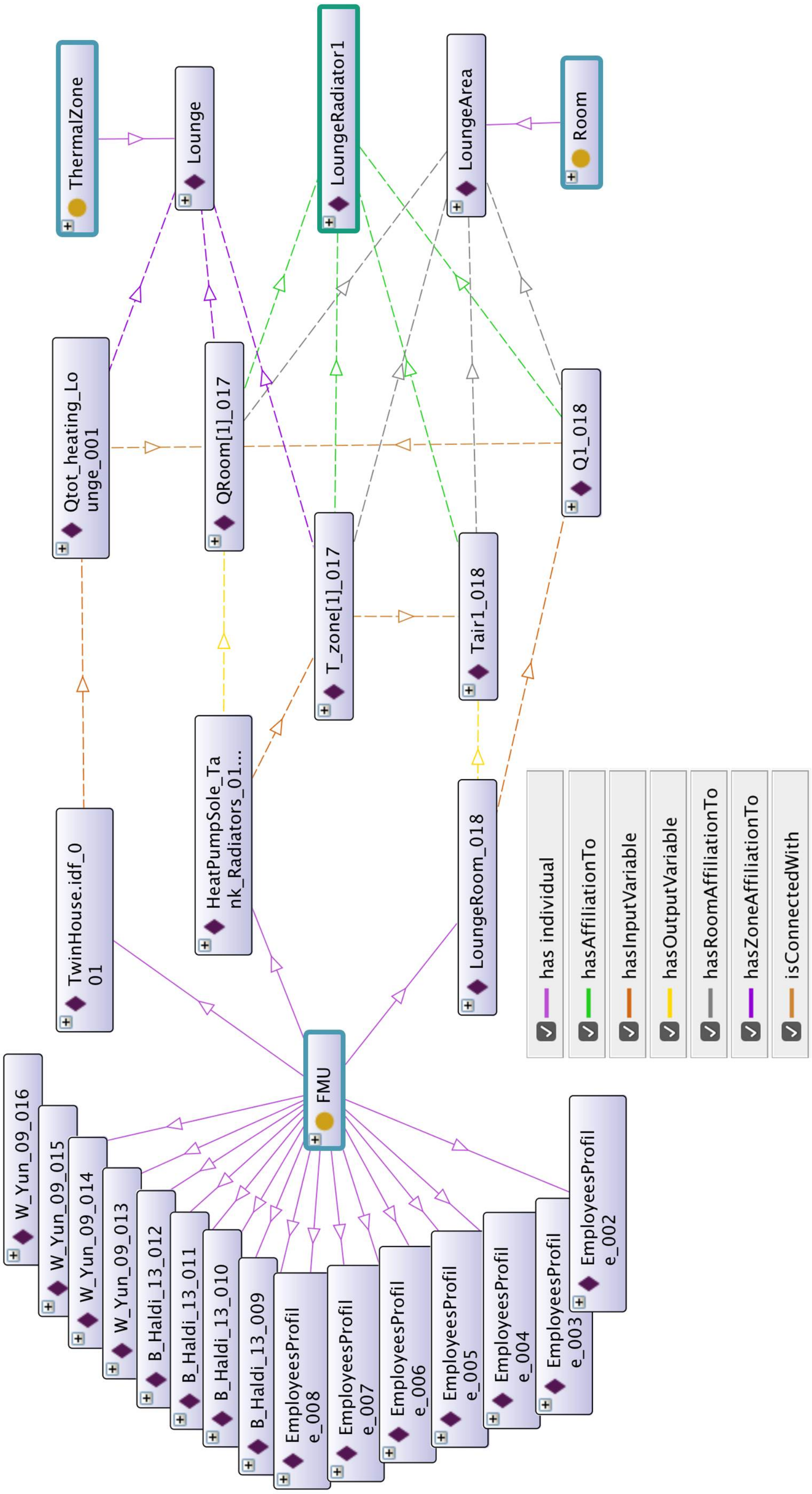


Figure 76. An excerpt of the simulation ontology in run 1.8 after executing the reasoner. Connections and associations to a digital building model are shown for four variables from three different FMUs. The associated instances from SimModel is framed in green, from FMUont in blue. (yellow circles indicate classes; purple diamonds indicate individuals)

The resulting co-simulation is able to provide a detailed thermal computation of an excerpt of the building envelope while considering the remaining domains, i.e. the HVAC system, occupancy etc., in their entirety. Hence, a holistic view is still provided when focusing the LOD of the simulation on single parts of the building. This allows for advanced assessment of thermal comfort in enclosed spaces.

The resulting temperature distribution for the Lounge in the present simulation is shown in Figure 77. The visualization depicts the situation on a winter day at 11.00 AM. It illustrates how the low surface temperatures at the inner side of the window front influence the air temperatures in the room. The cooled air close to the windows drops towards the floor and moves into the room inducing a risk for low thermal comfort. While the radiators are able to provide enough heat locally, they can not prevent this cold air stream from entering the center of the room. Hence, due to the positioning of the radiators, occupants are confronted with an air temperature difference of 1.1 °C.

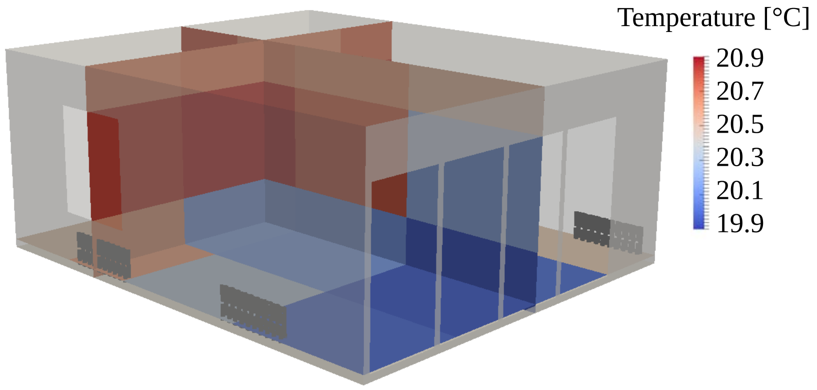


Figure 77. Visualization of the zonal airflow results for the Lounge in run 1.8. The situation for a cold winter day at 11.00 AM is depicted.

Run 1.9 - Calibrated Ground Heat Pump with Multi-Zone and modified Zonal Airflow Model

In order to improve the heterogeneous temperature distribution in the situation above, the following simulation run tests an alternative radiator placement. In this modified situation, two radiators are located close to the window front with the intention to prevent a cold air stream towards the center of the room. Figures 78 and 79 depict the resulting layout.

In order to realize the modification, the zonal airflow model requires an internal change leading to a regeneration of the room FMU. Since no changes regarding the exchanged variables are required, the simulation ontology is identical to run 1.8.

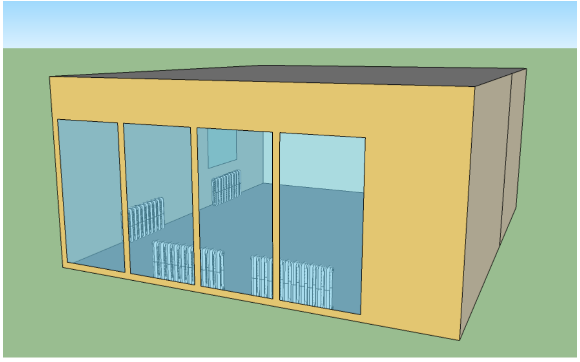


Figure 78. View of radiators in the Lounge for the zonal airflow simulation in run 1.9.

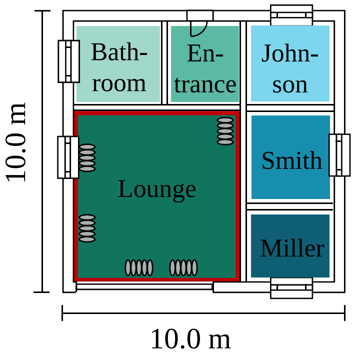


Figure 79. Layout of radiators in the Lounge for the zonal airflow simulation in run 1.9. The spatial boundaries of the Lounge are marked in red.

Figure 80 presents the results of the zonal model with the modified radiator positions for the same date and time as above. It illustrates that the cold airflow originating from the window front can be attenuated through immediate heating of air in the vicinity of the window. Despite the removal of radiators, also the corner and rear parts of the room can be maintained at a sufficient temperature level. The modified positions are therefore able to homogenize the temperature in the room.

Figure 81 further quantifies the effect on thermal comfort by depicting the operative temperature in the vicinity of the window over the course of this winter day. During the day, the operative temperature in run 1.8 is about 1 °C lower than in run 1.9. The latter is able to reach temperatures above 20 °C during this time. While also during the beginning of the day the chosen positions in run 1.9 show better performance, the situation can still be improved in order to reach 20 °C before 11.00 AM, e.g. by an earlier activation of the heating system.

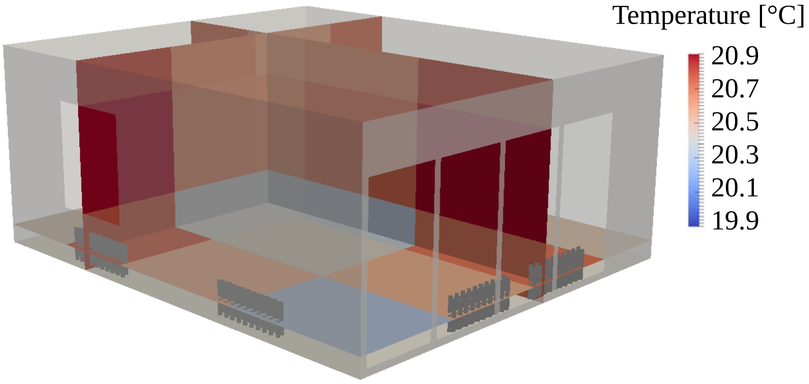


Figure 80. Visualization of the zonal airflow results for the Lounge in run 1.9. The situation for a cold winter day at 11.00 AM is depicted.

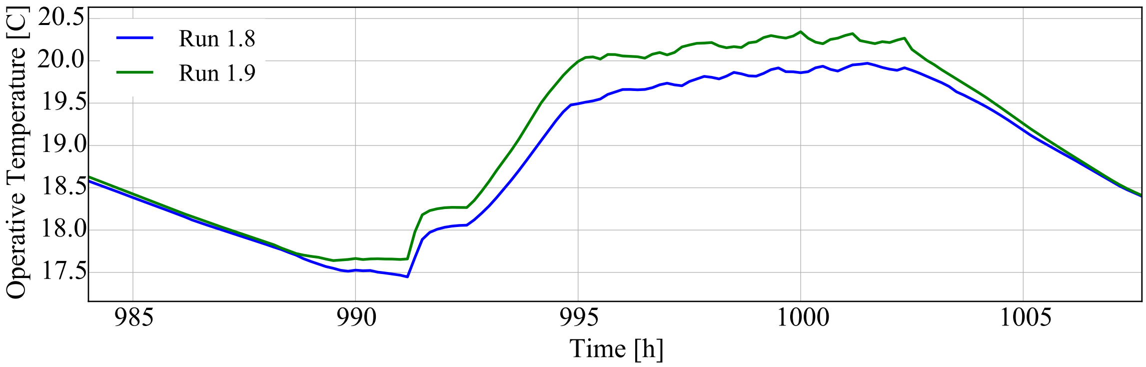


Figure 81. Comparison of operative temperatures in the vicinity of the window front in the Lounge in run 1.8 and run 1.9.

4.2 Example 2 - The FZK-House

The following section presents a second case study applying the developed methodology for a modular building performance simulation. Rather than demonstrating the functionality to increase the LOD during the course of the design process, as shown in section 4.1, this study aims at proofing the compatibility of the process with a digital building data model available as a populated IFC scheme. Opposed to SimModel, the definition of spaces in IFC is independent from the principle of thermal zones as applied in BPS.

The study is based on the IFC representation of the FZK-House from [60]. Figure 82 shows a view from south-west in the IFC visualization tool FZKViewer, available in [61]. Additionally, Figures 83 and 84 depict the floor layouts of the two-story building with the defined IFC spaces. The first floor consists of several separated rooms. The three spaces Hallway, Living Room and Kitchen are connected air volumes without internal walls. In the second floor only a single space, namely Gallery, exists. The air volume of this space is connected to the three mentioned spaces via the Living Room.

In order to integrate this IFC model in the developed methodology for a modular simulation, it is required to translate the building data model to OWL. In [99], Pauwels and Terkaj present a method to realize this step. Based on the in [102] available implementation of the method, the IFC representation of the FZK-House is converted and in the following applied to complement the knowledge base of the modular simulation.

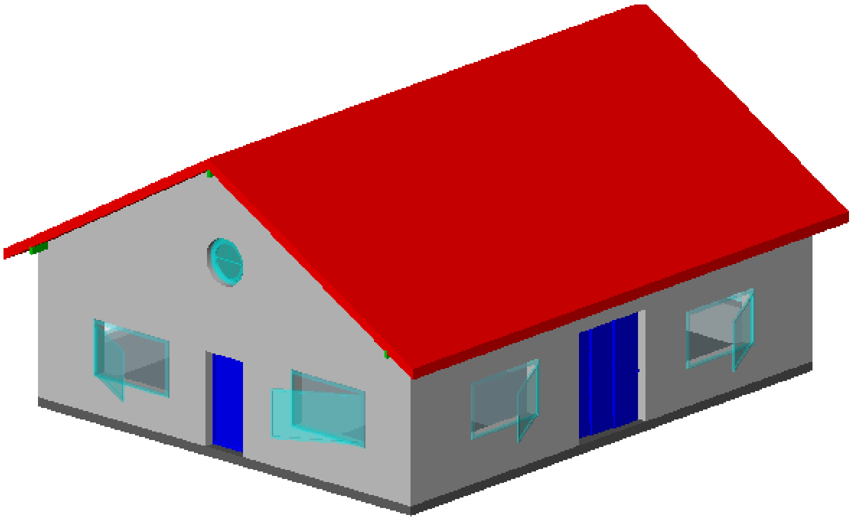


Figure 82. The FZK-House visualized with the FZK viewer. The building consists of two floors. The second floor is connected via an open air space to the living area in the first floor.

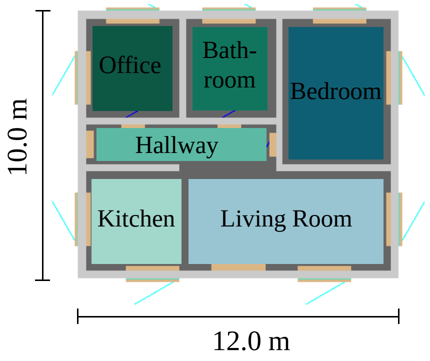


Figure 83. First floor of the FZK-House. IFC spaces are marked in the floor plan.

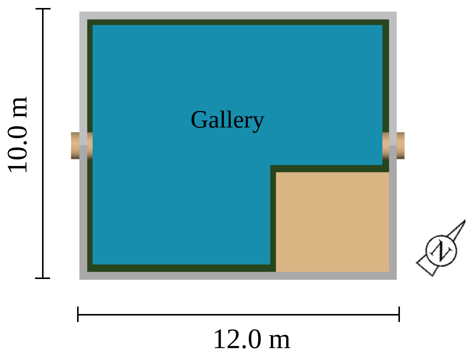


Figure 84. Second floor of the FZK-House. IFC spaces are marked in the floor plan.

Run 2.1 - Calibrated Ground Heat Pump with Multi-Zone Model

In a first simulation run, a multi-zone model of the FZK-House is generated. The distribution of zones is realized according to Figures 85 and 86. The first floor is subdivided into the Office Zone, Bath, Sleeping and the Common Rooms. While the former zones present single rooms that correspond to the definition of IFC spaces as shown in Figure 83, the latter comprises the three connected IFC spaces Hallway, Kitchen and Living Room in the first floor. Additionally, the Gallery in the second floor is included in order to combine the four connected IFC spaces in a single thermal zone.

To represent the HVAC domain and supply the building with heat, the FMU discussed in section 4.1 incorporating a calibrated heat pump and tank model with radiators is re-used. Occupancy is modeled using four FMUs representing different deterministic schedules from the residential sector where each is applied to a specific thermal zone. In total, six modules are used to form the simulation. The weather data for the fantasy building is chosen according to the climate in Munich, Germany.

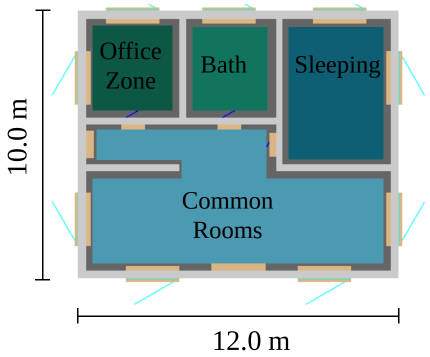


Figure 85. Zoning of the first floor of the FZK-House in simulation run 2.1.

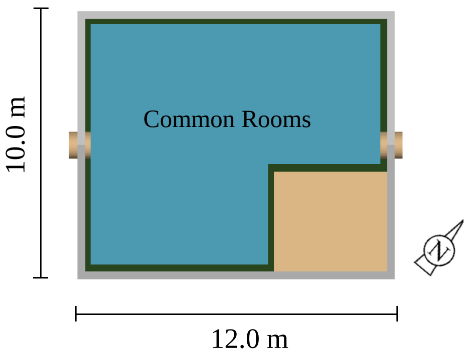


Figure 86. Zoning of the second floor of the FZK-House in simulation run 2.1.

Figure 88 shows an excerpt of the simulation ontology after aggregating each Abox and connecting the thermal zones in the multi-zone model as well as all exchange variables to the IFC data model. It illustrates the spatial associations of two heating rates originating from radiators in the HVAC FMU. Each heating rate is connected to an individual *IFC Space Heater* instance, which is associated with an *IFC Space* through the internal data structure. In this case, the radiators are located in the Kitchen and the Gallery. Since both spaces are associated with the same thermal zone instance from the multi-zone simulation model, the associated zone, the Common Rooms, is identical for both variables.
With the IFC information added to the simulation ontology, the reasoner is able to infer the connections between the FMUs by comparing their base properties, domain context and project specific associations. Among the 6,247 possible connections, 66 are realized.

The simulation results shown in Figure 87 for the air temperatures in the Common Area compared to the setpoint proof the ability of the current design to provide sufficient heating energy to this zone. In the two depicted winter weeks, the HVAC system is able to follow the setpoint with low differences.

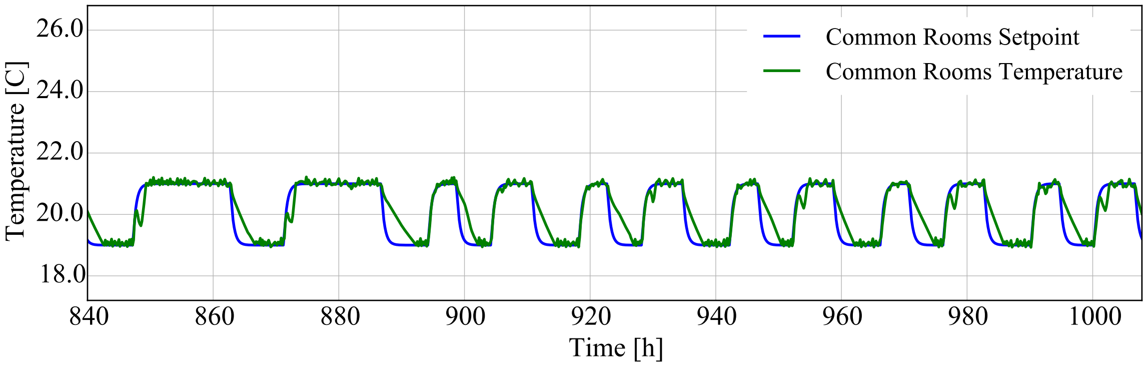


Figure 87. Air temperature in the Common Area for two winter weeks in simulation run 2.1 along with the setpoint.

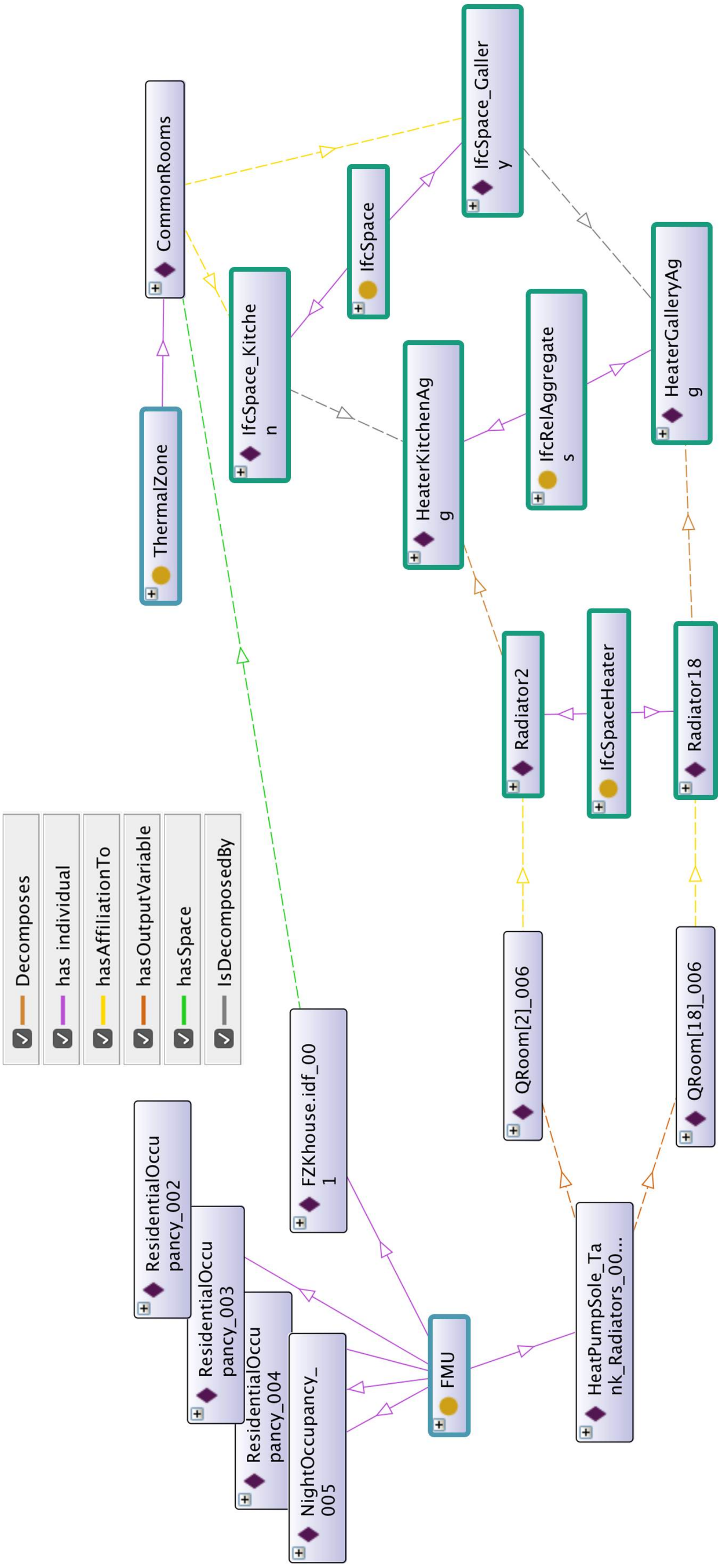


Figure 88. An excerpt of the simulation ontology in simulation run 2.1 after integration of the IFC model. Two heat flow rates from radiators in the HVAC module are expanded and their space association illustrated. Classes and instances from the IFC model are framed in green, from FMUont in blue. (yellow circles indicate classes; purple diamonds indicate individuals)

Run 2.2 - Calibrated Ground Heat Pump with modified Multi-Zone Model

A second simulation of the FZK-House is intended to quantify the energy savings due to a different usage of the Gallery. The design option projects the second floor to be entirely used as a non-occupied attic with reduced air temperature. In order to consider this in the building envelope module, the definition of thermal zones in the multi-zone model needs to be changed. Figures 89 and 90 show the resulting thermal zones in the floor plans. The Common Area now consists of the Kitchen, the Hallway and the Living Room. Opposed to above, the second floor is now incorporated in its own thermal zone, the Attic, and associated accordingly.

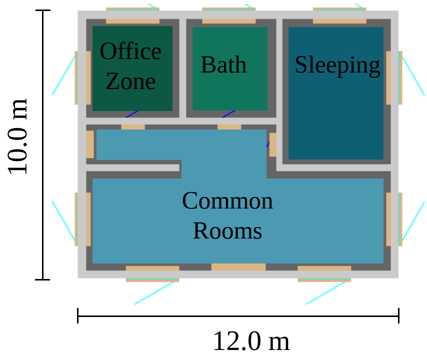


Figure 89. Zoning of the first floor of the FZK-House in simulation run 2.2.

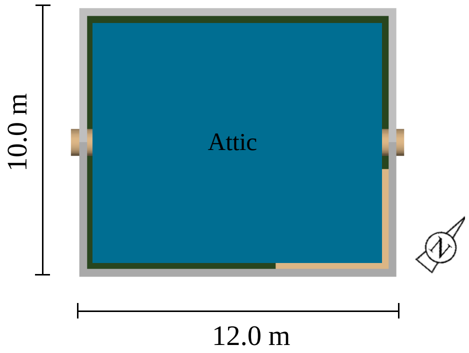


Figure 90. Zoning of the second floor of the FZK-House in simulation run 2.2.

The introduction of this change into the simulation ontology leads to the adaptation shown in Figure 91. Two variables are expanded, identical to Figure 88. The association to radiators remains equal. Through these, the spaces, namely the Kitchen and the Gallery, where the radiators are located in, can be identified as above. Since the zone affiliation of the Kitchen did not change, *QRoom[2]* is still associated with the Common Rooms. In the case of *QRoom[18]*, the zone association changes since the Gallery is now part of the thermal zone Attic. Hence, different connections are derived. Among the 8,402 the reasoner infers 68.

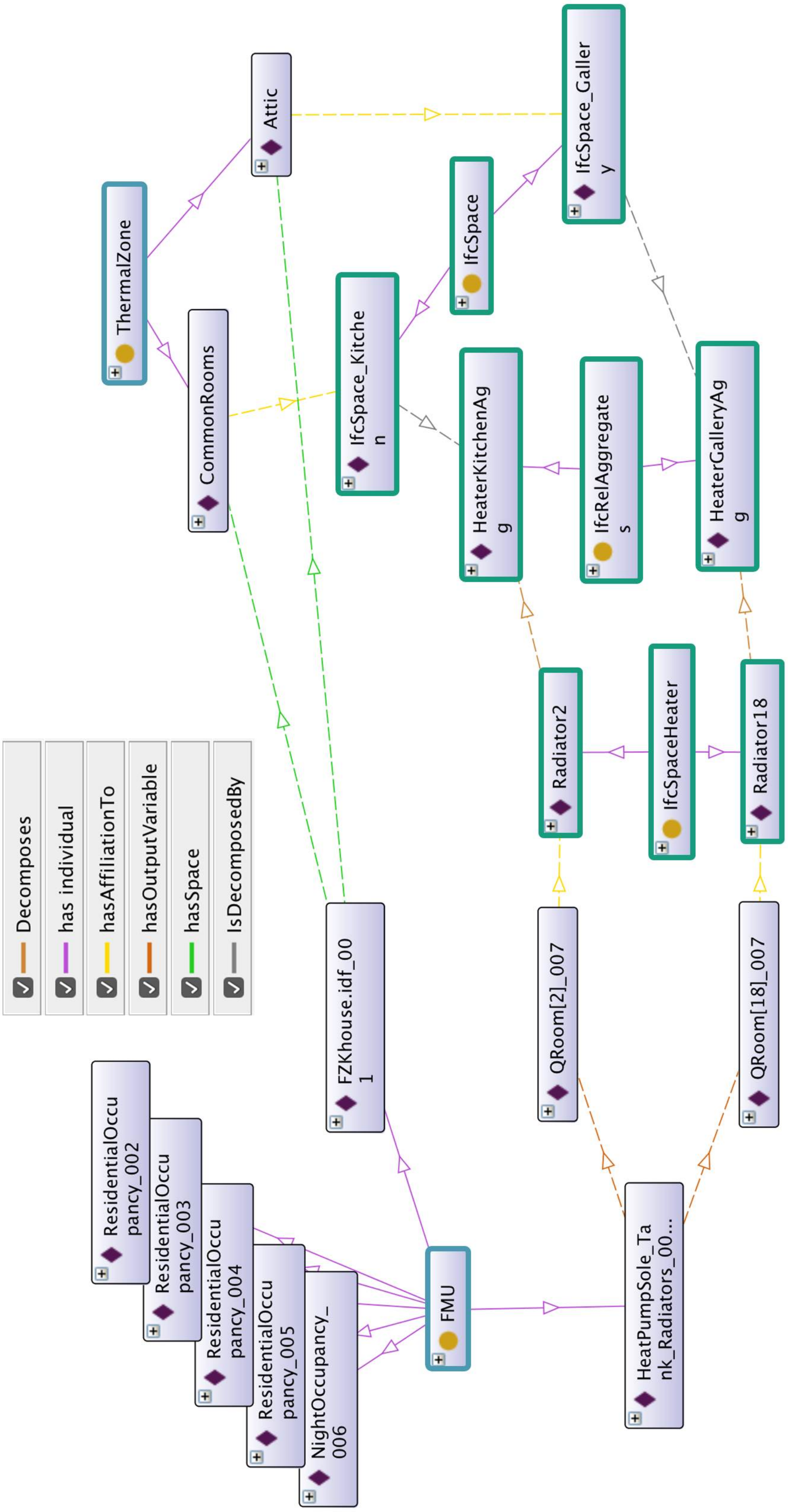


Figure 91. An excerpt of the simulation ontology in simulation run 2.2 after integration of the IFC model. Two heat flow rates from radiators in the HVAC module are expanded and their space association illustrated. Classes and instances from the IFC model are framed in green, from FMUont in blue. (yellow circles indicate classes; purple diamonds indicate individuals)

Figure 92 illustrates the resulting temperatures in the Common Area and the Attic after the exclusion of the Gallery from the occupied zones. It shows the decreased temperature level in the Attic and the heating response to a time-varying setpoint in the Common Rooms. The exclusion of the Gallery and the corresponding radiators from this zone does not affect the ability to generate the desired air temperatures. Due to the decreased temperature in the Gallery compared to simulation run 2.1, the required heating energy is now lower, as shown in Figure 93. Over the course of a year, savings in heating energy of 6.7% are quantified in the simulation.

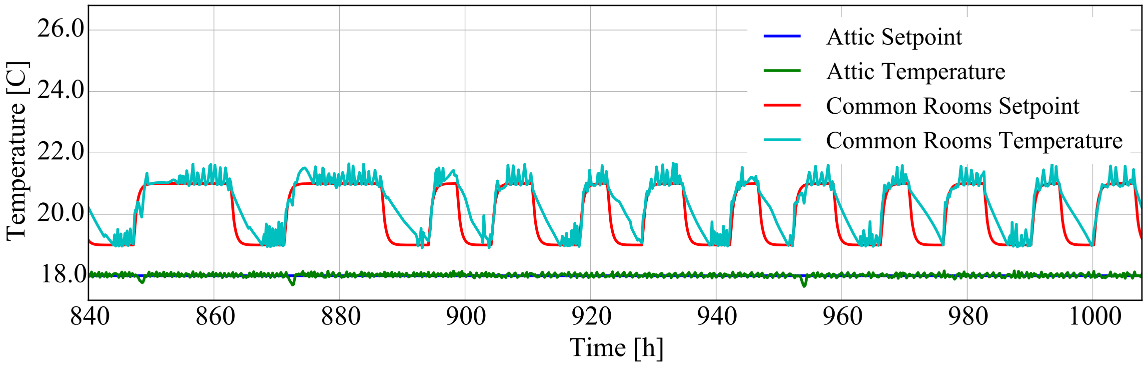


Figure 92. Air temperatures in the Common Area and the Attic for two winter weeks in simulation run 2.2 along with the setpoints.

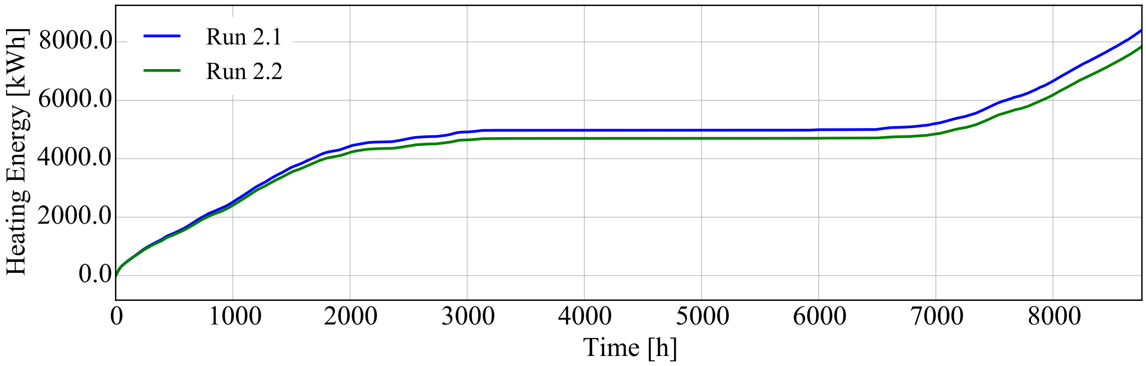


Figure 93. Comparison of required heating energy in simulation run 2.1 and 2.2.

Chapter 5

Critical Evaluation of the Methodology

The following chapter provides a critical evaluation of the developed methodology for a scalable BPS based on modular components. The discussion is divided into an assessment of the theoretical concept and its application in practice. The former is primarily concerned with technical implications of the methodology as well as its potential to contribute to the development of next generation BPS techniques. The latter treats its implementation and the resulting benefits for the daily design process as well as current restrictions.

5.1 Theoretical Aspects

The theoretical concept of the approach is aimed at mediating the development of future simulation processes. Hence, the methodology is firstly evaluated regarding the criteria for next generation BPS tools and procedures identified in the literature, as elaborated in section 2.2.1. This serves to estimate the potential of the presented solution to overcome the current shortcomings in BPS. Subsequently, the agreement with the characteristics of a modular simulation is examined and restrictions implied through the deployed technologies and developed concept are discussed.

5.1.1 The Wish-List Criteria

Crucial criteria for a fruitful application of BPS in the design process have been determined to be scalability, multidisciplinarity, openness, as well as an integrated and dynamic character of the simulation. In this sense, scalability refers to the continuous application of a simulation along the design process while taking into account the increasing quantity and quality of information as well as the varying relevance of different KPIs. Multidisciplinarity describes the possibility for several stakeholders to contribute to the simulation. This especially triggers benefits through immediate performance feedback for anticipated design options from individual planners. An open character of the simulation is crucial to this and presents the pre-condition for a broad application and further development in the research community. Furthermore, the advanced requirements and technologies in the building sector have increased the need for an integrated view. Models must encompass several domains and consider their interactions. Similarly, dynamic models have grown in importance in order to consider fluctuating boundary conditions and system behavior at a high temporal resolution. In the following, the developed methodology is evaluated based on these criteria.

Scalability

In order to support designers in the sense of a continuous assistant, the simulation methodology incorporates three models representing the building envelope's response to its surrounding domains. Namely, these models comprise a single-zone, a multi-zone and a zonal airflow representation of the building or parts of it. The chosen models provide the capability to simulate with an increasing LOD due to a growing amount of information along the design process. The model selection also allows for assessing different KPIs, i.e. energy usage and indoor thermal comfort, that may become relevant during different stages of the design phase.
The focus of the methodology is set on the integration of different LODs for the response model, i.e. the building envelope representation, of the simulation. This is due to the resulting implications of the different physical modeling approaches on the information requirements, as discussed in chapter 3. Hence, a scalability regarding spatial resolution was prioritized. The determined information levels comply with the growing information pool during a design process starting without the need for a digital building model until a room- and object-based integration of information into the simulation. A detailed assessment of information availability and requirement during individual design phases can lead to a further, improved adaptation of the simulation to the design process. Regarding other domains, provided that models comply with the defined domain boundaries, also different levels of complexity can be integrated. This ranges from steady-state to dynamic models and also involves empirical correlations. Similarly, also within building envelope models an increase of internal accuracy can be integrated, such as the extension from a single-layer to a multi-layer representation of a wall. The implementation of such functionality is, however, left to the modeler and the authoring tool. Since no implications for the connections to other simulation modules result, an adaptation to improved information availability on this level can be realized without adaptations of the procedure. Nevertheless, the continuous update to such an information growth must still be realized in order to provide the desired self-evolving simulation model.

Multidisciplinarity

Enabling multiple users to operate, execute and modify a simulation model is best realized through the co-operation of several process models. Section 2.2.5 refers to this advantage following the remarks from Clarke and Hensen [17]. Through its modular nature the methodology follows this concept and allows for several users to provide and manipulate own modules that can be tested in conjunction with the remaining system. The standardized communication through the usage of FMI furthermore ensures the ability to combine models from heterogeneous sources. This enables users to generate models in their preferred tool environment. Besides planners, the aspect of multidisciplinarity also comprises other stakeholders, such as manufacturers, building owners etc. The FMI presents a suitable technology for realizing the integration of models from such sources. In addition to the possibility for a common communication protocol, the encapsulation of source code and the resulting preservation of know-how is an important characteristic on the path to enhanced model sharing. The latter especially increases the possibilities of manufacturers to provide access to product specific simulation models that can be deployed within the developed simulation methodology, as applied in simulation run 1.5. This can ultimately lead to a quality improvement of simulation results since product specific models, eventually calibrated with measured data, can be included in the simulation. Within the developed method, the seamless introduction of these modules can be realized through enrichment of the FMUs with semantic information corresponding to the overarching data model. The usage of OWL to formulate this information model allows for referencing locally distributed modules within this framework and provides the appropriate characteristics to realize the integration of models from multiple sources. Hence, the deployed technologies present a sound basis for providing access to a variety of stakeholders to the simulation. In this sense, also their properties regarding openness of the simulation become significant, as discussed in the following.

Openness

As an established standard for model exchange and co-simulation, the FMI ensures the openness of the methodology for all parties. The implementation of the interface in tools allows to lift simulation models on a common, universal communication level that is the basis for bridging the issues arising from a heterogeneous group of model developers and tools. The open standard guarantees the possibility for every stakeholder to become involved with own models. Furthermore, the cross-linking of the distributed modules through OWL allows for several opportunities to extend and customize functionality. One public, central data model, such as the presented ontology, is required to ensure a common understanding by setting single modules into the same context. Such a data model can be the center of a common development of modular simulation techniques. Its accessibility through the www and the formulation in the standardized OWL provide the open platform for multiple distributed teams to work on enhancements regarding its capabilities. This can lead to additions by means of new module categories or functionality. Furthermore, this data model can be associated and expanded with several existing or new data models. This can yield in individual solutions customized for specific issues, such as building automation. Other fields of application can therefore be connected and granted access to functionality provided by simulations and vice versa.

Integrated Simulation

The integration of several domains within the simulation is achieved trough the modules representing these domains. Hence, the interactions between different planning domains can be quantified and considered in the simulation. At this stage, modules originating from the domains determined in section 3.1.1 can be integrated. An integration of modules from other fields, such as power consumption, requires an extension of the semantic type description corresponding to the relevant domain.

The holistic aspect is maintained throughout the different LODs of the simulation. When setting the focus on the indoor climate in single parts of the building, also the behavior of the remaining, interacting parts of the building are integrated. This also allows for considering other domains, such as HVAC, in their entirety. As an example, while assessing a single room of the building in more detail, simulation runs 1.8 and 1.9 account for the load of the entire building and therefore the holistic reaction of the HVAC system. This feature allows for combining the indoor climate assessment of a single room with the dynamics of the HVAC system being considered for the entire building.

Dynamic Models

The formulation of a model depends on the modeler and the modeling tool respectively. Through the usage of FMUs, the foundation for integrating dynamic models in the developed methodology is provided. However, also other model types, e.g. of statistical or empirical nature, can be applied. Simulation run 1.7 illustrates this with the integration of stochastic models in order to represent the occupants' behavior in different zones. The reduction to input and output variables enables the combination of different model types. The resulting flexibility exceeds the functionality of most simulation tools and also allows for integration of individual models, such as derived from measured data or inquiries. Ultimately, the model type depends on the users' preferences and intentions.

5.1.2 Modularity

In order to provide a common understanding of modular simulation techniques, Mazzarella and Pasini [74] defined four layers of modularity, as discussed in section 2.3.1 and briefly described in the following.

Functional layout modularity targets the application of modules as re-usable and combinable entities from a user's perspective. Mathematical models modularity assumes the persisting integrity of a resulting simulation, after module changes or additions. The possibility for communication among modules and their numerical solution is considered in the standardized mathematical models modularity, while modularity on code level poses the fourth level. The following paragraph evaluates the developed methodology based on these definitions and determines the considered layers of modularity within the approach.

Functional Layout Modularity

The functional layout modularity describes the topmost level of modularity, which is directly interacting with a user. In the developed methodology, this interaction is realized through the selection of library or self-generated FMUs. The FMUs behave like modules that can be exchanged and re-used. An exception in this regard are modules from the building envelope domain. Due to the generally unique character of buildings they have to be re-generated for each project individually. Even within a project multiple re-generations can occur due to the restrictions of FMUs to be editable only through parameters. In order to facilitate this effort, the in chapter 4 mentioned SketchUp plugin was implemented to generate building envelope FMUs directly from a CAD model. This reduces the barrier to quickly evaluate architectural design changes and compare design options. Models from the remaining domains feature commonly usable patterns that can be applied beyond single projects and are therefore suitable to be maintained in libraries. These patterns can be found in data-sets of identical structure, such as weather data, but also in models describing people's behavior and occupancy. Similarly, models incorporating the physics of devices from batch production, such as found in HVAC systems, can be re-used across several projects.

In addition to the separation of the simulation into single modules, operable by a user, the agglomeration process is automated in the developed methodology. This limits the effort for a user to module selection and avoids time-intensive manual updates of the simulation topology. As such, the developed procedure can be seen as a self-adapting modular simulation offering the desired functional layout modularity with restrictions regarding the building envelope domain.

Mathematical Models Modularity

In the developed process, the simulation functionality is inherited within the single FMUs. Interaction with other modules is limited to the input and output variables of each FMU following the paradigm of mathematical modularity. The methodology allows for generation and modification of modules individually without affecting the resulting co-simulation. This is primarily realized through avoiding the necessity to having knowledge about other modules during editing or exchanging modules (section 5.1.3 discusses this aspect in more detail). This also allows to change the mathematical description of a model. As mentioned earlier, various model strategies may therefore be included, such as stationary or dynamic formulations, empirical or statistical logics or also the consideration of different physical phenomena within a model may differ, as long as the input and output variables can be related to domain boundaries and follow the developed description pattern. In terms of the building envelope, mathematical modularity is also maintained across three LODs, each characterized through a different physical view of the building. This functionality requires a different level and handling of meta-information, however, if provided with the corresponding data, the integrity of the simulation topology can be maintained across the considered LODs. Hence, the mathematical models modularity is provided through the methodology.

Standardized Mathematical Models Modularity

Through the usage of the FMI specification, standardized communication between modules is ensured over the entirety of the methodology. In contrary to modularity concepts limited to a single tool, the developed method separates the numerical solution of modules and the solution of the overall simulation system corresponding to the principles of co-simulation. The solving algorithm for single modules is inherited within the FMUs. Following the FMI specifications, the source tools generate an executable algorithm during the translation of a model to an FMU. The nature of this algorithm corresponds to the tool-specific functionality. Hence, contributing modules do always contain an executable numerical solution as individual entities.

In order to achieve a solution of the entire simulation consisting of several modules, the methodology relies on the in section 2.3.2 discussed algorithms for co-simulation. Within the implementation a loose-coupling algorithm was realized. Executed at small time steps, this algorithm ensures the numerical correctness of the simulation, which is still preserved when exchanging, adding or removing modules. Similarly, the remaining coupling algorithms, as often applied in the mentioned platform solutions, can be applied to maintain the numerical integrity of the simulation.

Hence, the aspects of standardized mathematical modules modularity concerning communication among modules and numerical solution of modules as individuals as well as part of an overall system, are fulfilled.

Code's Modularity

Since the source code of models is generally not accessible within FMUs, this level of modularity can not be integrated in the developed approach. The functionality to re-use and benefit from base classes and incorporated functionality for facilitated modeling is left to the source tool of an FMU and therefore features a clear separation from the developed methodology. The ease of implementing a model lies within the responsibility of a tool. An object-oriented paradigm can be one path to realize this goal. However, also other approaches my be suitable depending on the targeted issue. The modular simulation approach generated in this thesis merely allows to combine the various strategies via the FMI standard.

Goals for Modular Simulation

In concluding the discussed levels of modular simulation and their complete or partial fulfillment in the developed methodology, the achievement of goals for a modular simulation, namely system decomposition, model re-use, extendibility and system composition modification, is assessed in the following. The decomposition of a BPS is successfully realized using an inter-domain approach. This step is the basis for the developed methodology. It guarantees the compatibility of modules and allows for a separation of module responsibilities corresponding to a clear thematic pattern. Following the aspects of functional layout modularity, modules can be re-used within a design process and beyond single projects. Regarding this, an important characteristic of the methodology is the definition of information layers for the description of FMUs and the exclusion of project specific information in the general Abox description. This allows for re-using the FMU-specific Abox beyond projects. An exception concerning model re-use poses the building envelope domain as discussed above. Extendibility of the simulation is provided throughout the entire process. Besides the possibility to consider models in different LODs, this aspect contributes to the scalability of the methodology, since more phenomena can be considered in the simulation through the addition of new modules. Simulation run 1.7 exemplifies this through the integration of occupant behavior models. A modification of the system composition through the exchange of modules is enabled through the defined model boundaries and the automatic derivation of the simulation topology. An example is the change from simulation run 1.3 to 1.4. The HVAC FMU, originally incorporating a solar collector as secondary heat source, is exchanged with an HVAC FMU solely relying on a heat pump. Since model boundaries remain equal and the altered variable connections are updated automatically, the exchange can be realized instantly. Hence, the integrity of the simulation after modification of the system composition is ensured. This aspect is a precondition for the application of optimization algorithms to the modular simulation in order to find an optimum system composition.

5.1.3 Functional Implications of the Methodology

The following section discusses the implications of the FMI standard and the usage of BIM on the functionality of the developed simulation methodology. Especially the relation between information and simulation functionality regarding scalability of either during the course of a project is illuminated.

Cardinality of Connections

The methodology was developed to meet the above-mentioned criteria. As such, the possibility to exchange modules and recombine them through the usage of library models is crucial to the concept of modularity. The chosen information layers withstand this principle. However, a further pre-requesite to accomplish this goal is the independence of other modules when generating a module. Naturally, the compatibility of modules regarding model scope needs to be considered. With regards to this, the chosen inter-domain decomposition leads to clear boundaries between modules and provides the basis for compatibility regarding the covered matter of the modules. However, the transmission of content by means of exchange variables must also be compatible. The following discussion therefore targets the influences of cardinality within the modular simulation.

In co-simulation setups, the FMI standard implies a one-to-one or one-to-many relation of input and output variables. While for many applications this may be the basis for clear patterns in the topology derivation, for the field of BPS, the possibility of a many-to-one relation can provide significant advantages. In the following, this aspect is discussed based on the example illustrated in Figures 94 to 96.

The case considers the connection of an HVAC FMU with a multi-zone building envelope FMU. The building is sub-divided into several thermal zones. Due to the neglect of local positioning of heating units, each zone features a single input for an accumulated heating rate. The HVAC system features a number of heating units providing these heating rates. The implied ono-to-one cardinality for variable connections leads to the requirement of an identical number of heat flows and thermal zones, assuming that all thermal zones are heated. Hence, when choosing or modeling an HVAC FMU, the author must already have knowledge about the number of zones in the building model and aggregate heat flows from heating units in the same thermal zone. This leads to the same number of inputs and outputs and allows for direct connections as depicted in Figure 94. However, the adaptation of the HVAC FMU to the project specifics degrades the possibility to re-use the HVAC FMU and as such, is not aligned with the principles of modularity.

Another possibility to solve this issue is presented in Figure 95. In this case, the HVAC FMU provides heat rate outputs for each individual heating unit. Hence, the re-usability of this module is generally possible and feasible across various projects. The building envelope FMU provides an identical number of corresponding inputs. This requires multiple heat flow rate input variables in each thermal zone according to the number of heating units in this thermal zone. Hence, when modeling the building envelope FMU, knowledge about the HVAC system must already be present. Due to the fact that a building envelope is generally a unique model and can not be re-used, the above disadvantage is negligible. Nevertheless, the principles of modularity are not respected, since a possible exchange of the HVAC FMU might now require adaptations in the building envelope FMU.

Opposed to the one-to-one relations applied in the two mentioned solutions, a third option lies in the application of a many-to-one connection of variables. Figure 96 illustrates this strategy for the present example. A building envelope FMU features one heat flow input per thermal zone. The HVAC FMU features heat flow rate outputs for each heating unit. The association to the corresponding objects in a BIM allows for connecting these to thermal zones. In the case of multiple heating units in a thermal zone, a many-to-one connection is set up. This solution avoids the disadvantages mentioned above. The generation or choosing of simulation modules can be realized without knowledge about other modules. Exchange of modules does not affect the integrity of the simulation topology, since connections are merely updated and a modification of other modules is avoided.

The example suggests that besides the modeled content, also the chosen exchange variables need to be set according to the domain-specific knowledge. A many-to-one cardinality can then still lead to a flexible module exchange and re-usage. In the HVAC domain, a modeler is aware of the number of heating units that are applied and can choose the number of outputs accordingly. In the building envelope domain, a modeler has no knowledge about the specifics of the HVAC model. However, since no differentiation is made for spatial distribution of heating units in a zone, it is always feasible to generically define a single input for a heating rate per thermal zone. The association of heating units in the HVAC domain with objects in a digital building representation allows for determining their corresponding zones, which ultimately leads to aggregated heat flows.

Within the implementation of this thesis, the many-to-one cardinality was ensured during the exchange of variable values. The master-algorithm orchestrating the co-simulation aggregates the values of output variables for input variables with multiple connections (see line 41/42 in Listing 10). This presents a solution, however, it is arguable if the responsibility to realize the aggregation of variable values lies within the master-algorithm. Alternatively, such functionality can be inherited in the simulation modules, requiring an extension of the FMI standard in order to enable a many-to-one connection.

An example for realization of many-to-one relationships can be found in the Modelica language. The modeling language allows for the definition of different physical connectors, such as fluid or heat flow connectors. The connector models contain the relevant physical equations in order to combine an arbitrary number of inputs in one connector and forward the resulting information to a further component according to the governing physics for aggregation of the flow. Opposed to the transmission of single values, these connectors require the transmission of a defined set of variables in order to unambiguously define the physical state of the flows at the connector. The concept ensures the flexibility increase or decrease the number of inputs and therefore enables the discussed many-to-one cardinality.

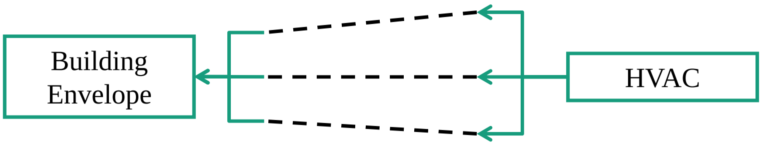


Figure 94. Application of the one-to-one cardinality for the example of a building envelope and an HVAC FMU. The chosen relations imply knowledge about the characteristics of the building envelope model when choosing or modeling the HVAC FMU.

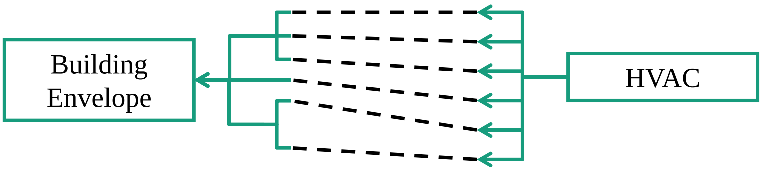


Figure 95. Application of the one-to-one cardinality for the example of a building envelope and an HVAC FMU. The chosen relations imply knowledge about the characteristics of the HVAC model when modeling the building envelope FMU.

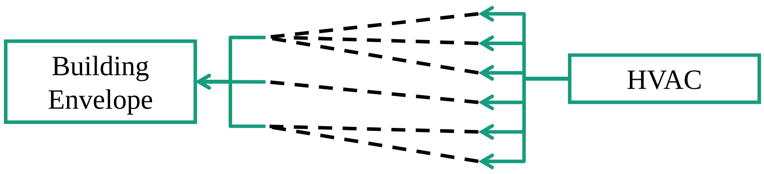


Figure 96. Solution of the cardinality issue through introduction of a many-to-one relation for variable connections. The aggregation of variables is transferred from within the model formulation to the definition of variable connections.

The aggregation of multiple variables to a single value is not possible for every variable type. In the implementation, this issue is solved through allowing only variables of certain physical quantities, such as heat flow rates, to be aggregated. Regarding other exchange variables, often of non-physical nature, this leads to restrictions in the simulation functionality, as discussed in the following.

Influence of Information on Simulation Functionality

The many-to-one relation between variables increases the flexibility of a modular simulation, however, several quantities are not of cumulative nature, among others temperatures, signals or relative humidity. For such variables functional restrictions within the developed modular simulation exist.

As an example, the opening of a window through the action of an occupant is considered. In a single-zone representation of a building envelope, the provided information to derive the simulation topology does not allow for differentiation between more than one input variable responsible to trigger a window opening. Differentiation would require spatial information, i.e. the relation of these variables to specific building objects, in this case windows. However, this would also require the introduction of a digital building data model at the stage of a single-zone simulation and therefore lead to increased information requirements at the beginning of a project. In order to avoid this and comply with the course of available information at different project stages, a single-zone model is limited to one input for a window opening signal. Hence, only one signal for controlling the opening of windows can be provided in the simulation. Opposed to this, cumulative variables are aggregated and can therefore be considered regardless the number of variables.

Accordingly, a multi-zone model, leads to the possibility of opening windows in multiple zones of the building. However, within a thermal zone, again only one opening trigger can be considered in the simulation, whereas multiple heat flows from heating units can be aggregated to form the heating rate in this zone. In this case, the association to a digital building data model must already be realized to derive the simulation topology. Signals for each window could therefore also be associated with the corresponding BIM objects. Hence, the differentiation of window opening within a thermal zone would simply require the reasoning for direct associations to objects in a digital building representation similar to the reasoning process for zonal airflow models. Signals can then be differentiated for single individuals in the room just as several heat flow rates from heating units can be considered individually.

These aspects illustrate the idea of the chosen levels of information and their reasoning approach to present the required information minimum in order to compose a simulation of the desired LOD. Enriched information and enhanced reasoning can lead to more functionality for the individual simulation LOD, especially regarding variables of non-cumulative nature. Variables that can be aggregated can feature this functionality when applying the discussed many-to-one relation.

5.1.4 Simulation Decomposition

The decomposition of the simulation was realized with an inter-domain strategy in order to provide clear boundaries for the scope of the individual modules. As an alternative, the intra-domain approach features the advantage of higher accuracy, as discussed in section 2.3.2. However, the unregulated module boundaries pose a threat to its feasibility when used in widely applicable methodologies, such as developed within this work. Instead, single use cases or implementations with a limited number of fixed tool interfaces are suitable scenarios for the intra-domain decomposition. The inter-domain decomposition furthermore respects the responsibilities of planners and allows for each engineer and architect to contribute with domain-specific expertise.

Besides the mentioned benefits for the overall approach, the inter-domain decomposition features two additional aspects that need to be considered. Many recent innovations in building technology are aimed at exploiting synergy effects from several domains. Among others, these encompass thermally activated slabs or floor heating. A clear assertion of these highly integrated components to a single domain can not be realized since functionality of either domain is incorporated within the component. As an example, a floor heating system is considered. The separation of module functionality can be realized as described in the following. The HVAC domain computes the thermal heating rate provided by the system to a room. Therefore, the heat conduction from the heating fluid through the floor layers is modeled within the HVAC module while considering the thermal capacity of the floor. The building envelope module receives the heating rate and computes the thermal response of the building. In this module the floor is modeled without the liquid-filled pipes of the floor heating system. Regarding the expertise of planners, this configuration features a reasonable differentiation of responsibilities between e.g. an architect and an HVAC engineer. However, drawbacks regarding the correctness of floor surface temperatures in the building model and the resulting long-wave radiation heat transfer in the building envelope model occur. While the surface temperature of the floor heating system may still be retrieved from the HVAC module, the consideration of correct short-wave radiation within the room is not possible. As an alternative, different module boundaries may be chosen. For example the exchange of surface temperatures instead of heat flow rates may be preferred. This, however, requires functionality that is currently not provided in every tool, as discussed in the following chapter. A further option is the modeling of the floor heating system within the building envelope module based on the thermodynamic state of the heating fluid at the inlet to the heating system. Similarly to the first option, tool functionality regarding model boundaries not always allows for such a solution. Furthermore, one might argue that the design responsibilities in this case are not assigned appropriately since the heat exchanging system is part of the building envelope instead of the HVAC domain.

Another implication resulting from the inter-domain approach is the limited functionality to test modules at smaller granularity. The chosen decomposition only allows for exchanging modules with their scope corresponding to the domain boundaries. Regarding, e.g. the HVAC domain, this leads to the necessity of exchanging entire systems instead of single HVAC components. In order to enable manufacturers of single HVAC parts to provide FMUs for the simulation, a smaller module granularization must be realized resulting in an intra-domain decomposition. Hence, a definition of module boundaries within domains must be established.

5.2 Application Aspects

Analogue to the evaluation of the theoretical concept, the procedure is in the following discussed regarding its application in practice. The chapter therefore begins with an assessment of involved technologies and their current shortcomings to fully exploit the possibilities of the developed methodology. A closing evaluation of the simulation time based on the studies from chapter 4 serves to identify further requirements in future developments.

5.2.1 Current Limitations

As discussed in section 2.3.1, simulation tools have been originating from particular needs and motivation. Hence, the design and structure of tools was often adapted to specific issues while building on the current best practice solutions for tool development. Tools, such as EnergyPlus, incorporate a fixed DAE that is built upon pre-defined variables which feature a pre-determined location within the DAE. The current implementation of the FMI in EnergyPlus limits exchange variables to certain variable types within the DAE. This prevents the input of e.g. surface temperatures into an EnergyPlus model during the co-simulation, as mentioned above. Other tools, such as TRNSYS or Modelica provide more flexibility when defining exchange variables for FMUs. This is due to the generation of the solution algorithm after model formulation and the therefore possible adaptation to the defined exchange variables.

Further issues are seen in the limited support of the FMI standard in BPS tools. Currently (*September 2017*) 103 tools offer an FMU export or import [81] functionality. However, the adaptation rate in the building sector, especially of the newer FMI 2.0 standard, is low. Disadvantages of the FMI 1.0 range from missing information for an increase of co-simulation-solver efficiency to the dependency on a local installation of the FMU exporting software. As an example, TRNSYS and EnergyPlus feature an implementation of the FMI 1.0 standard in tool co-simulation mode. Hence, a completely tool-independent co-simulation is not possible when involving FMUs from these source tools. Full support of FMI 2.0 could ensure a growing number of simulation modules available for reuse as flexible black-box units applicable on every platform independently from tool installations. The increasing popularity in other industries, such as automobile, can help to mediate a faster adaptation of the standard in the building sector.

5.2.2 The Time Dimension

Section 2.2 mentions the different approaches of current simulation tools to solve a resulting equation system from a model formulation. Traditionally, pre-defined DAEs, as mentioned above, are used, which are populated corresponding to the model in question. Other approaches determine the solving algorithm individually depending on the current model specification. In order to optimize the computation time, the algorithms are tailored to the models. Hence, when creating a holistic building model in a single tool, the tool is able to consider the characteristics of the entire model during the optimization of the solving algorithm. When decomposing the model into several parts, each modeled in a different tool, this optimization process can only be realized for a fragment of the overall model. Additionally, a master algorithm is required which is able to combine the solving algorithms of each fragment while maintaining the numerical correctness of the overall solution. These aspects result in disadvantages for the co-simulation approach regarding the duration of a simulation. Therefore, the following section discusses this issue and provides references for future work based on the examples from chapter 4. In addition, a summary of the executed simulations regarding time performance and further characteristics is provided in appendix C.

In order to differentiate between the single processes during the co-simulation, the duration of each simulation from chapter 4 was recorded regarding three parts of the implemented loose-coupling algorithm: connecting variables, simulating FMUs and retrieving variable values. Figure 97 provides an overview of the recorded times. At this stage it is reminded that simulations were carried out with a time step of two minutes for an entire year with the exception of simulation runs 1.8 and 1.9 where the simulation time was one week at the same time step. The largest share of the total simulation time is in all cases the net duration for simulating the FMUs. The variable exchange and retrieval are secondary but considerable portions of, in general, comparable amount. The percentage for variable exchange from the total simulation time varies between 1% and 13% depending on the net FMU simulation time. In run 1.1, the low number of realized connections leads to a portion of 4%. Run 1.2 has considerably more connections, however, net simulation time is still relatively low due to the single-zone LOD. The multi-zone simulations show low variation. Generally, the time required for variable exchange is 10% of total simulation time in these cases. Run 1.8 and 1.9 are exceptions due to the introduction of the zonal model. Its relatively high net simulation time leads to a negligible portion of 1% for the exchange of variable values. When including the time required for retrieval of variable values, the mentioned percentages are doubled in most cases. Solely run 1.1 poses an exception, since the number of retrieved output variables is high compared to the realized connections.

When considering the total simulation time of each run, the increasing LOD is recognizable. From run 1.1 to 1.2 the introduction of an advanced HVAC model can be seen. A distinct gain can be detected for the subsequent adjustment to a multi-zone building envelope model and the resulting increase in the number of contributing FMUs. While the multi-zone simulations are generally at a similar level, the exclusion of the solar collector from the HVAC module is still recognizable. A strong increase comes along with the refinement of the office zone into single zones and the consideration of occupant behavior in these zones. Similarly, the total duration from run 2.1 to 2.2 increases due to the addition of a further zone. The maximum duration for the simulation of a year amounts to 1.5 hours whereas a simulation of a week including the zonal model totaled to 0.75 hours in run 1.8.

In conclusion, the time required to execute the co-simulation is within an acceptable time frame. However, improvements of simulation time are still desirable, especially regarding the vision of immediate response-feedback for a designer or the possibility to run a holistic optimization algorithm, which is evaluating different components and parameter settings. In order to increase the efficiency of master platforms, the present study suggests that a pre-assessment of the individual FMUs and their computational effort as well as the number of exchanged variables should be considered in the master algorithm.

The results show that the scalability of the simulation is recognizable in the simulation time. This is partly due to an increased number of FMUs, however, the primary factor is the increased net simulation time for more detailed individual models. Hence, when allocating computational resources, a master platform should favor these models in order to considerably decrease simulation time. Similar behavior is found for the individual processes involved in the master algorithm. While a decrease of the portions for variable exchange and value retrieving in the co-simulation is generally possible and would lead to considerable improvements, their significance is highly influenced by the net simulation time of the FMUs. Hence, with computationally expensive models, these portions become less relevant. However, for fast FMUs and an increased number of exchange variables, these processes can significantly extend the total simulation time. For master platforms, which successfully target the aspect of a reduction of net simulation time, they will become increasingly important, since their duration will then dictate the simulation speed.

The variation of LOD and hence the addition of new FMUs as well as more detailed models, significantly influence the simulation speed. In order to investigate the influence of the time step length in the implemented loose coupling algorithm, simulation run 1.6 is executed with two additional alternatives, a time step of five minutes (run 1.6.1) and a time step of ten minutes (run 1.6.2). Figure 98 summarizes the required time intervals corresponding to the scheme applied above.

Due to the increasing time step and the therefore reduced number of interaction points along the course of the simulation, the time for value exchange and retrieval decreases from sim-

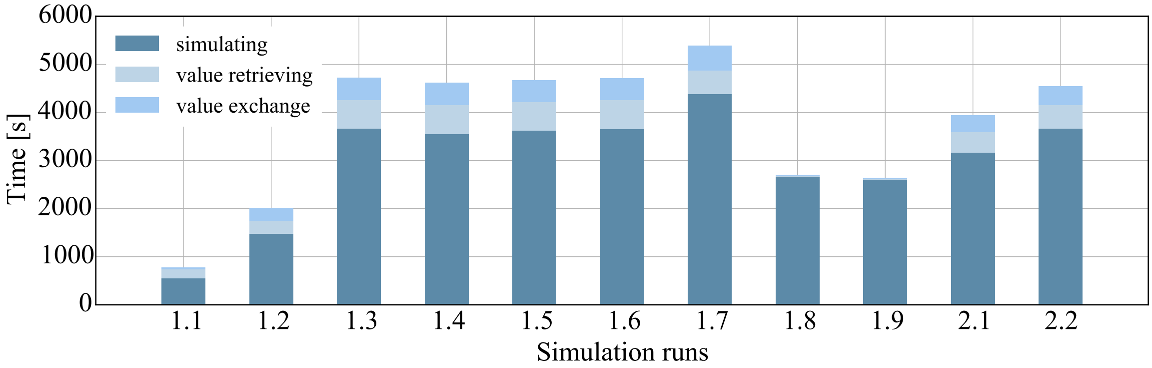


Figure 97. Comparison of required time for connecting variables, simulating FMUs and retrieving variable values for the simulation runs from chapter 4.

ulation run 1.6 to 1.6.2. This can not only be recognized in a net decrease, but also in the share compared to total simulation time. Initially amounting to 10% in run 1.6, the portion decreases to 5% and 3% in the following simulations. While this corresponds to results one might expect, the development of net simulation time is counter-intuitive, as discussed in the following.

Since the time frame of the simulations remains identical and therefore the scope, which is covered by the simulation, one might expect identical net simulation time over all three alternatives. However, the extended time steps lead to longer advances in the individual FMUs before interacting with the co-simulating models. This leads to larger differences regarding the exchanged variable values from one time step to another and hence an increased unsteady change which must be considered in the internal solver of the simulation units. It could be expected that this causes longer simulation times in order to reach convergence. Opposed to this, the net simulation time considerably decreases from simulation run 1.6 to 1.6.2. This leads to the assumption that a significant time is required when changing the status of an FMU from hold to running and vice-versa at each time step. In fact, the net simulation time inversely correlates with the length of the time step approaching a minimum with an increasing time step. At this minimum, the net simulation time is reduced to the tasks solely related with the computation of the physical model. This leads to the conclusion that simulation platforms, which target the overload due to triggering FMUs to start and stop a simulation step, can provide considerable improvements in simulation speed. In conjunction with this, the role of the length of the time step needs to be of further focus in the implementation of co-simulation algorithms. The following paragraph provides further insights concerning this aspect.

As mentioned in section 2.3.2, the accuracy of the loose-coupling algorithm relies on a small simulation time step. The influence of a coarse time resolution on the simulation with the altered time step in simulation runs 1.6.1 and 1.6.2 is shown in Figures 99 and 100. The former depicts the heating rate in the Lounge, the latter depicts the resulting air temperatures for the second half of a winter day. Both graphs are characterized by high fluctuations,

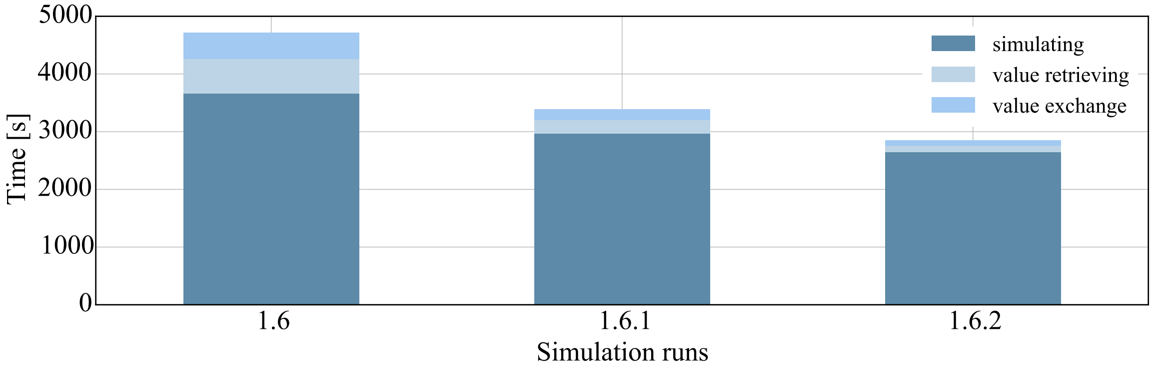


Figure 98. Comparison of required time for connecting variables, simulating FMUs and retrieving variable values for simulation run 1.6 with a variation of the simulation time step from two to ten minutes.

which grow with an increased simulation time step. It can be seen that the modeled control mechanism in the HVAC FMU, which regulates the heat flow rate corresponding to a given setpoint temperature, requires small time steps to provide proper results.

With a ten minute time step, the heating rate oscillates considerably. When provided with a heat flow rate, the building envelope FMU keeps this value constant over the entire time step. Hence, when provided with a high heat flow rate, the air temperature exceeds the setpoint within this time. At the next time step, the large change and high gradient cause the control mechanism to significantly decrease the heat flow rate. This in turn, results in a low air temperature after executing the following time step and restarts the cycle. This effect looses in significance with a decreasing time step. When entering an area where the setpoint is yet to be approached, as shown after hour 1002, the influence of the time step can be neglected. In these conditions, the modeled control maintains a similar temperature gradient for each time step.

To conclude, the time step has a strong influence on the simulation results. However, this influence differs for variable types and even for single variables over the course of the simulation. Figures 99 and 100 show the dependence of continuous physical quantities on the time step. The derived conclusions would differ for deterministic variables, e.g. from schedules, where the time step length might only lead to restrictions regarding a proper resolution of the schedule, but not a varying behavior. Hence, the nature of exchange variables needs to be considered when choosing the time step within a master algorithm. In addition, its adaptation during the simulation by means of a flexible time step, can avoid unnecessary simulation steps with the advantages regarding simulation time, as discussed above. This adaptation can be applied for each FMU separately, depending on the nature of its exchange variables as well as their current status. In this sense, also stochastic variables, as applied in the occupancy behavior models introduced in simulation 1.7, need to be considered. Opposed to other variables, these might not allow for intelligent time step adaptation, since fixed patterns or gradient-based prediction is not available in this case.

In addition to these aspects, also the selection of domain boundaries and therefore the selection of exchange variables is of significance for the required time step. In the shown example, the air temperatures embody the thermal response of the building envelope to the provided heating rates. As a primary target, it is the enclosed air volume which provides the immediate response to the heating rate. Since the air volume features a low thermal mass, it shows a sensitive and immediate reaction. A scenario, which incorporates the air volume in the HVAC FMU and simply provides the current indoor air temperatures to the building envelope, which only incorporates the relatively high thermal mass of solid materials, would lead to a more stable solution and the possibility for longer time steps. Hence, by limiting variable exchange to slow reacting systems or variables, and computing sensitive model parts internally, master algorithms are able to apply longer time steps and increase simulation speed with no loss in result quality. In terms of the developed methodology, however, the selection of exchange variables is still subject to a unique association to design domains and is therefore restricted. Furthermore, the capabilities of simulation tools for defining model boundaries, as discussed in section 5.2.1 need to be considered.

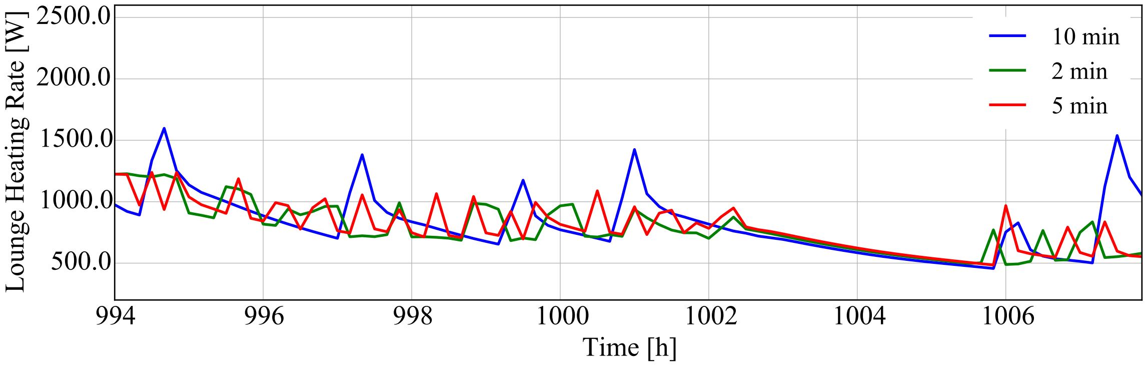


Figure 99. Comparison of air temperatures in the Lounge in simulation run 1.6 for simulation time steps of two, five and ten minutes.

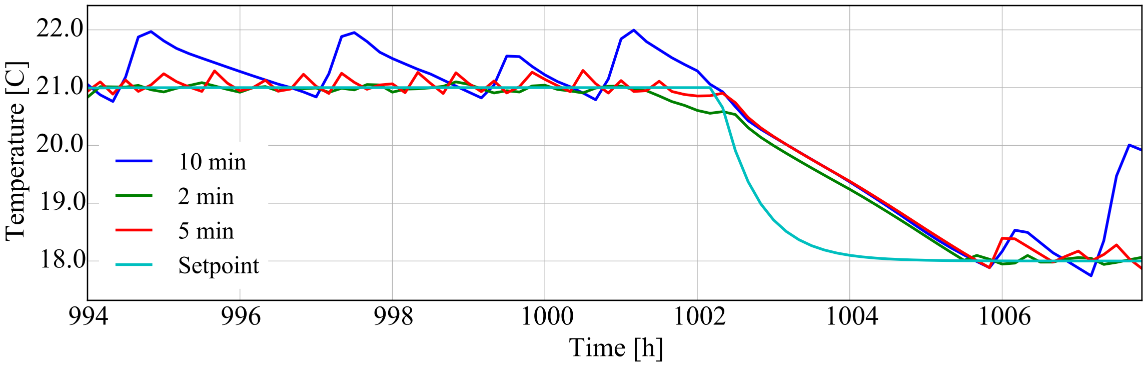


Figure 100. Comparison of heating rates in the Lounge in simulation run 1.6 for simulation time steps of two, five and ten minutes.

Chapter 6

Conclusion and Future Work

> “We have not succeeded in answering all our problems. The answers we have found only serve to raise a whole set of new questions. In some ways we feel we are as confused as ever, but we believe we are confused on a higher level and about more important things.” [62]

A research process is characterized by the continuous investigation of known and unknown matter in order to derive and develop new insights and extend the human capability of understanding this matter. Following the remarks from Earl C. Kelley about this enduring activity, the following chapter concludes the most important aspects and findings of this work. Some of which may not even have been a primarily targeted achievement but may still help to reduce confusion in related topics. Finally, ideas for future work are presented that build on the achieved knowledge level and which might in turn lead to a next knowledge level of greater importance themselves.

6.1 Conclusion

The presented work shows the development of a scalable building simulation based on modular components. The principles of functional layout, mathematical models and standardized mathematical models modularity are incorporated to meet the characteristics of a modular simulation. Correspondingly, the methodology enables the decomposition of a BPS into single components, the re-use of these components beyond project level, extendibility with new components and a modification of a simulation’s composition. These characteristics are utilized to incorporate goals for BPS tools and procedures from the literature within the methodology. As such, the procedure provides the required openness for several stakeholders to become involved in the simulation through the contribution of own modules. This allows for consideration of tool preferences and domain responsibilities within the design process. Furthermore, the demanded scalability of the simulation is considered through involving single- and multi-zone as well as zonal airflow representations of a building. This enables the adaptation to an evolving building data space during the course of a design process and allows for computing different KPIs, which vary in their relevance along this time scale. The methodology allows for integrating modules from several domains based on dynamic model formulations. This enables a holistic view of a building through consideration of inter-dependencies between domains. Besides dynamic models, models based on stochastic or empirical data can be integrated.

The concept builds on the newly developed FMI standard for model exchange and co-simulation and establishes a methodology for its application in the building sector. As such, it exceeds the examples from the literature for single use cases [93, 100, 103] with a general concept, which is able to benefit from the enabled model sharing and co-operation functionality on a broader scale and incorporates the mentioned goals for future BPS tools.
With respect to the building sector, the development closes the gap between a range of ongoing activities in the field of modular simulation. Figure 22 illustrates this technology chain. The FMI as the enabling technology for standardized tool co-operation is the basis of this tool set. The developed methodology is able to create a scalable BPS consisting of a selection of FMUs, which originate from heterogeneous sources. With the SSP format, a data scheme is currently under development, which allows for standardized description of such co-simulation setups. This description can be used to transfer the topology of a modular simulation to a co-simulation platform that is capable of solving the resulting simulation with tailored algorithms and increased computational resources. While the completion of this chain through this thesis provides considerable potential for next generation BPS technologies, the basic principles of the methodology can also be applied to other fields.
For the implementation of this technology chain, chapter 5 provides a summary of findings from this thesis. This concerns the introduction of additional cardinality options in FMU co-simulation. The option for a many-to-one connection, as applied in Modelica through the concept of physical connectors, is identified as a valuable addition to the standard. Furthermore, recommendations regarding the improvement of simulation time are provided. These contain the individualized treatment of FMUs depending on computational effort and a flexible adaptation of the simulation time step depending on type and status of variables. Additionally, a further aspect within the tool chain is noted at this stage. Within this thesis, unit conversion of exchange variables was realized in the coupling algorithm. In order to integrate this step within the tool chain, two options exist. First, information about a conversion factor and type, such as addition or multiplication is forwarded to a simulation platform. This requires a placeholder within the SSP format and the functionality of simulation platforms to process this information. Second, information about units is provided in the SSP format. In this case, the simulation platform must be able to identify the conversion process from information in the SSP file and execute the conversion based on an inherited database of conversion factors and processes.

A crucial feature of the developed methodology is the introduction of SWT into the field of simulation. This step embodies the required development regarding recent innovations in information technology, as demanded in Mazzarella and Pasini [74], to realize an open, modular simulation. Through OWL a meta-description of FMUs and their variables in the form of triples with respect to the overarching ontology *FMUont* is realized. The formal description logic enables a reasoner to infer connections between FMUs and leads to an automated derivation of the simulation topology. This overcomes the issue of manually finding these connections, as remarked in the literature [16, 107, 120]. The description is realized for the BPS context and allows for formally specifying information services and requests an FMU demands and provides within a simulation. Corresponding to the paradigm of SOA, FMUs therefore act as flexible entities in a self-adapting network, which instantly matches provided and requested information through updating its topology. The methodology enables this functionality across several domains and LODs, as mentioned above. The introduction of SWT into simulation poses a novelty and leads to vast opportunities for extending the possibilities of simulation due to the increasing propagation of SWT in the www. Besides the connection to existing or new ontologies, the possibility to directly address content in the www generates several chances for gaining functionality in simulation and expand its application. Section 6.2 addresses some of these opportunities.

The implementation of scalability within the procedure leads to the definition of information layers which must be provided for the individual LODs. In order to realize a multi-zone simulation, the addition of project specific information in the form of a digital building representation is required. This link is realized through common building data schemes, as demonstrated in chapter 4. Their conversion to OWL enables the connection of variables from simulation modules to objects within a building data model. The independence of the methodology from a specific data model is shown through deploying IFC and SimModel in two separate case studies. It is shown that despite the absence of, e.g. the concept of thermal zones in IFC, the methodology can still be applied. This is due to the required concepts for realizing the automated connection of modules being inherited in the overarching ontology *FMUont*. Objects from SimModel and IFC are merely related to these concepts and therefore receive the required context information independently from their internal definitions. This procedure is opposed to traditional BIMtoSIM approaches, which deploy the processing of a building data model or parts of it, in order to generate a simulation model. Welle et al. [127] show an example in this regard. Instead, the developed methodology establishes a persistent connection between variables of a simulation and objects in a digital building representation. This extends the possibilities for post-processing and the continuation of simulation models during the operation phase of a building. With this complementing functionality to BIM, the developed methodology poses a promising contribution to the digitization process in the building sector.

6.2 Future Work

The following section summarizes necessary research and ideas for future work based on the results from this thesis. The order prioritizes basic research and ascends with thoughts and visions beyond that, which further pursue the digitization of the building sector.

6.2.1 Simulation Time Performance

In order to increase the user-friendliness and the feasibility of optimization algorithms, the performance of the modular simulation regarding the required time must be enhanced. Potential to reach improvements is especially seen in simulation platforms. Optimized solvers, intelligent coupling algorithms and especially the possibility to access a nearly unlimited resource of computational power through clusters and cloud computing are promising opportunities to decrease simulation time. Despite the existence of several platforms, extensive tests involving a high number of FMUs have not been reported yet. It is hoped that the launch of the SSP format can trigger a series of experiments, which provide comparisons and ultimately lead to new innovations in this area. The references and findings from section 5.2.2 can provide a basis for these developments.

6.2.2 Extension with Domains and LODs

A manifest addition to the procedure is the extension with new domains and LODs. As an example, an electric domain can provide further insights into the interaction of Photovoltaics (PV) with batteries and on-site power consumers like electric cars. Furthermore, the introduction of other LODs, such as the planning of entire quarters, yield in increased functionality. This allows to investigate, for instance micro-grid concepts or the interactions in local heat supply systems. Hence, the application of the simulation can be extended to grid load computation or the planning of self-sustaining quarters. Additional benefits on this path can arise from the connection to common data schemes of Geographical Information Systems (GIS).

6.2.3 Intra-Domain Decomposition

As elaborated in section 5.1.4, the chosen inter-domain decomposition limits the functionality of the simulation regarding the exchange of modules with smaller granularity. As an example, the exchange of an air-water heat pump with a brine-water heat pump model requires the replacement of the entire HVAC FMU regardless of the fact that the other components remain identical. In order to enable module exchange on a smaller scale, an intra-domain decomposition must be realized. This requires an extension of the developed ontology with intra-domain-specific structural information. As an example, for the HVAC domain possible flow paths and resulting device connections can be defined. The challenge is to accommodate the immense variety of HVAC configurations in this description. An opportunity persists in the pre-definition of certain configuration types, which are selected corresponding to the FMUs involved in a simulation. Hence, a selection of FMUs that involves a primary energy generator, a thermal storage facility and a number of heat transfer units, always features the same flow path configuration. Nevertheless, in order to maintain compatibility of simulation modules, clear boundaries between them must be defined. As a remedy to realize this task, BIM formats can provide a standardized guideline, as discussed in the following.

6.2.4 BIMtoSIM

BIM formats, such as IFC or SimModel, subdivide a building and its components into numerous objects which are connected in a descriptive scheme. For the realization of a BIMtoSIM concept for modular simulation, an adaptation of the module scope to the defined objects in a BIM format is necessary. On the one hand, this considerably increases the number of modules and therefore the number of exchange variables in a simulation, on the other hand this results in further possibilities for advanced application.

Concerning the discussed intra-domain decomposition, the defined objects in BIM formats can pose the required module scope definition. Regarding the mentioned BIMtoSIM concept for modular simulation, this decomposition allows for a unique derivation of simulation modules from a populated BIM scheme. BIM objects can be interpreted as smallest components which can be simulated individually. Hence, they feature a reference to an FMU incorporating the corresponding simulation model of this component. The screening of a BIM scheme can then yield a collection of FMUs, which, as a collective, represent the simulation model of the digital building. The adaptation of modules to BIM objects thereby allows for the definition of generic FMUs which can be re-used beyond projects, even for the building envelope domain. For example, an FMU incorporating a multi-layered wall model or a window can be parameterized with area, material properties, orientation etc. Additionally, the association of FMUs to the specific BIM objects can immediately be realized during the screening. In conjunction with the, in this thesis determined information to realize a modular simulation, the connections between modules can be inferred following the developed process.

Furthermore, this enables new possibilities for the integration of Computer-Aided-Design (CAD) and simulation. As an example, a CAD design tool can be based on IFC object definitions. When dragging a geometrical representation of this object, e.g. a radiator, into the design space, also an FMU incorporating a physical model of this radiator can be added to the collection of simulation modules.

6.2.5 Product Data Integration

The semantic description of variables in this thesis is limited to input and output variables of simulation modules in order to derive connections between them. The description of parameters of these modules was out of scope in the current work. However, as applied manually for the parameters listed in Table 3, the integration of product data into the simulation can lead to an individualization of generic simulation modules to represent products available on the market. The connection can be established via linked data similar to the developed approach. URIs help to access the product specifications from the web, e.g. from manufacturer websites. A matching of parameters can then be achieved through a corresponding semantic description.

Efforts to establish product data catalogs which support the principle of linked data are still rare. An example is SemCat developed by Gudnason and Pauwels in [40]. The tool allows for generating a semantic description from manufacturer data for building products held in Excel in order to provide consistent semantics across multidisciplinary processes. Such a tool can be the basis for establishing a distributed knowledge base for product parameters which can be accessed in the www. Their standardized description ultimately allows for relating required simulation parameters to the corresponding product specifications. Additionally, a reversed inference process can yield in a range of products that fulfill the simulation parameters, which yield in the desired performance of a building.

6.2.6 Internet of Simulation Things

Similar to above, the following vision targets a direct integration of characteristics from commercially available products into BPS. However, in contrary to the adoption of product specific parameters in the simulation, an integration of product specific simulation modules is the main goal. The openness of the FMI and OWL guarantees each stakeholder to contribute own simulation modules to the simulation. Hence, besides standard documents, such as a manual and a product data sheet, also a calibrated, highly individualized simulation model in the form of an FMU can be provided online. This ensures decentralized maintenance of simulation modules and can significantly increase the quality of simulation results. Hence, a "web-based repository", as demanded in [74], of simulation modules can be realized. This concept is aligned with the at the beginning described vision of future design processes formulated by Wetter in 2011 [128]. He outlines a dashboard accessible by an integrated design team, allowing for seamless testing of different design variants with immediate performance feedback enabled through components from an electronic product catalog.

In a scenario in which each planner can access a number of ready-to-simulate product models provided by manufacturers or other stakeholders, an "Internet of Simulation Things", as first mentioned in [77], is pictured. An intelligence capable of gathering models online and collocating them in a simulation can be realized based on the procedure and technologies shown in this work. A pre-condition of this concept is a further granularization of the modular simulation. This guarantees access for a variety of stakeholders, also smaller manufacturers specialized on single components.

6.2.7 Intelligent Building Data Space

The establishment of a persistent connection between objects in a BIM scheme and simulation variables extends the possibilities of simulation post-processing. Therefore, simulation results can be evaluated based on their associated BIM objects. Semantic consequences for the evaluation of a desired KPI can be considered in the post-processing step. As an example, an identical surface temperature of two walls can yield a different evaluation regarding mold growth due to different substrates provided by the inner layer of the wall. The remaining connection to a BIM scheme enables this intelligent post-processing functionality and furthermore allows to report the corresponding result directly in the digital building model.

Chapter 7

Summary

This thesis is motivated by the increasingly important role of buildings in human society. As a space where humans spend most of their time, requirements on buildings are rising continuously while including a growing number of parameters to improve our daily life. Among others, these comprise thermal comfort, air quality, availability of daylight, noise levels etc. Besides criteria focusing on the occupant, also global aspects, such as costs or environmental impact need to be considered. BPS serves as a remedy to a variety of stakeholders in order to realize these goals through development and evaluation of new and innovative building solutions.

Recent literature identifies an ideal simulation process to be *scalable* throughout the course of a design project by enabling continuous performance feedback in different project stages for different KPIs. This feedback should be based on an *integrated* simulation, which considers the *dynamic* nature of a building's subsystems and their interactions. Furthermore, it is desired to enable *multiple users* to contribute, execute and evaluate a simulation in order to receive immediate performance feedback while considering domain-specific particularities in the overall simulation. Several authors [17, 48, 128] demand an *open* simulation environment to realize the integration of numerous users on a single platform and foster its development. Citherlet [16] and Hensen [48] identified four simulation procedures, which have been developed over time: the stand-alone approach, data model interoperation, process model interoperation and process model co-operation. With each approach having its individual motivation and historical foundation within the development process, differences in covering the above-mentioned aspects can be detected. Several authors [16, 17, 48, 74, 120] determine the process model co-operation to be a promising approach to correspond to the requirements. A recent development in this field is the FMI, a standard for tool-independent model exchange and co-simulation of dynamic models exported as FMUs. Examples from the literature [93, 100, 103] illustrate the benefits of the standard in the field of BPS arising from the combination of specialized tools and therefore extended simulation functionality. With a tool chain consisting of simulation platforms and a standardized format (SSP), which allows for forwarding information about a co-simulation setup, a holistic methodology for the application of FMI in building simulation is yet missing. A decisive drawback of the decomposition of a simulation into single modules is the requirement to find connections between them in order to achieve the recombination. This issue has repeatedly been reported [16, 107, 120]. Based on developments in the www arising from a similar issue, which led to the semantic web revolution, the characteristics of the ontology description language OWL are discussed and determined to feature considerable potential in order to complement the development of a modular BPS based on FMI. This development is intended to close the gap in the mentioned tool chain and provide a methodology for a scalable BPS, which features the above-mentioned characteristics for an ideal simulation process.

In order to realize the modular simulation, the decomposition of a BPS is discussed. Opposed to an intra-domain approach, an inter-domain approach features clear boundaries between different fields ensuring the compatibility of modules within the simulation. Furthermore, classification of responsibilities of planners and stakeholders as well as the modeling capabilities of tools can be considered in the generation of simulation modules. Hence, an inter-domain decomposition is determined to be the foundation of the developed methodology. The considered domains are: Building envelope, environment, HVAC, people, building equipment and building automation.

The scalable character of the simulation is inherited in the building envelope domain with single-zone, multi-zone and zonal airflow models of building parts, representing the different LODs. The selected models allow for assessing different KPIs, e.g. annual energy usage or thermal indoor climate of single rooms, and find application at different stages of the design process. To automatically combine simulation models of each domain, which represent different functionality and vary in their LOD corresponding to the previous definition, modules are perceived as service providing and service requesting entities corresponding to the principles of SOA. When confronted with an accumulation of simulation modules, required and provided services, i.e. exchange variables, need to be matched. In order to automatically derive these connections and to maintain the integrity of the simulation after module exchanges, the advantages of OWL are exploited. An overarching ontology, termed *FMUont*, serves as the basis of a knowledge framework. The standard content of FMUs is extended with an additional *owl* file, which contains an FMU-specific description. The descriptions of each FMU are associated with concepts from the overarching ontology *FMUont*. This sets each FMU into the same context and allows for identifying the role of an FMU within a BPS and more specifically its provided and requested services to the simulation. The formally specified information model allows a reasoner to automatically infer the connections between variables of selected FMUs based on defined rule sets. Through the combination of a knowledge representation in the form of an ontology and a reasoner capable of inferring new information from this ontology, the method is by definition a knowledge-based approach [106], a discipline from the field of Artificial Intelligence (AI).

The applied rule sets to compose a modular simulation vary, depending on the required information level due to the applied LOD. For a single-zone simulation, information about base properties and semantic meaning of a variable in the context of BPS is determined to be sufficient information. Regarding a multi-zone model and the parallel assessment of single rooms through a zonal airflow model, project-specific information must be included. This is realized through the association of variables to objects within a digital representation of

a building enabled through BIM. Variables do therefore possess information about the object they are associated with in the overall building. Spatial associations of these objects to simulated rooms and thermal zones are derived through applying a shortest-path algorithm, the Dijkstra-algorithm, to the internal structure of the populated BIM format. The added project-specific information is ultimately considered in the rule set for the reasoning process and completes the formal description of variables in the knowledge base.
The procedure is implemented in the Python programming language. A GUI called the *FMUOntologyXtender* facilitates the description of FMU variables corresponding to the overarching ontology *FMUont*, and extends the FMU with the resulting *owl* file. The Python algorithm aggregates these descriptions for selected FMUs and executes the methodology with the co-simulation being orchestrated in a loose-coupling algorithm.

Two case studies illustrate the feasibility and the technical merit of the methodology. The first example features one of the Twin Houses, two identical buildings located at the test site of the Fraunhofer IBP in Holzkirchen, Germany. The study gradually refines the LOD from a single-zone representation of the entire building to a zonal airflow model of a single room while maintaining a holistic view through consideration of other domains, such as HVAC and people, in their entirety. For the continuously increased LOD, an information enrichment is required. This corresponds to the growing information about a building during the design process. The different LODs furthermore serve to assess different aspects, i.e. KPIs. The single-zone model provides an estimation of the thermal load for the building and allows for quickly evaluating HVAC systems regarding their capability to provide this load. At this stage, the modular simulation can be assembled for basic information about variables and their associated role in the field of BPS. In order to advance to a multi-zone model of the building, a description of the building in the SimModel format is associated with the simulation modules. This step allows for detailed investigation of the response behavior of the building and its interacting domains. As such, the HVAC module is exchanged several times in order to test a different system. Finally, the integration of a product-specific model based on a heat pump and a thermal storage unit is realized. Due to the calibration of the model components to commercially available products and the resulting high-quality prediction, it is used for improving the tank design temperature with regards to annual power consumption. In order to further assess the indoor climate in a single room of the building, a zonal airflow model of this space is added to the simulation. The model allows for computing the air temperature distribution and hence is able to evaluate local differences within a single room. In the case study it is applied to improve the positioning of radiators in order to ensure thermal comfort in a room. This assessment can be realized under parallel consideration of the interactions with the whole building as well as the entirety of other domains, such as the HVAC system. A second example demonstrates the independence of the methodology from specific formats established in BIM. The example features a building from the IFC example repository and illustrates the compatibility with the IFC data model. The independence is due to the definition of required concepts in the knowledge base. Objects from a BIM scheme are related to these concepts and therefore receive the required context information to realize the modular simulation. Hence, e.g. the absence of thermal zone definitions in IFC does not pose an obstacle.

The methodology is evaluated with high potential to meet the requirements of next generation BPS tools. The integration of different LODs allows for a continuous adaptation to a growing information pool during the design process. Furthermore, assessment criteria can be adapted, i.e. from whole building energy analysis to the investigation of indoor climate in a single room. The openness and extendibility guaranteed through the involved technologies, i.e. FMI and SWT, enable multiple stakeholders to contribute models to the simulation. Hence, a holistic view of a building relying on fully dynamic simulation modules from different domains can be realized. In addition, models based on stochastic or empirical data can be integrated.
The flexibility to accommodate these characteristics arises from the modular nature of the simulation, which is based on system decomposition, model re-use, extendibility and system composition modification. Current issues are identified in the restriction of the FMI standard to a one-to-one or one-to-many cardinality of connections. A many-to-one connection, as applied in physical connectors, can lead to increased flexibility and functionality for the modular simulation. Future work lies in the extension of the overarching data model in order to allow for an intra-domain decomposition. This enables the exchange of single components of a domain in order to test different, domain internal system compositions. The successful combination of simulation and SWT further leads to the vision of integrating online available product data from OEMs in the simulation. A semantic description of parameters in generic simulation modules can allow for direct integration of corresponding product data in order to represent the product specifics in the simulation model. Beyond this, the concept of an "Internet of Simulation Things" is sketched, which provides online access to calibrated FMUs of commercially available products. An intelligence capable of retrieving and aggregating a selection of these modules, in order to form a holistic simulation, can be based on the procedure and technologies presented in this thesis.

Bibliography

[1] Aasem E. *Practical simulation of buildings and air-conditioning systems in the transient domain.* PhD thesis, Mechanical Engineering Dept., Strathclyde University, Glasgow, UK, 1993.

[2] Asadi E., Silva M. G., Antunes C. H. and Dias L. A multi-objective optimization model for building retrofit strategies using TRNSYS simulations, GenOpt and MATLAB. *Building and Environment*, 56:370 – 378, 2012.

[3] Attia S. Building performance simulation tools: Selection criteria and user survey. `http://hdl.handle.net/2268/168350`, 2010. Accessed: 2017-03-01.

[4] Attia S., Hensen J. L., Beltrán L. and De Herde A. Selection criteria for building performance simulation tools: contrasting architects' and engineers' needs. *Journal of Building Performance Simulation*, 5(3):155–169, 2012.

[5] Augenbroe G. Trends in building simulation. *Building and Environment*, 37(8–9):891 – 902, 2002.

[6] Augenbroe G., De Wilde P., Moon H. J., Malkawi A., Brahme R. and Choudhary R. The design analysis integration (DAI) initiative. In *Proceedings of the 8th IBPSA Conference, August*, pages 11–14, 2003.

[7] Bazjanac V. IFC BIM-based methodology for semi-automated building energy performance simulation. `http://escholarship.org/uc/item/0m8238pj`, 2008. Accessed: 2017-03-01.

[8] Beausoleil-Morrison I. *The adaptive coupling of heat and air flow modelling within dynamic whole-building simulation.* Dissertation, University of Strathclyde, Strathclyde, 2000.

[9] Beausoleil-Morrison I., Macdonald F., Kummert M., McDowell T., Jost R. and Ferguson A. The design of an ESP-r and TRNSYS co-simulator. In *Proc. Building Simulation*, volume 12, pages 2333–2340, 2011.

[10] Bleil De Souza C. and Knight I. Thermal performance simulation from an architectural design viewpoint. In *Sixth International IBPSA Conference, China*, pages 87–94, 2007.

[11] Blochwitz T., Otter M., Arnold M., Bausch C., Elmqvist H., Junghanns A., Mauß J., Monteiro M., Neidhold T., Neumerkel D., Olsson H., Peetz J.-V., Wolf S. and Clauß C. The Functional Mockup Interface for tool independent exchange of simulation models.

In *Proceedings of the 8th International Modelica Conference, March 20th-22nd, Technical University, Dresden, Germany*, number 63, pages 105–114. Linköping University Electronic Press, Linköpings Universitet, 2011.

[12] Blochwitz T., Otter M., Åkesson J., Arnold M., Clauss C., Elmqvist H., Friedrich M., Junghanns A., Mauss J., Neumerkel D., Olsson H. and Viel A. Functional Mockup Interface 2.0: The standard for tool independent exchange of simulation models. In *Proceedings of the 9th International Modelica Conference*, pages 173–184. The Modelica Association, 2012.

[13] Bolineni S., Stratbücker S. and van Treeck C. Development of reduced order flow responsive models of convective heat transfer for human body segments. *Building Simulation*, 8(4):393–404, 2015.

[14] Borrmann A., König M., Koch C. and Beetz J. *Building Information Modeling: Technologische Grundlagen und industrielle Praxis*. Springer-Verlag, 2015.

[15] buildingSMART. International home of open BIM. `http://www.buildingsmart-tech.org/specifications/ifc-overview`, 2017. Accessed: 2017-03-01.

[16] Citherlet S. *Towards the holistic assessment of building performance based on an integrated simulation approach*. PhD thesis, Swiss Federal Institute of Technology, Lausanne, Switzerland, 2001.

[17] Clarke J. A. and Hensen J. L. M. Integrated building performance simulation: Progress, prospects and requirements. *Building and Environment*, 91:294–306, 2015.

[18] Clarke J. Prospects for truly integrated building performance simulation. In *Proceedings of Building Simulation*, volume 99, pages 1147–1154, 1999.

[19] Clarke J. Integrated building performance simulation. In *Proceedings of the 4th International Indoor Air Quality Ventilation and Energy Conservation in Buildings Conference (IAQVEC 2001)*, pages 1395–1404, Hong Kong, April 2001. City University Hong Kong.

[20] Clarke J. A. *Energy simulation in building design*. Routledge, 2001.

[21] Crawley D. B., Hand J. W., Kummert M. and Griffith B. T. Contrasting the capabilities of building energy performance simulation programs. *Building and Environment*, 43(4):661 – 673, 2008. Part Special: Building Performance Simulation.

[22] Crawley D. B., Pedersen C. O., Lawrie L. K. and Winkelmann F. C. EnergyPlus: Energy simulation program. *ASHRAE Journal*, 42:49–56, 2000.

[23] Cremona F., Lohstroh M., Tripakis S., Brooks C. and Lee E. A. FIDE - an FMI integrated development environment. In *Symposium on Applied Computing*, April 2016.

[24] Dassault Systems. Dymola. `http://www.3ds.com/products\hbox{-}services/catia/products/dymola`, 2015. Dassault Systemes AB, Lund, Sweden, Accessed: 31-12-2016.

[25] De Sturler E., Hoeflinger J., Kale L. and Bhandarkar M. A new approach to software integration frameworks for multi-physics simulation codes. In *The Architecture of Scientific Software*, pages 87–104. Springer, 2001.

[26] Wilde P. J. C. J. and Voorden M. *Computational support for the selection of energy saving building components.* DUP Science, 2004.

[27] Deutsches Institut für Normung. *DIN V 18599: Energetische Bewertung von Gebäuden; Berechnung des Nutz-, End- und Primärenergiebedarfs für Heizung, Kühlung, Lüftung, Trinkwarmwasser und Beleuchtung; Vornormenreihe.* Bauwesen. Beuth, Berlin, 2013.

[28] Dibowski H., Holub O. and Rojicek J. Knowledge-based Fault Propagation in Building Automation Systems. In *SIMS*, Riga, Latvia, 2016.

[29] Dibowski H., Ploennigs J. and Kabitzsch K. Automated design of building automation systems. *IEEE Transactions on Industrial Electronics*, 57(11):3606–3613, 2010.

[30] Dijkstra E. W. A note on two problems in connexion with graphs. *Numer. Math.*, 1(1):269–271, December 1959.

[31] Djunaedy E. *External coupling between building energy simulation and computational fluid dynamics.* PhD thesis, Technische Universiteit Eindhoven, the Netherlands, 2005.

[32] Dorer V. and Weber A. Air, contaminant and heat transport models: integration and application. *Energy and Buildings*, 30(1):97 – 104, 1999.

[33] Eastman C. M., Eastman C., Teicholz P., Sacks R. and Liston K. *BIM handbook: A guide to building information modeling for owners, managers, designers, engineers and contractors.* John Wiley & Sons, 2011.

[34] Energy Systems Research Unit, University of Strathclyde. ESP-r. http://www.esru.strath.ac.uk/Programs/ESP-r.htm, 2017. Accessed: 2017-03-01.

[35] Fraunhofer Institute for Building Physics (IBP). WUFI® Plus. `https://wufi.de/de/software/wufi-plus/`, 2017. Accessed: 2017-04-01.

[36] Fuller D. and McNeil A. Radiance - a validated lighting simulation tool. `https://www.radiance-online.org/`, 2016. Accessed: 2017-04-01.

[37] Galtier V., Vialle S., Dad C., Tavella J.-P., Lam-Yee-Mui J.-P. and Plessis G. FMI-based distributed multi-simulation with DACCOSIM. In *DEVS*, pages 39–46, Alexandria, USA, 2015.

[38] Gao N. and Niu J. CFD study on micro-environment around human body and personalized ventilation. *Building and Environment*, 39(7):795 – 805, 2004.

[39] Green Building XML (gbXML) Schema, Inc. gbXML. `http://www.gbxml.org/`, 2017. Accessed: 2017-03-01.

[40] Gudnason G. and Pauwels P. SemCat: Publishing and Accessing Building Product Information as Linked Data. In *ECPPM*, pages 659–666, Limassol, Cyprus, 2016.

[41] Guglielmetti R., Macumber D. and Long N. OpenStudio: an open source integrated analysis platform. In *Proceedings of the 12th conference of international building performance simulation association*, 2011.

[42] Guglielmetti R., Macumber D. and Long N. Openstudio: an open source integrated analysis platform. In *Proceedings of the 12th conference of International Building Performance Simulation Association; Sidney, Australia*. International Building Performance Simulation Association, 2011.

[43] Haghighat F., Li Y. and Megri A. C. Development and validation of a zonal model — POMA. *Building and Environment*, 36(9):1039 – 1047, 2001.

[44] Haldi F. A probabilistic model to predict building occupants' diversity towards their interactions with the building envelope. In *Proceedings of the international IBPSA conference, Chambery, France*, page 147, 2013.

[45] Ham Y. and Golparvar-Fard M. Mapping actual thermal properties to building elements in gbXML-based BIM for reliable building energy performance modeling. *Automation in Construction*, 49, Part B:214 – 224, 2015. 30th ISARC Special Issue.

[46] Han J., Jeong Y.-K. and Lee I. A rule-based ontology reasoning system for context-aware building energy management. In *CIT/IUCC/DASC/PICOM*, pages 2134–2142, 2015.

[47] Haves P. and Xu P. The building controls virtual test bed a simulation environment for developing and testing control algorithms, strategies and systems. In *Proceedings of the 10th IBPSA Conference*, pages 1440–1446, 2007.

[48] Hensen J. Towards more effective use of building performance simulation in design. In *Proc. 7th International Conference on Design & Decision Support Systems in Architecture and Urban Planning*, pages 2–5, 2004.

[49] Hensen J. A comparison of coupled and decoupled solutions for temperature and air flow in a building. *ASHRAE transactions*, 105:962, 1999.

[50] Hensen L. Introduction to building performance simulation. *Building Performance Simulation for Design and Operation*, pages 1–14, 2011.

[51] Hitzler P., Krötzsch M. and Rudolph S. *Foundations of Semantic Web technologies.* CRC Press, Boca Raton, USA, 2010.

[52] Hitzler P., Krötzsch M., Parsia B., Patel-Schneider P. F. and Rudolph S. OWL 2 web ontology language primer. *W3C recommendation*, 27(1):123, 2009.

[53] Horrocks I., Patel-Schneider P. F., Boley H., Tabet S., Grosof B., Dean M. et al. SWRL: A semantic web rule language combining OWL and RuleML. *W3C Member submission*, 21:79, 2004.

[54] Huang J., Winkelmann F. C., Buhl F., Pedersen C. O., Fisher D., Liesen R., Taylor R., Strand R., Crawley D. B. and Lawrie L. K. Linking the COMIS multi-zone airflow model with the EnergyPlus building energy simulation program. In *Proc. of the 6-th IBPSA Conference*, volume 2, pages 1065–1070, 1999.

[55] Huber-Vatres J. *Integrative Ansätze in der energetischen Gebäudesimulation auf den Modellskalen Einzelraum, Gebäude und Stadtquartier.* PhD thesis, Technische Universität Berlin, Institut für Energietechnik, Fachgebiet Maschinen- und Energieanlagentechnik, 2015.

[56] IBPSA. Modelica IBPSA Library. `https://github.com/ibpsa/modelica-ibpsa`, 2017. Accessed: 2017-07-02.

[57] Inard C., Bouia H. and Dalicieux P. Prediction of air temperature distribution in buildings with a zonal model. *Energy and Buildings*, 24(2):125 – 132, 1996.

[58] ITEA Office Association. Modelisar - from system modeling to running on the vehicle. `https://itea3.org/project/modelisar.html`, 2017. Accessed: 2017-04-01.

[59] Janak M. Coupling building energy and lighting simulation. In *Proc. Building Simulation*, volume 2, pages 313–319, 1997.

[60] Karlsruher Institut für Technologie. KIT IFC Examples. `http://www.ifcwiki.org/index.php?title=KIT_IFC_Examples`, 2017. Accessed: 2017-08-02.

[61] Karlsruher Institute of Technology. FZKViewer. `https://www.iai.kit.edu/english/1309.php`, 2017. Accessed: 2017-08-02.

[62] Kelley E. C. *The Workshop Way of Learning*, volume 1. New York: Harper & Brothers, 1951.

[63] Kofler M., Reinisch C. and Kastner W. A semantic representation of energy-related information in future smart homes. *Energy and Buildings*, 47:169–179, 2012.

[64] Köhler J., Heinkel H.-M., Mai P., Krasser J., Deppe M. and Nagasawa M. Modelica-Association-Project 'System Structure and Parameterization' - early insights. In *Japanese Modelica*, pages 35–42, Tokyo, Japan, 2016.

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

[66] Lamy J.-B. Owlready: Ontology-oriented programming in Python with automatic classification and high level constructs for biomedical ontologies. *Artificial Intelligence in Medicine*, 2017.

[67] Lindner A. J. M., Park S. and Mitterhofer M. Determination of requirements on occupant behavior models for the use in building performance simulations. *Building Simulation*, Aug 2017.

[68] Macdonald F., Jost R., Beausoleil-Morrison I., Kummert M., McDowell T. and Ferguson A. A demonstration of the runtime coupling between ESPr and TRNSYS. *IBPSA-USA Journal*, 5(1):521–528, 2012.

[69] Mahdavi A. Computational decision support and the building delivery process: a necessary dialogue. *Automation in Construction*, 7(2):205 – 211, 1998.

[70] Mahdavi A. People in building performance simulation. *Building Performance Simulation for Design and Operation*, pages 218–234, 2011.

[71] Mahdavi A., Feurer S., Redlein A. and Suter G. An inquiry into the building performance simulation tools usage by architects in Austria. In *Proc. 8th International IBPSA Conferenc, Eindhoven, Netherlands*, pages 777–784, 2003.

[72] Mahdavi A., Mathew P., Kumar S. and Wong N.-H. Bi-directional computational design support in the semper environment. *Automation in Construction*, 6(4):353 – 373, 1997.

[73] Masoso O. and Grobler L. The dark side of occupants' behaviour on building energy use. *Energy and Buildings*, 42(2):173 – 177, 2010.

[74] Mazzarella L. and Pasini M. Building energy simulation and object-oriented modelling: review and reflections upon achieved results and further developments. In *Building Simulation conference, Glasgow*, pages 638–645, 2009.

[75] McDowall R. *Fundamentals of HVAC systems.* Academic Press, 2006.

[76] Mitterhofer M., Norrefeldt V. and Stratbücker S. Advanced modeling and simulation methods for assessment and optimization of indoor climate. In *9th International Conference on Indoor Air Quality Ventilation and Energy Conservation In Buildings*, Seoul, South Korea, 2016.

[77] Mitterhofer M., Schneider G., Stratbücker S. and Steiger S. Semantics For Assembling Modular Network Topologies In FMI-Based Building Performance Simulation. In *Proceedings of the 15th IBPS Conference for Building Simulation*, San Francisco, USA, 2017.

[78] Mitterhofer M. and Stratbücker S. Automated Collocation of Simulation Modules in FMI-Based Building Energy Co-Simulation. In *BauSIM/CESBP*, pages 559–564, Dresden, Germany, 2016.

[79] Mitterhofer M., Schneider G. F., Stratbücker S. and Sedlbauer K. An FMI-enabled methodology for modular building performance simulation based on semantic web technologies. *Building and Environment*, 125(Supplement C):49 – 59, 2017.

[80] Modelica Association. FMI - Functional Mock-up Interface. `https://fmi-standard.org/`, 2017. Accessed: 2017-04-01.

[81] Modelica Association. FMI support in tools. `https://www.fmi-standard.org/tools`, 2017. Accessed: 2017-02-05.

[82] Modelica Association. Modelica. `https://www.modelica.org/`, 2017. Accessed: 2017-03-01.

[83] Modelon AB. PyFMI. `http://www.jmodelica.org/page/4924`, 2017. Accessed: 2017-01-16.

[84] Morbitzer C. A. *Towards the integration of simulation into the building design process.* PhD thesis, University of Strathclyde, 2003.

[85] Motik B., Cuenca Grau B. and Sattler U. Structured objects in owl: Representation and reasoning. In *Proceedings of the 17th International Conference on World Wide Web*, WWW '08, pages 555–564, 2008.

[86] Mourshed M., Kelliher D. and Keane M. Integrating building energy simulation in the design process. *IBPSA News*, 13(1):21–26, 2003.

[87] Musy M., Wurtz E., Winkelmann F. and Allard F. Generation of a zonal model to simulate natural convection in a room with a radiative/convective heater. *Building and Environment*, 36(5):589 – 596, 2001.

[88] Negendahl K. Building performance simulation in the early design stage: An introduction to integrated dynamic models. *Automation in Construction*, 54:39–53, 2015.

[89] Negendahl K. and Nielsen T. R. Building energy optimization in the early design stages: A simplified method. *Energy and Buildings*, 105:88–99, 2015.

[90] Nicolai A. MASTERSIM Co-Simulation Master. `http://bauklimatik-dresden.de/mastersim/index.php?aLa=en`, 2017. Accessed: 2017-01-16.

[91] Norrefeldt V. *Velocity propagating zonal model: VEPZO, a locally refined airflow model for confined spaces to use in optimization applications*, volume Bd. 14 of *Forschungsergebnisse aus der Bauphysik.* Fraunhofer-Verl., Stuttgart, 2013.

[92] Norrefeldt V., Grün G. and Sedlbauer K. VEPZO – velocity propagating zonal model for the estimation of the airflow pattern and temperature distribution in a confined space. *Building and Environment*, 48:183 – 194, 2012.

[93] Nouidui T., Wetter M. and Zuo W. Functional Mock-up Unit for co-simulation import in EnergyPlus. *Journal of Building Performance Simulation*, 7(3):192–202, 2014.

[94] O'Donnell J., See R., Rose C., Maile T., Bazjanac V. and Haves P. SimModel: A domain data model for whole building energy simulation. In *SimBuild*, pages 382–389, Sydney, Australia, 2011.

[95] Papamichael K. M., LaPorta J., Chauvet H. L., Collins D., Trzcinski T., Thorpe J. A. and Selkowitz S. E. The Building Design Advisor. In *ACADIA 1996 Conference*, Tucson, AZ, 03/1996 1996.

[96] Pathak A., Norrefeldt V., Lemouedda A. and Grün G. The Modelica thermal model generation tool for automated creation of a coupled airflow; radiation model and wall model in Modelica. In *Proceedings of the 10 th International Modelica Conference; March 10-12; 2014; Lund; Sweden*, number 096, pages 115–124. Linköping University Electronic Press, 2014.

[97] Pauwels P., Corry E. and O'Donnell J. Representing SimModel in the web ontology language. In *Computing in Civil and Building Engineering*, pages 2271–2278, 2014.

[98] Pauwels P., Deursen D. V., Verstraeten R., Roo J. D., Meyer R. D., Walle R. V. and Campenhout J. V. A semantic rule checking environment for building performance checking. *Automation in Construction*, 20(5):506–518, 2011.

[99] Pauwels P. and Terkaj W. EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology. *Automation in Construction*, 63:100–133, 2016.

[100] Pazold M., Burhenne S., Radon J., Herkel S. and Antretter F. Integration of Modelica models into an existing simulation software using FMI for co-simulation. In Otter D., M.and Zimmer, editor, *Proceedings of the 9th International Modelica Conference*, Linköping Electronic Conference Proceedings, pages 949–954, 2012.

[101] Petersen S. *Simulation-based support for integrated design of new low-energy office buildings.* PhD thesis, Technical University of Denmark, 2011.

[102] Pieter Pauwels. IFC-to-RDF Converter. `http://smartlab1.elis.ugent.be:8889/IFC-repo/`, 2017. Accessed: 2017-07-02.

[103] Plessis G., Amouroux É. and Haradji Y. Coupling occupant behaviour with a building energy model - a FMI application. In *Proceedings of the 10th International Modelica Conference*, pages 321–326, 2014.

[104] Prívara S., Váňa Z., Gyalistras D., Cigler J., Sagerschnig C., Morari M. and Ferkl L. Modeling and identification of a large multi-zone office building. In *2011 IEEE International Conference on Control Applications (CCA)*, pages 55–60, 2011.

[105] Rijgersberg H., Assem M., Willems D., Wigham M., Broekstra J. and Top J. Ontology of units of Measure (OM). `http://www.wurvoc.org/vocabularies/om-1.8/`, 2016. Accessed: 2017-01-16.

[106] Russel S. and Norving P. *Artificial Intelligence: A modern approach.* Pearson Education Limited, 3rd edition, 2013.

[107] Sagerschnig C., Gyalistras D., Seerig A., Prívara S., Cigler J. and Vana Z. Co-simulation for building controller development: The case study of a modern office building. In *Proc. CISBAT*, pages 14–16, 2011.

[108] Santin O. G., Itard L. and Visscher H. The effect of occupancy and building characteristics on energy use for space and water heating in dutch residential stock. *Energy and Buildings*, 41(11):1223 – 1232, 2009.

[109] Schütze M., Riegel J. P. and Zimmermann G. Psigene: A pattern-based component generator for building simulation. *Theory and practice of object systems*, 5(2):83–95, 1999.

[110] Sirin E., Parsia B., Grau B. C., Kalyanpur A. and Katz Y. Pellet: A practical OWL-DL reasoner. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(2):51 – 53, 2007. Software Engineering and the Semantic Web.

[111] Spitler J. D. Building performance simulation: The now and the not yet. `http://www.tandfonline.com/doi/abs/10.1080/10789669.2006.10391194`, 2006.

[112] Stanford University. Protege - A free, open-source ontology editor and framework for building intelligent systems. `http://protege.stanford.edu/`, 2017. Accessed: 2016-08-02.

[113] Struck C. *Uncertainty propagation and sensitivity analysis techniques in building performance simulation to support conceptual building and system design.* University of Technology Eindhoven, Department of the Built Environment, 2012.

[114] Tan G. and Glicksman L. R. Application of integrating multi-zone model with CFD simulation to natural ventilation prediction. *Energy and Buildings*, 37(10):1049 – 1057, 2005.

[115] Tang D. Object technology in building environmental modelling. *Building and Environment*, 32(1):45 – 50, 1997.

[116] Temam R. *Navier-Stokes Equations*, volume 2. North-Holland Amsterdam, 1984.

[117] Thermal Energy System Specialists, LLC. TRNSYS. `http://www.trnsys.com/`, 2017. Accessed: 2017-03-01.

[118] Tiller M. FMQ Platform. `http://www.xogeny.com/products`, 2017. Accessed: 2017-01-16.

[119] Trčka M., Hensen J. and Wetter M. Co-simulation of innovative integrated HVAC systems in buildings. *Journal of Building Performance Simulation*, 2(3):209–230, 2009.

[120] Trcka M. Co-simulation for performance prediction of innovative integrated mechanical energy systems in buildings. *Technische Universiteit Eindhoven*, 2008.

[121] Trimble Inc. SketchUp. `https://www.sketchup.com`, 2017. Accessed: 2016-08-02.

[122] U.S. Department of Energy. Energy Plus. `https://energyplus.net/`, 2017. Accessed: 2017-03-01.

[123] Treeck C. A. *Gebäudemodell-basierte Simulation von Raumluftströmungen*. Dissertation, Technische Universität München, München, 2004.

[124] W3C. The SPARQL query language for RDF. `https://www.w3.org/TR/rdf-sparql-query/`, 2017. Accessed: 2017-01-16.

[125] W3C. Web Ontology Language - OWL. `http://www.w3.org/OWL/`, 2017. Accessed: 2017-01-16.

[126] Wang W., Zmeureanu R. and Rivard H. Applying multi-objective genetic algorithms in green building design optimization. *Building and Environment*, 40(11):1512 – 1525, 2005.

[127] Welle B., Haymaker J. and Rogers Z. ThermalOpt: A methodology for automated BIM-based multidisciplinary thermal simulation for use in optimization environments. In *Building Simulation*, volume 4, pages 293–313. Springer, 2011.

[128] Wetter M. A view on future building system modeling and simulation. *Building Performance Simulation for Design and Operation*, pages 481–504, 2011.

[129] Wetter M. Co-simulation of building energy and control systems with the building controls virtual test bed. *Journal of Building Performance Simulation*, 4(3):185–203, 2011.

[130] Yang R. and Wang L. Multi-zone building energy management using intelligent control and optimization. *Sustainable Cities and Society*, 6:16 – 21, 2013.

[131] Yun G. Y., Tuohy P. and Steemers K. Thermal performance of a naturally ventilated building using a combined algorithm of probabilistic occupant behaviour and deterministic heat and mass balance models. *Energy and Buildings*, 41(5):489 – 499, 2009.

[132] Zhai Z. *Developing an integrated building design tool by coupling building energy simulation and computational fluid dynamics programs.* PhD thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 2003.

[133] Zhai Z. and Chen Q. Y. Numerical determination and treatment of convective heat transfer coefficient in the coupled building energy and CFD simulation. *Building and Environment*, 39(8):1001 – 1009, 2004. Building Simulation for Better Building Design.

[134] Zhang L.-J., Zhang J. and Cai H. Service-Oriented Architecture. In *Services computing*, pages 89–113. Tsinghua University Press and Springer, Beijing and Berlin and New York, 2007.

[135] Zimmermann G. A new approach to building simulation based on communicating objects. In *Seventh International IBPSA Conference Proceedings*, volume 2, pages 707–714, 2001.

List of Figures

List of Tables

List of Listings

Appendix A

Implemented Classes and Methods for the Modular Simulation

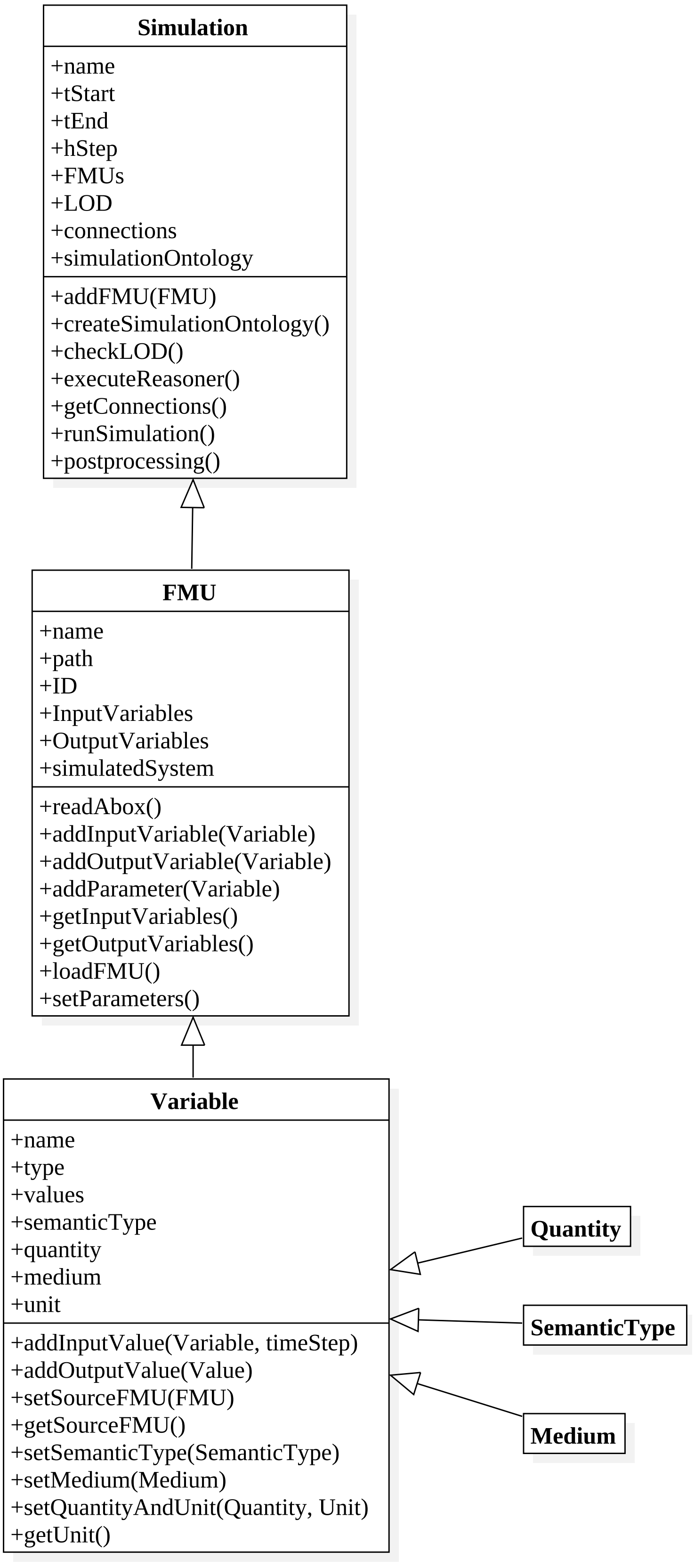
The implementation of the modular simulation in Python is built on several classes, their characterizing methods and attributes. The UML scheme in Figure A.1 illustrates these classes and their inter-dependencies.

Central to the implementation is the class *Simulation*. This class features attributes, such as start time and end time of a simulation. An instance of this class is created with a list of FMUs that form the simulation. From this list, FMU instances are generated and subsequently appended to the simulation instance. Further methods inherit functionality to create the simulation ontology from the Aboxes inside the FMUs and derive the connections among them through invoking a reasoner. Ultimately, methods for executing and post-processing the simulation are provided.

The *FMU* class features an *ID* within each simulation as well as input and output variables. Instances of these are generated through information contained in the Abox using the *Variable* class. This class features attributes corresponding to the Abox information, i.e. semantic type, quantity and medium. Instantiation of these attributes is realized based on individual classes, which are derived from the instances for semantic type, quantity and medium in the overarching ontology.

Whereas the semantic type and medium classes merely serve as descriptive elements in the implementation, the quantity classes inherit functionality to perform unit conversion during the simulation. The description of these classes is continued in sections A.1, A.2 and A.3.

Figure A.1. A UML representation of the classes, their associated attributes and methods as applied in the implementation of the modular simulation.



A.1 Quantities

The concept of the implemented quantity classes is built upon the *Ontology of units of Measure (OM)* [105]. As discussed in section 3.2.1, these definitions were extended with frequently used quantities in BPS which, in addition to their physical character, inherit further pre-defined meaning. The depicted classes in this chapter are therefore based on super-class modules. These classes describe the possible physical units of their sub-classes. After instantiation with a unit as defined in the Abox definition, they are able to provide a factor to perform conversions when unit differences with a connected variable are detected.
Since not all super-classes possess multiple sub-classes, identical naming can occur in the two modules.

Figure A.2. UML representation of the quantity Area and its sub-classes.

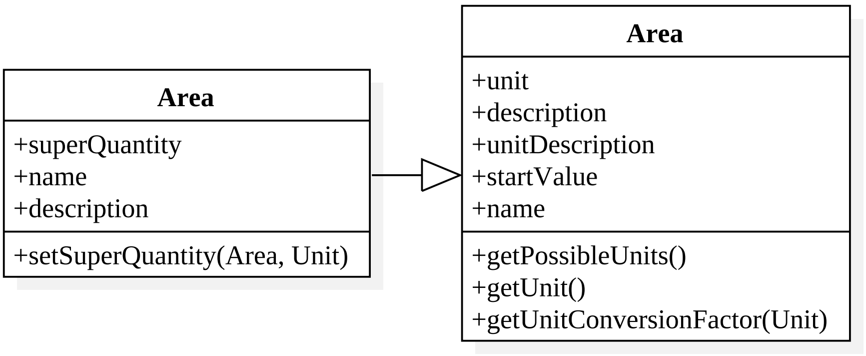


Figure A.3. UML representation of the quantity Density and its sub-classes.

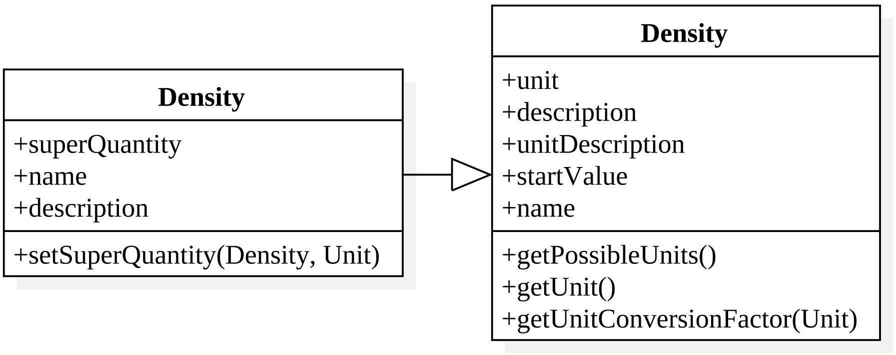


Figure A.4. UML representation of the quantity Energy and its sub-classes.

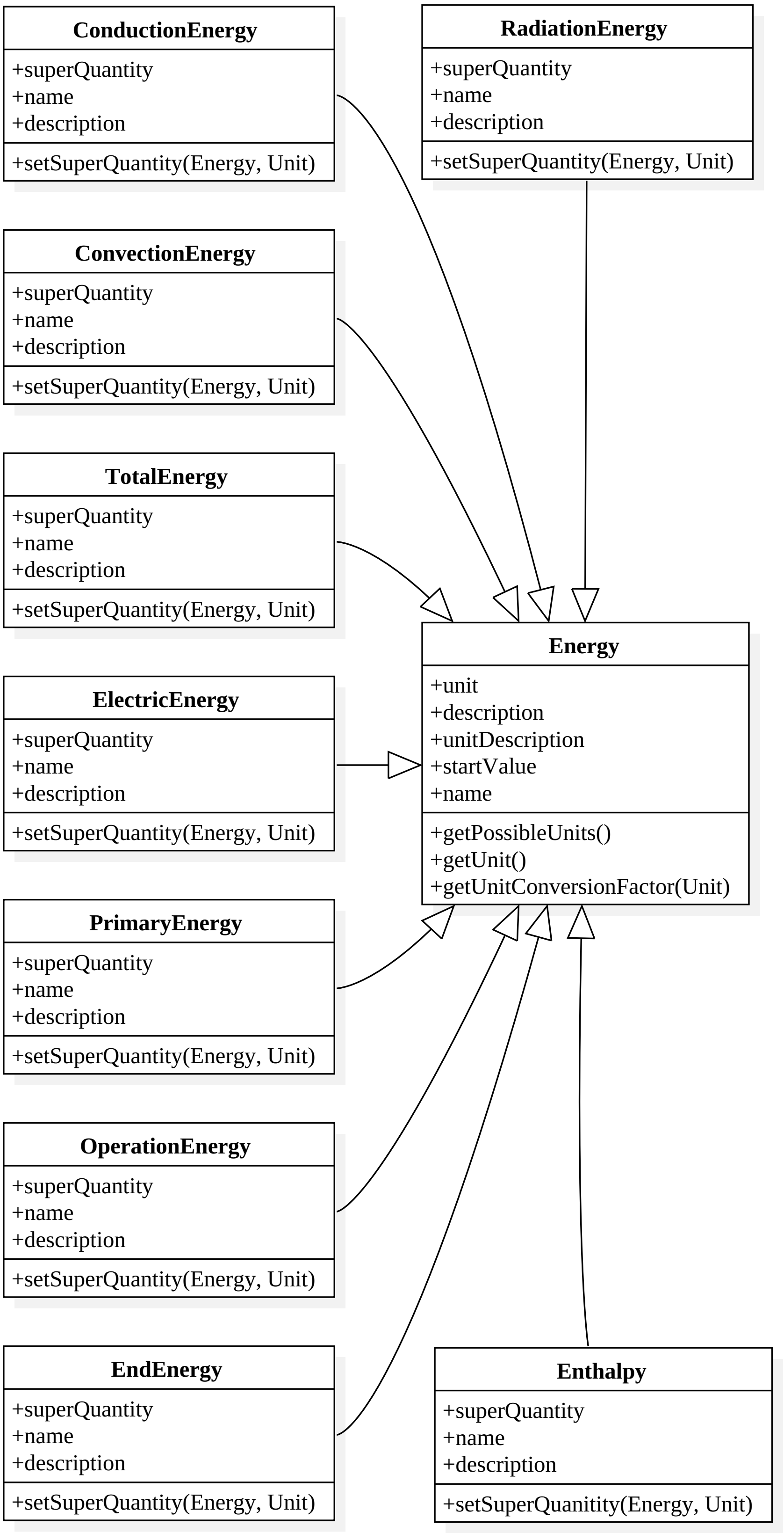


Figure A.5. UML representation of the quantity Energy Density and its sub-classes.

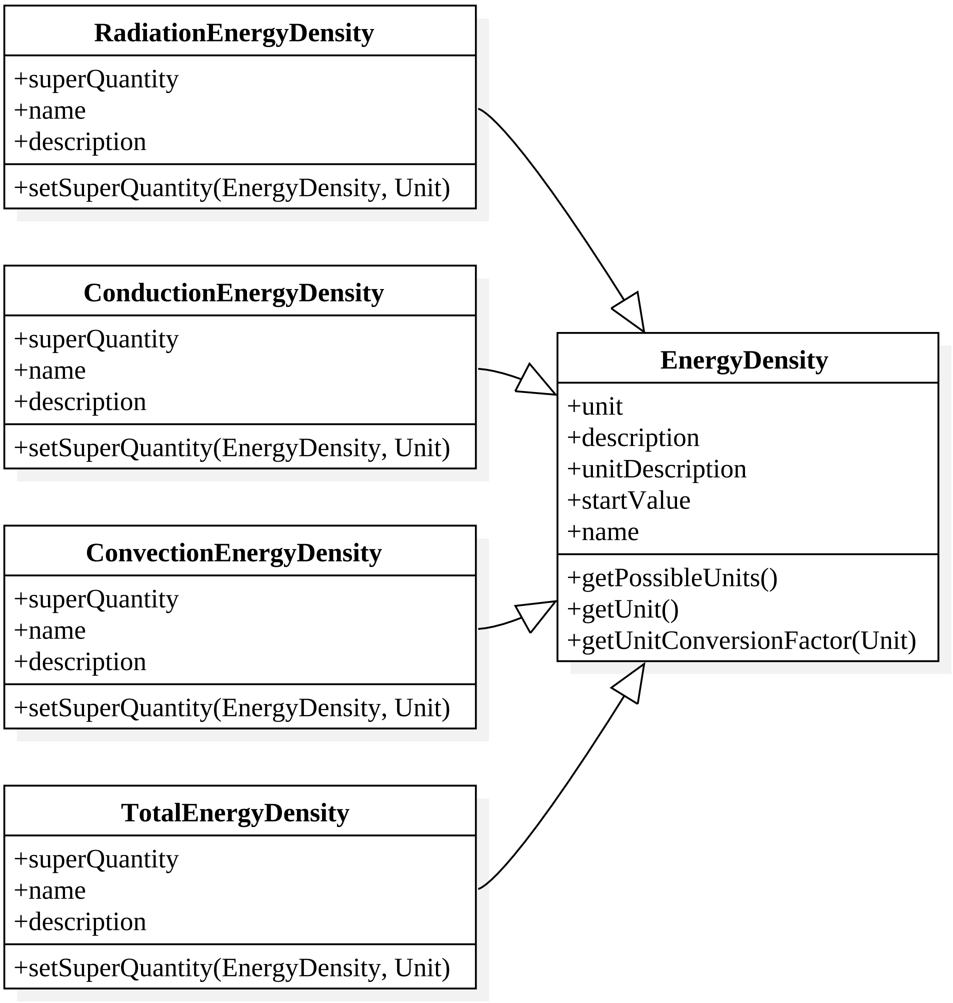


Figure A.6. UML representation of the quantity Frequency and its sub-classes.

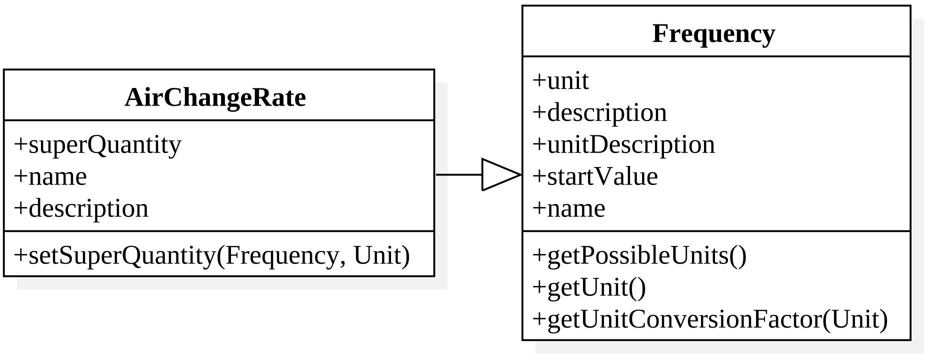


Figure A.7. UML representation of the quantity Length and its sub-classes.

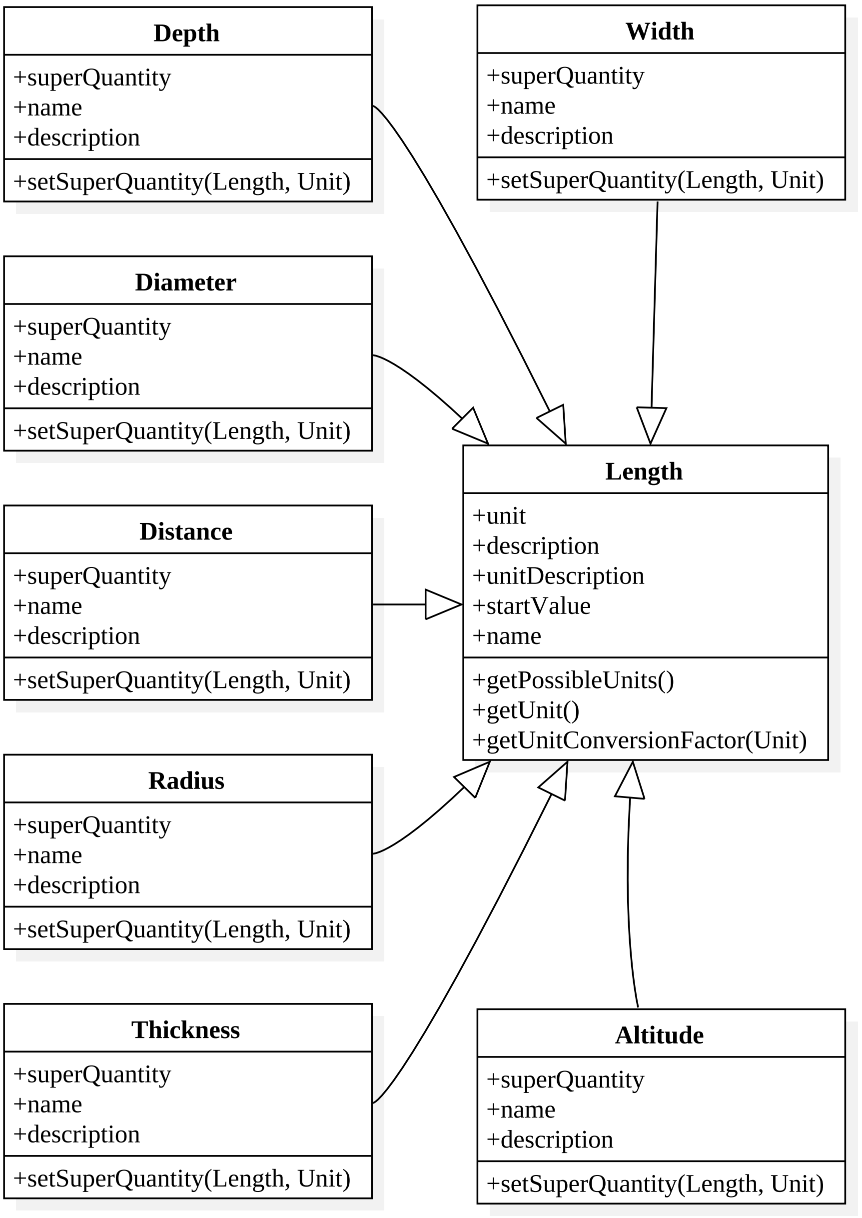


Figure A.8. UML representation of the quantity Mass and its sub-classes.

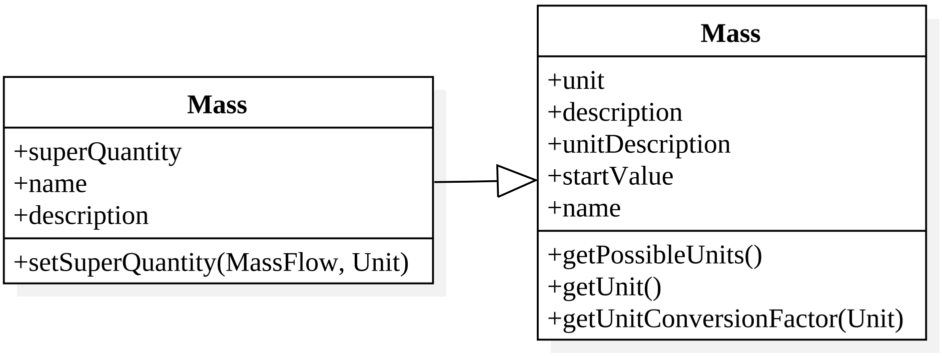


Figure A.9. UML representation of the quantity Mass Flow and its sub-classes.

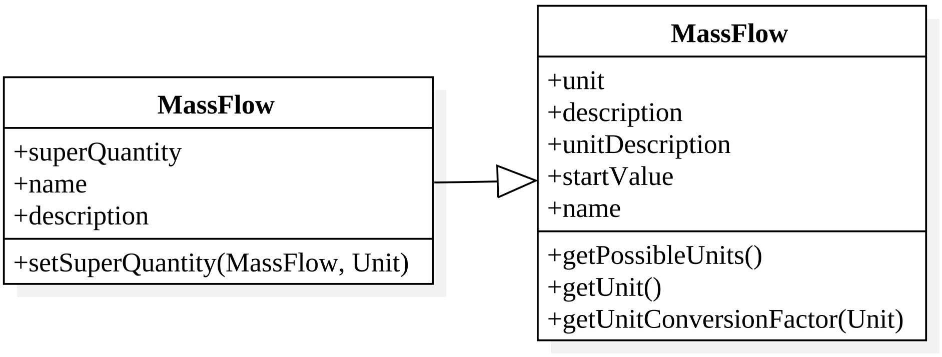


Figure A.10. UML representation of the quantity Mass Fraction and its sub-classes.

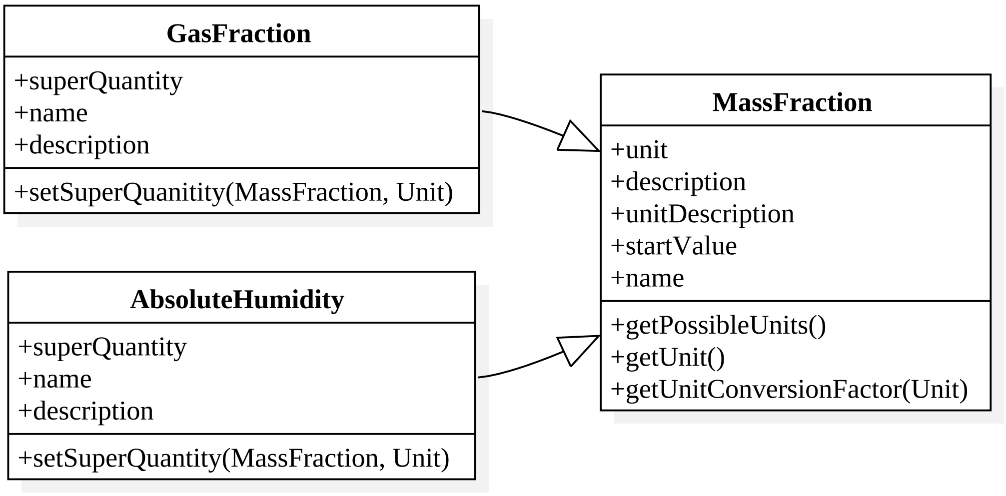


Figure A.11. UML representation of the quantity Number and its sub-classes.

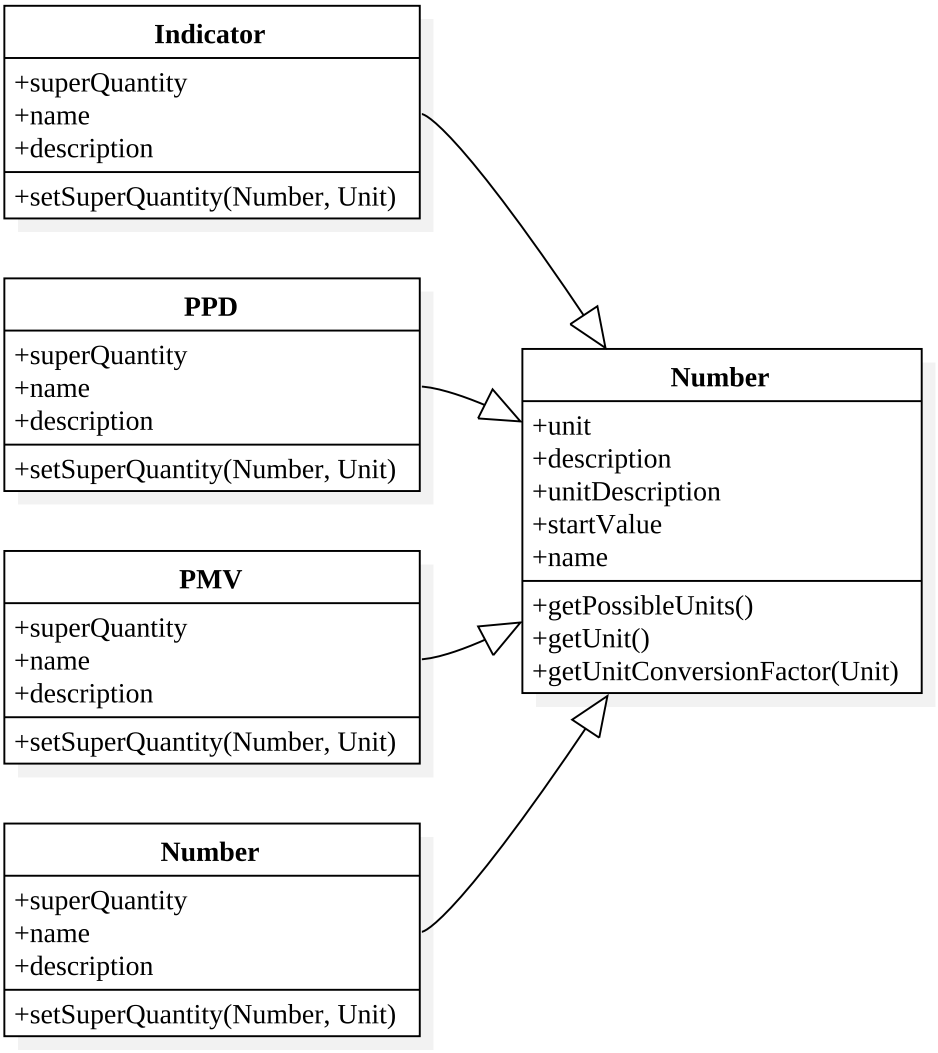


Figure A.12. UML representation of the quantity Plane Angle and its sub-classes.

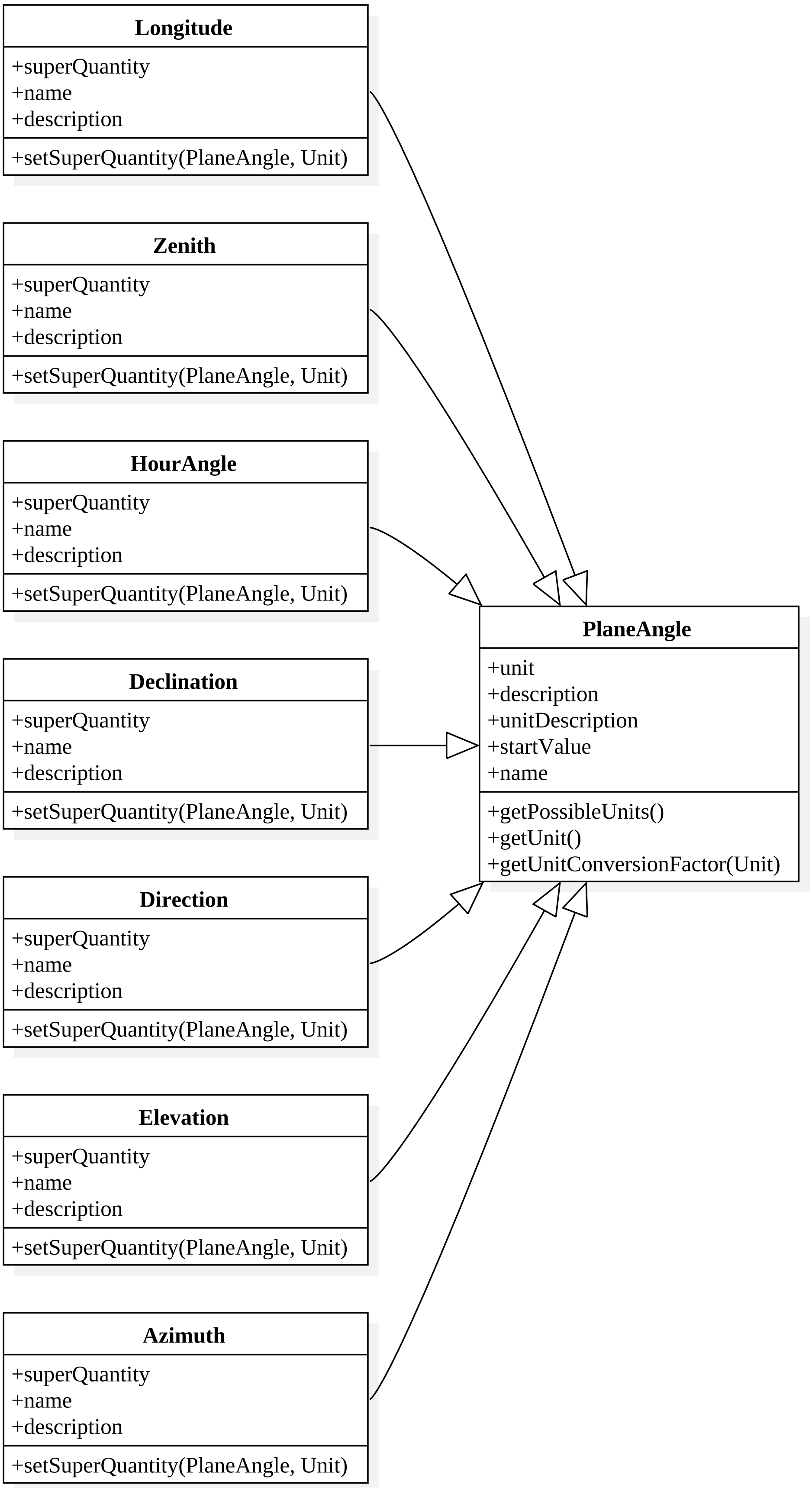


Figure A.13. UML representation of the quantity Power and its sub-classes.

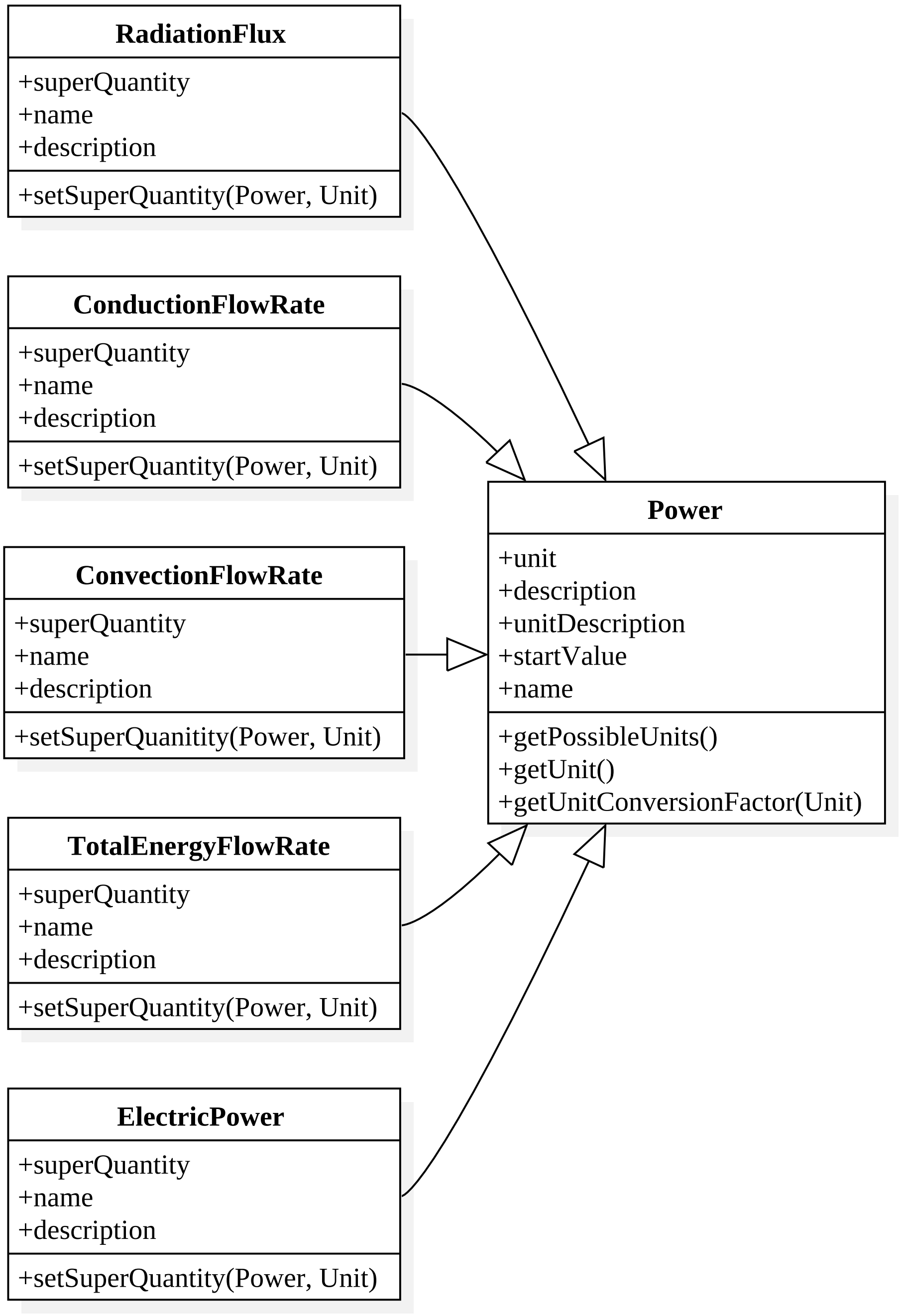


Figure A.14. UML representation of the quantity Power Density and its sub-classes.

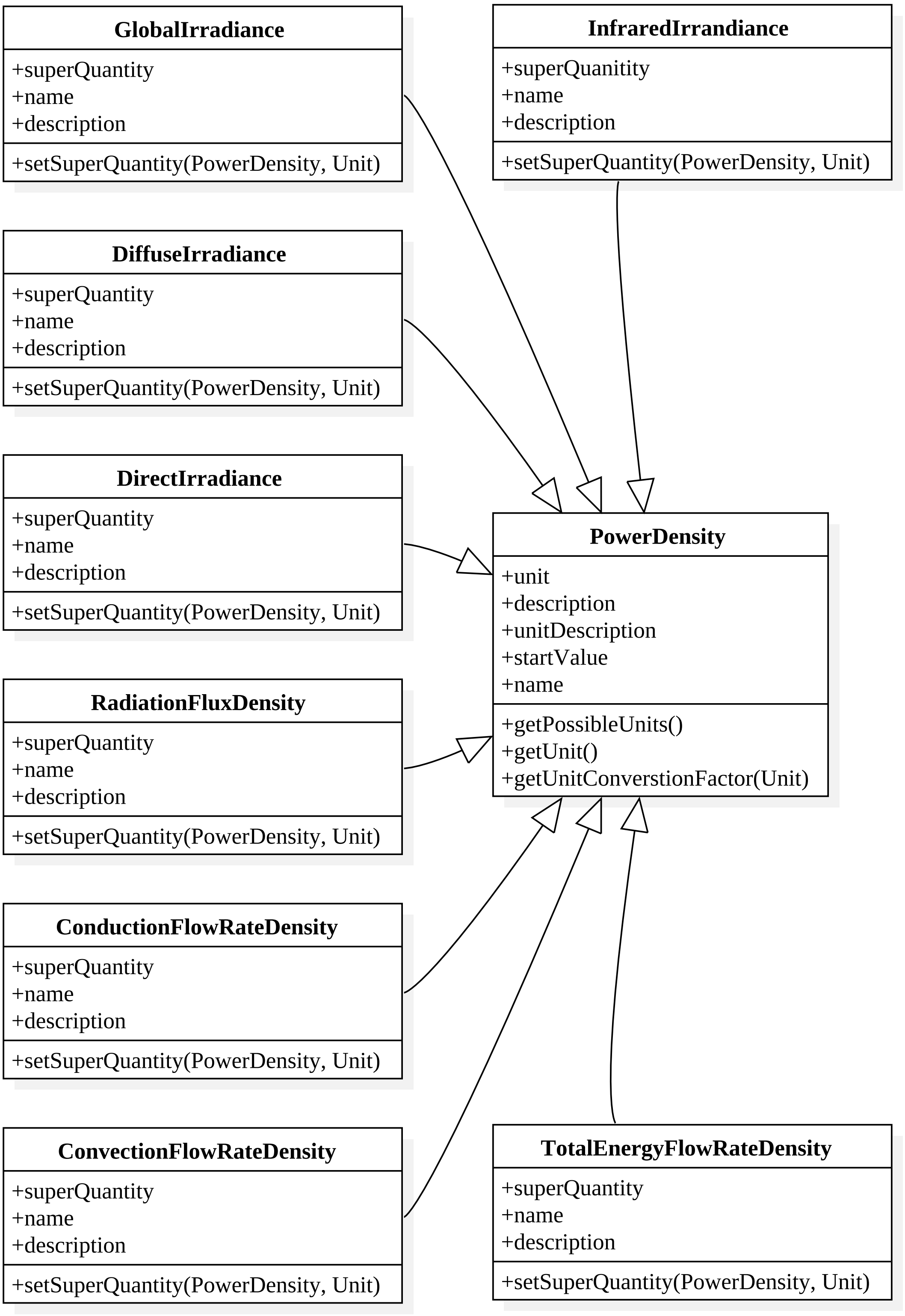


Figure A.15. UML representation of the quantity Pressure and its sub-classes.

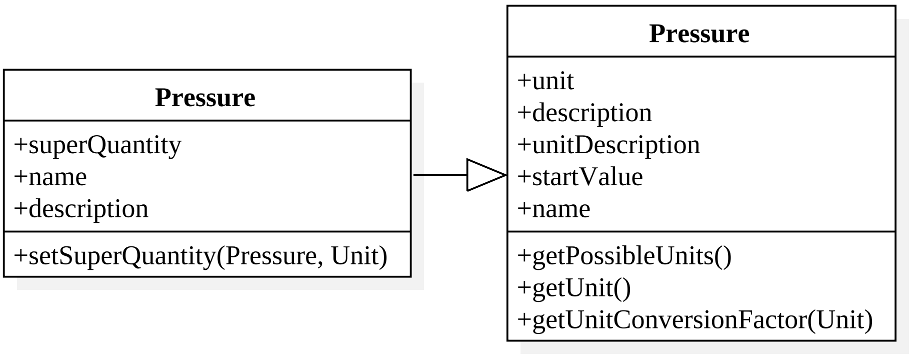


Figure A.16. UML representation of the quantity Ratio and its sub-classes.

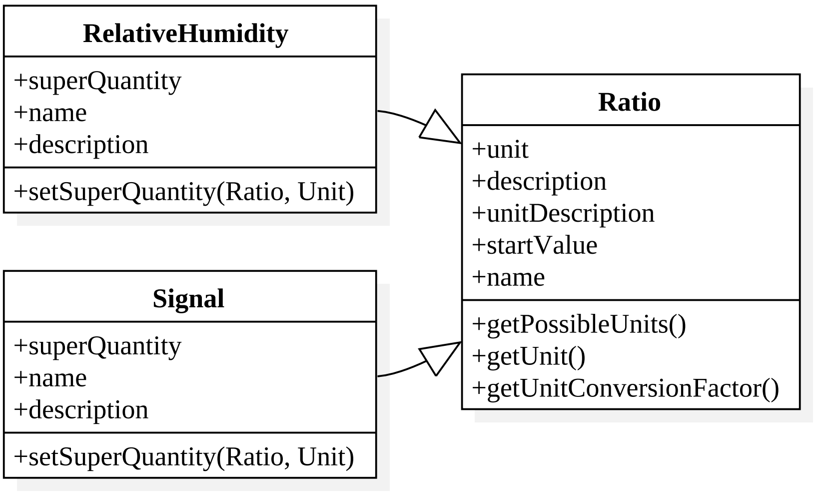


Figure A.17. UML representation of the quantity Speed and its sub-classes.

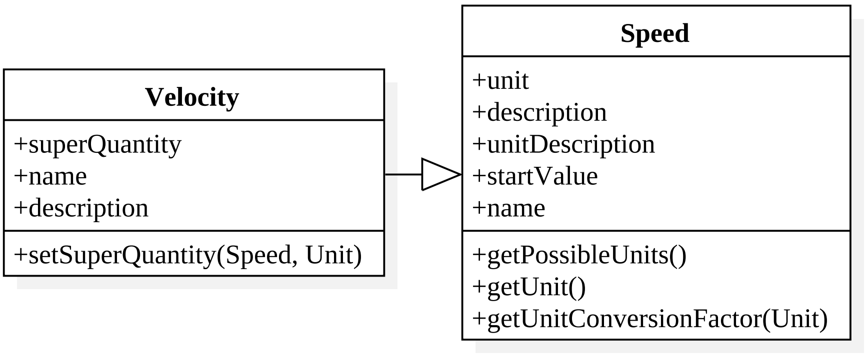


Figure A.18. UML representation of the quantity Temperature and its sub-classes.

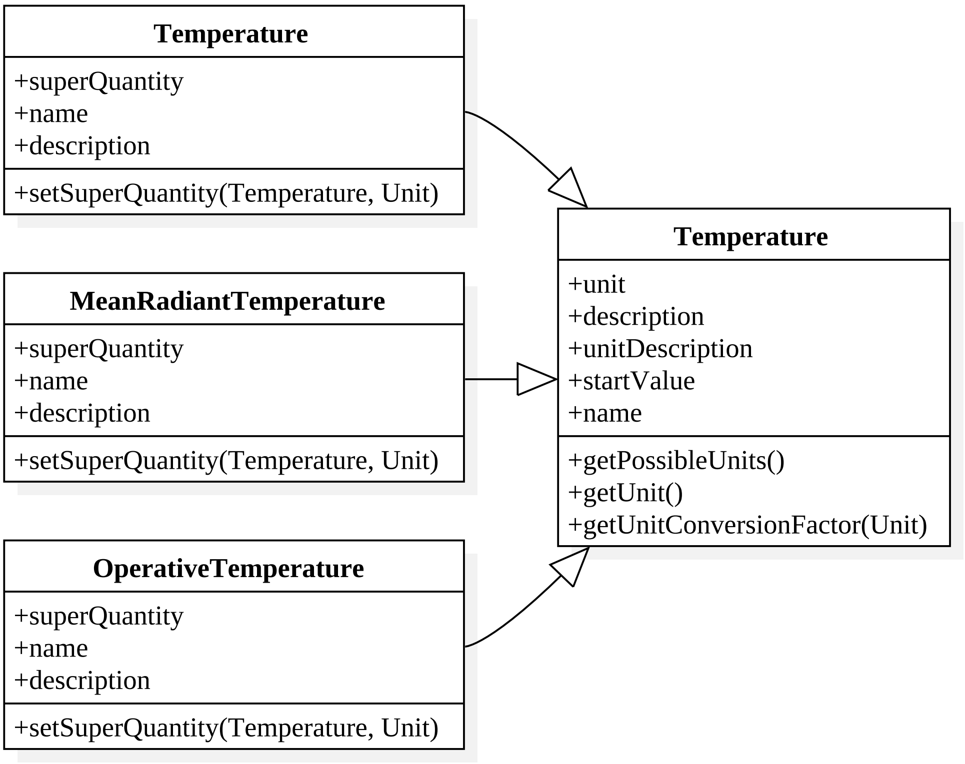


Figure A.19. UML representation of the quantity Volume and its sub-classes.

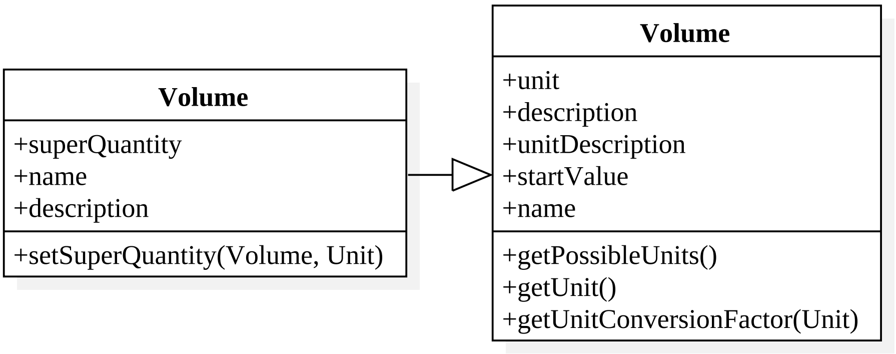


Figure A.20. UML representation of the quantity Volumetric Flow Rate and its sub-classes.

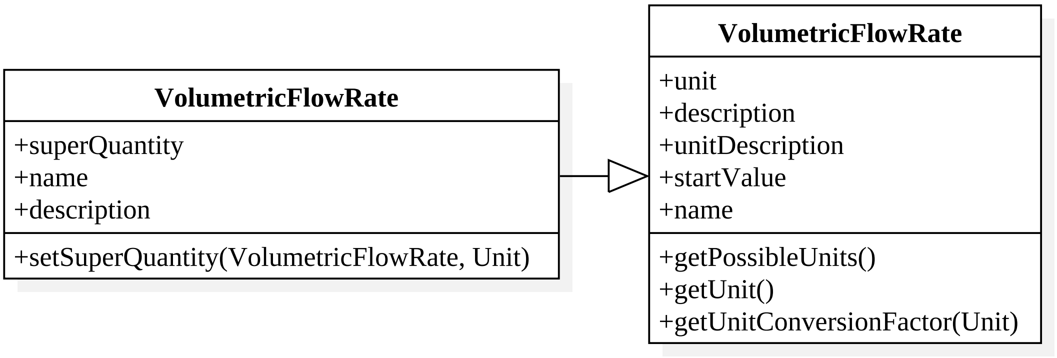
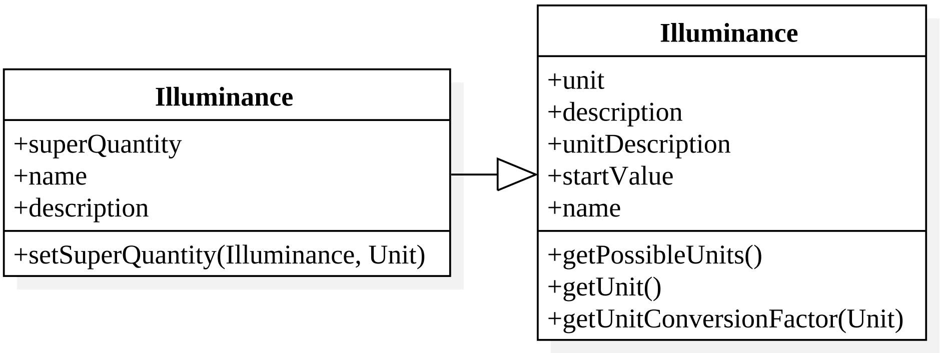


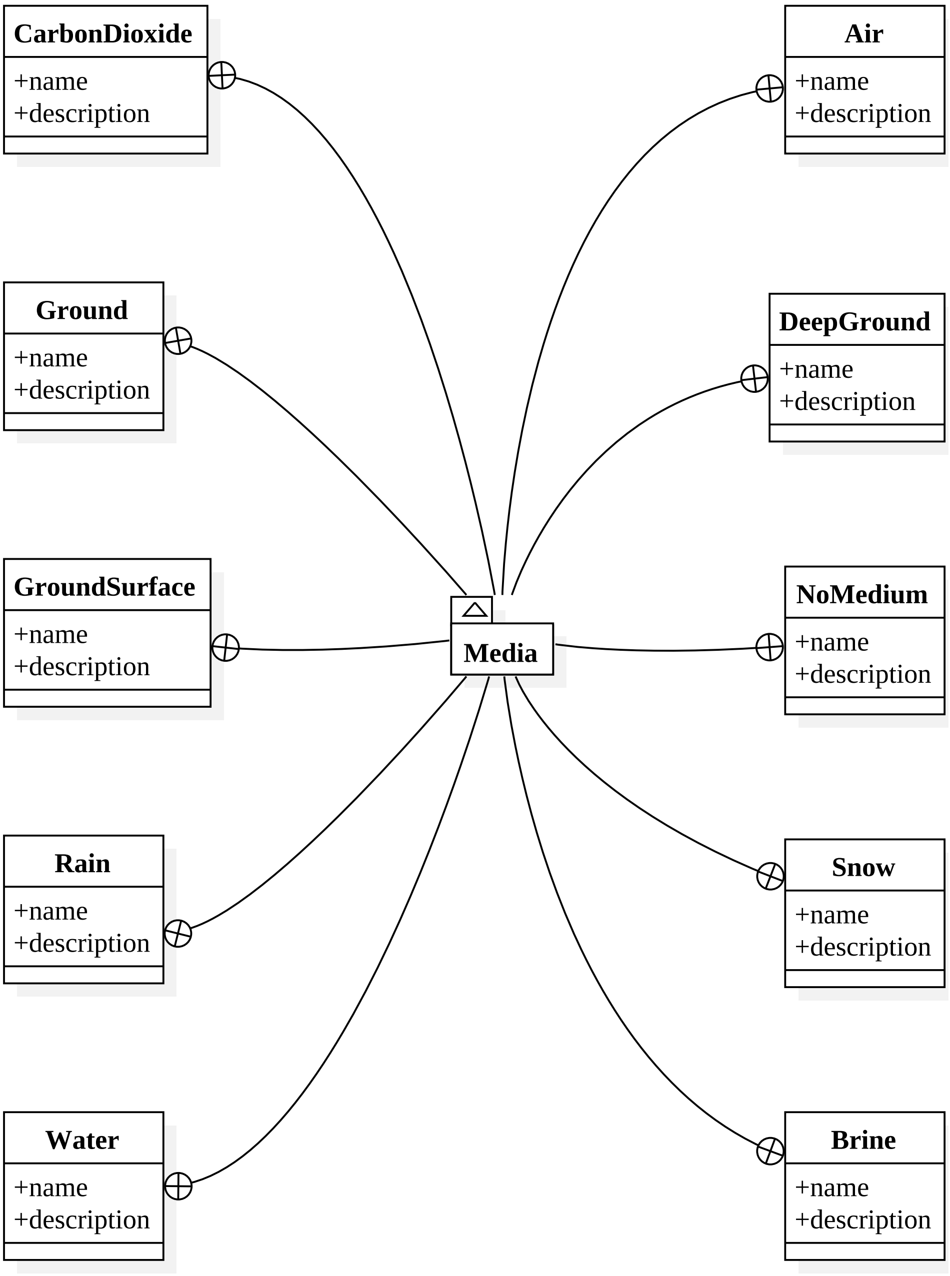
Figure A.21. UML representation of the quantity Illuminance and its sub-classes.



A.2 Media

In order to complement the definition of a variable instance, the media module provides classes with descriptions tailored to the following types of media. These allow for standardized usage in post-processing. In contrary to the quantity classes, no further functionality is inherited. Figure A.22 illustrates the content of the media package.

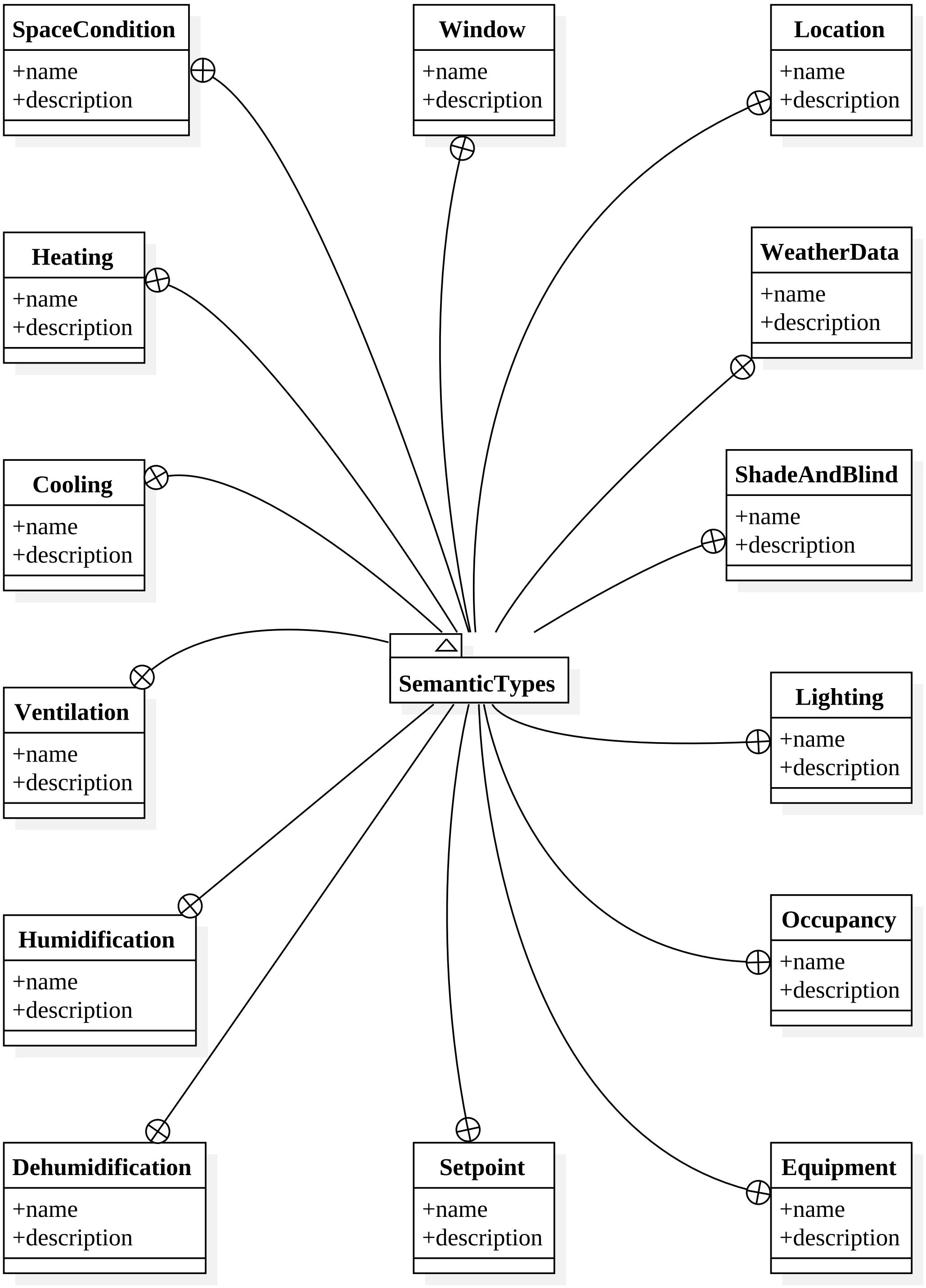
Figure A.22. UML representation of the Media package and its classes.



A.3 Semantic Types

Similarly to the media classes, the semantic type classes complement the description of variables without providing methods required to realize the simulation. Figure A.23 shows an overview of the module.

Figure A.23. UML representation of the Semantic Type package and its classes.



Appendix B

Validation of Heat Pump and Thermal Storage Model

The simulation models for the heat pump and the thermal storage are based on the Modelica IBPSA library [56]. Generic models for these components were extracted and calibrated with experimental data collected at the test site of the Fraunhofer IBP in Holzkirchen, Germany. The following section summarizes the validation of the simulation models and compares simulated and measured data for each component individually.

B.1 Heat Pump

The brine-water heat pump is operated at constant mass flow rates in both sides, condenser and evaporator. In order to validate the COP of the heat pump at these mass flow rates for different conditions, several steady-state experiments were conducted. These comprise a number of different combinations for condenser and evaporator inlet temperature. The former ranges between 30 °C and 55 °C, the latter between -10 °C and 20 °C. As a boundary condition for the experiments, these temperatures were provided to the model as well as the deployed compressor power of the heat pump. Figures B.1 to B.3 show the resulting comparisons between simulated and measured data. Either quantity as well as either combination of boundary condition can be replicated with high agreement.

Figure B.1. Measured and simulated condenser outlet temperature in steady-state experiments.

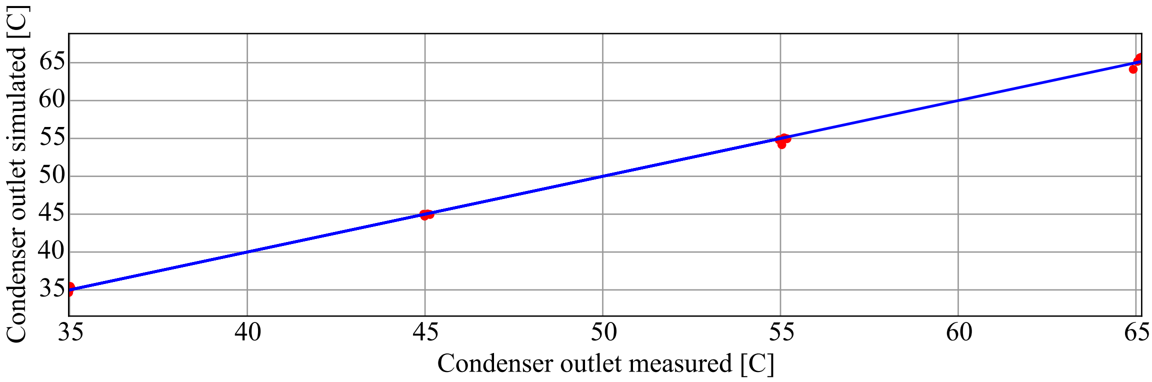


Figure B.2. Measured and simulated evaporator outlet temperature in steady-state experiments.

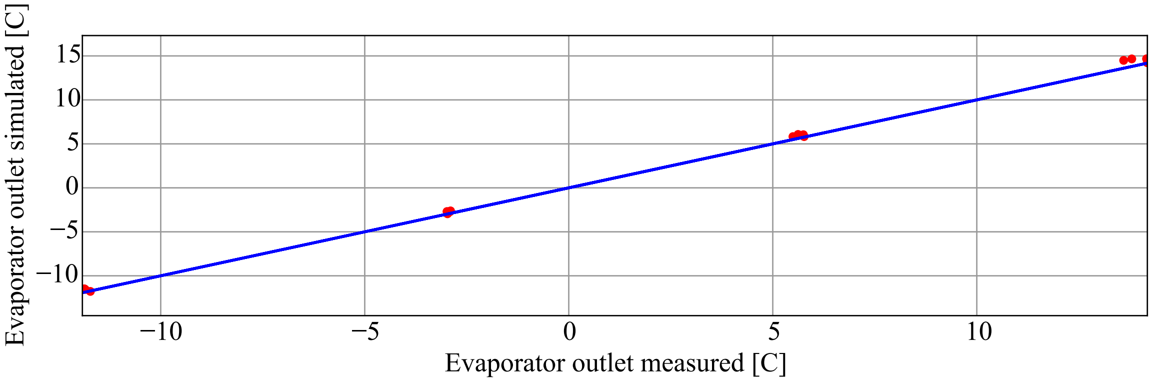
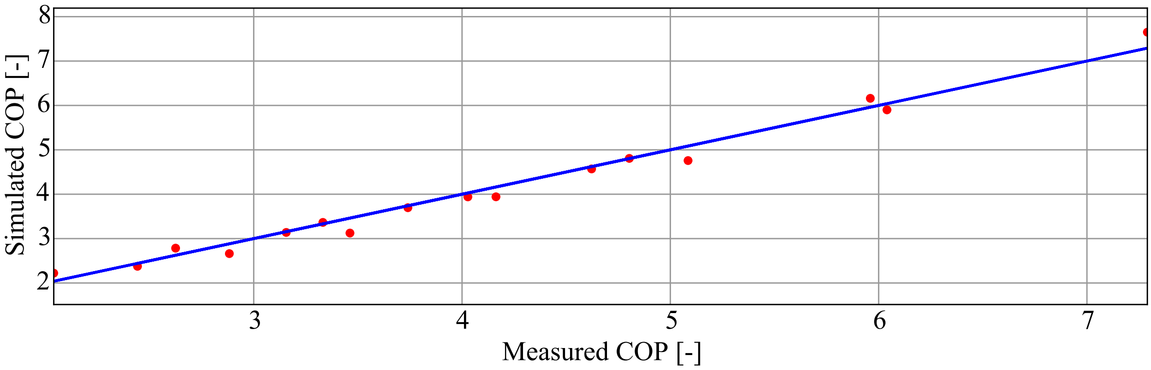


Figure B.3. COP from experiment and simulation in steady-state experiments.



In order to apply the heat pump model within an HVAC system in conjunction with other components, the operation strategy of the heat pump must be included in the model. Therefore, the data gathered in the steady-state experiments serves to derive an equation that allows for quantification of the compressor control depending on current conditions. This equation computes the applied compressor power and is determined with the condenser and evaporator inlet temperatures as dependent variables as follows. It approximates the measured data from the steady-state experiments with an R^2-value of 0.9998.

$$\begin{aligned} P_{el} = {} & 1095.26563904605 + 8.39280035611738 * T_{cond,in} \\ & + 6.77017920478803 * T_{eva,in} \\ & + 0.294981283262045 * T_{cond,in}^2 \end{aligned} \tag{B.1}$$

The integration of equation B.1 into the model, in order to simulate the internal control of the applied compressor power, allows for assessing the dynamic behavior of the heat pump during operation. Validation of this transient behavior is shown for two test runs. Both runs are characterized by a continuous increase of condenser inlet temperature. Figures B.4 and B.5

summarize the results for the first test, Figures B.6 and B.7 summarize the results for the second test. In both cases, the model is able to predict the applied compressor power as well as the resulting temperatures at condenser and evaporator outlet with high agreement.

Figure B.4. Measured and simulated condenser and evaporator outlet temperatures in experiment with continuously varying boundary conditions.

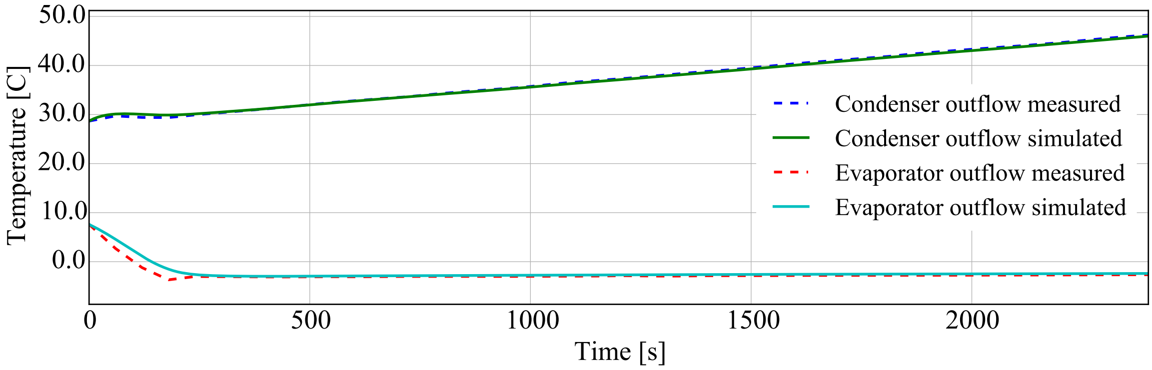


Figure B.5. Measured and simulated compressor power in experiment with continuously varying boundary conditions.

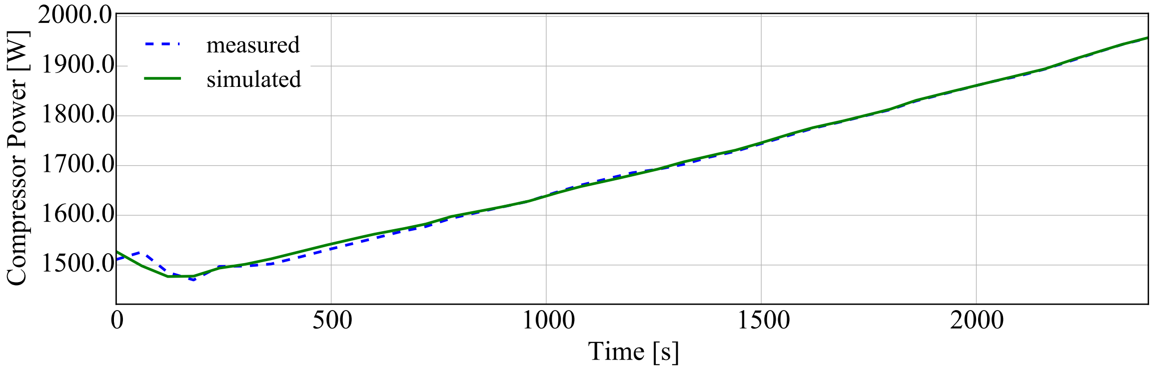


Figure B.6. Measured and simulated condenser and evaporator outlet temperature in experiment with continuously varying boundary conditions.

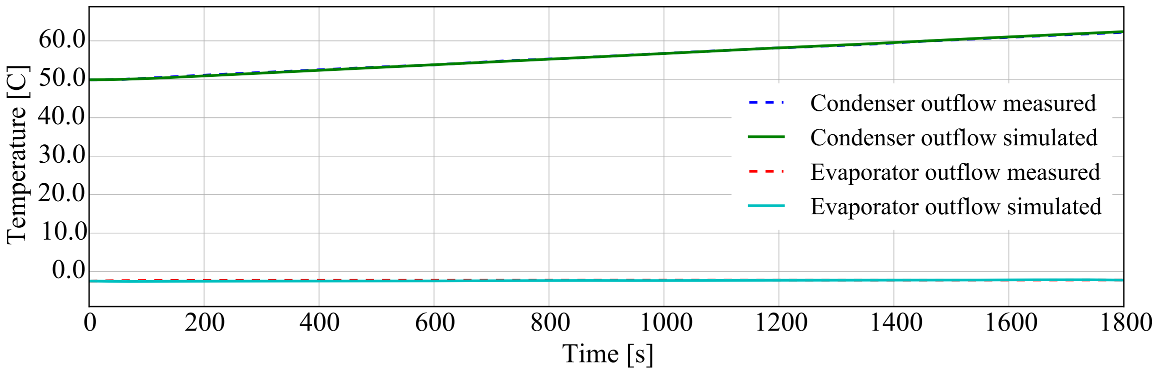
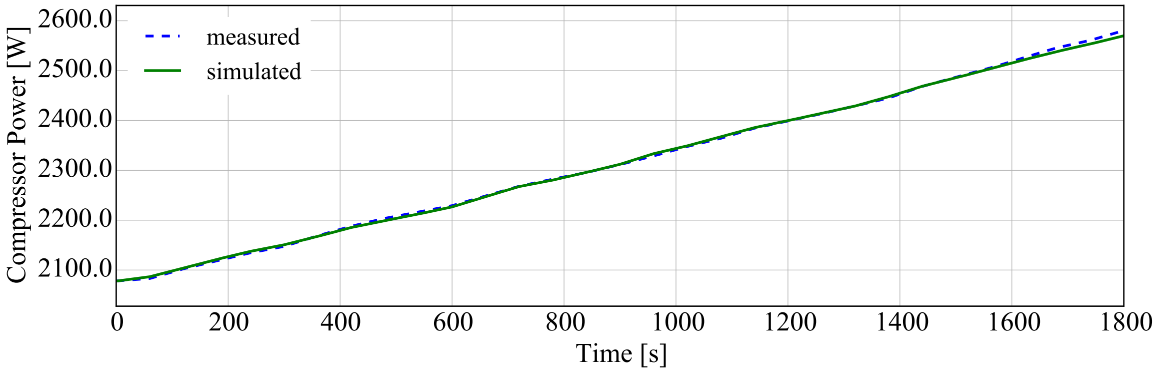


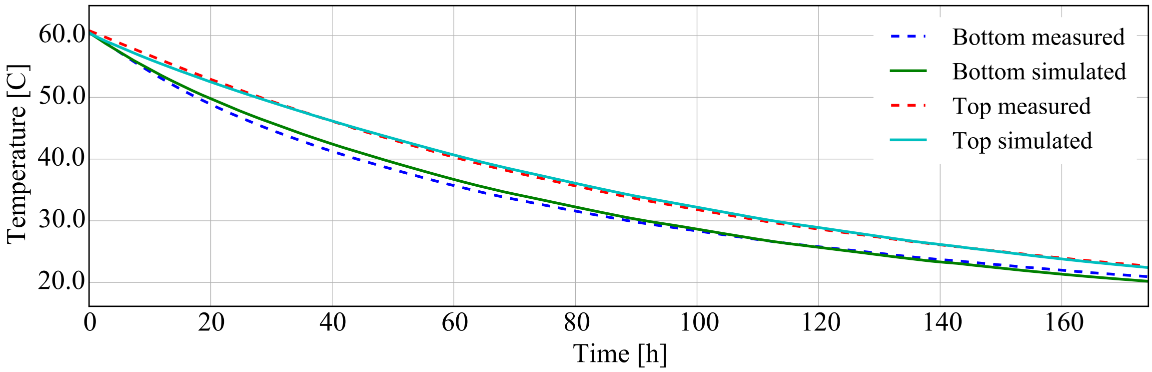
Figure B.7. Measured and simulated compressor power in experiment with continuously varying boundary conditions.



B.2 Thermal Storage

The validation of the thermal storage model was conducted in a heat loss experiment. The tank was therefore heated to a homogeneous temperature of 60 °C. The fluid temperature was measured at two locations in the tank - one in the top, one in the bottom quarter - in order to capture buoyancy effects. Figure B.8 shows the agreement of measured and simulated data for these two locations.

Figure B.8. Validation of the tank model in a heat loss experiment.



Appendix C

Summary of Case Study Simulations

In order to provide a reference for performance comparisons of FMU-based co-simulations, Table C.1 summarizes the required time for individual steps during the execution of the simulations presented in chapter 4. This includes the net time for simulating the contributing FMUs, which is the accumulated time required to execute line 48 in the pseudo-code representation of the co-simulation algorithm in Listing 10. The time for exchanging variable values, i.e. for realizing the connections in the co-simulation network, was determined to be the time required for the code lines 37 to 45. Retrieving the variable values from the current FMU states corresponds to the accumulated duration of lines 49 and 50. Simulations were executed on an Intel Core i5 processor with 2.20 Ghz.

In addition to measures regarding the time performance of the simulations, further parameters are provided, such as the realized time step, the simulated time frame and the number of possible connections corresponding to equation 3.5 as well as the number of realized connections. Besides simulations executed within the case studies presented in chapter 4, also two simulations incorporating a different time step, as discussed in section 5.2.2, are included.

Table C.1. Summary of simulations runs regarding the contributing FMUs, connections and time performance.

Simulation run	Time frame [d]	Timestep [min]	Number of FMUs	Net time for simulating [s]	Net time for value exchange [s]	Net time for value retrieving [s]	Possible connections	Inferred connections
1.1	365	2	3	558.1541	31.9340	179.0430	199	5
1.2	365	2	3	1481.7549	264.8610	266.1901	1476	68
1.3	365	2	3	3660.7350	463.9531	600.7269	10161	76
1.4	365	2	7	3551.8941	463.6470	599.1940	10029	74
1.5	365	2	7	3623.0359	453.6811	589.7260	10029	74
1.6	365	2	7	3660.1779	457.3010	597.3102	10029	74
1.6.1	365	5	7	2969.2180	182.5330	231.2900	10029	74
1.6.2	365	10	7	2644.7980	90.2370	114.1100	10029	74
1.7	365	2	17	4378.0551	516.3660	499.0890	21259	118
1.8	7	2	18	2668.6560	16.7610	16.2880	24832	135
1.9	7	2	18	2604.6630	16.7120	16.1160	24832	135
2.1	365	2	6	3164.9800	351.3810	427.4690	6247	66
2.2	365	2	7	3664.9460	386.8150	490.7430	8402	68