Virtual Computational Testbed for Building-Integrated Renewable Energy Solutions

Author: Jad Al Koussa  
Student ID: 0826610

Supervisors:  
prof.dr.ir. Jan Hensen, TU/e  
Dr. John Bynum, TU/e  
Ing. Hans Phaff, TNO  
Dr. Henk Polinder, TNO

August 5, 2014
Abstract

Buildings are increasingly moving from being not just energy consumers but also energy producers through the installation of various renewable technologies. To fully realize energy neutral buildings and the decentralization of electricity generation, efforts are needed to support the development of various technologies such as building-integrated solar systems (both photovoltaic and thermal) which can be easily implemented in an aesthetically pleasing manner. To this end, a cooperation known as SolarBEAT has been established between the Eindhoven University of Technology (TU/e) and the Solar Energy Application Centre (SEAC), an independent research organization. On the TU/e campus, an installation for the pilot scale testing of building-integrated photovoltaic/thermal (BIPVT) solar collectors is being constructed on the roof of the Vertigo building, home of the Department of the Built Environment at TU/e. The pilot scale testing includes prototype products from a variety of solar system manufacturers.

The main objective of this thesis is the development of a virtual computational testbed for assessing the full scale performance potential of the technologies involved in the SolarBeat project in addition to the possibility of calibrating the testbed according to the experimental results from the testing facility. More specifically, this work concentrates on BIPVT solar collectors for a typical detached single family Dutch house. The model is developed using the Modelica modeling language and the Dymola simulation environment. The complete model includes BIPVT solar panels, thermal storage, heating system and a complete model of the subject building’s thermal and electrical loads. It is developed using the Buildings library and the Districts library issued by the Lawrence Berkeley National Laboratory of the United States Department of Energy.

This report presents multiple building related and renewable energy related Modelica libraries. Then, it describes how the model was developed and results are presented to demonstrate how this testbed can be used. Moreover, an assessment of the advantages and disadvantages of Dymola and Modelica for this case is performed including a comparison with other tools. The results show the importance of this testbed in assessing the technologies used and its contribution in the decision making process regarding low/zero energy building designs. It also shows that Dymola/Modelica provides a lot of flexibility, ease of use and multi-domain simulation capabilities. However, the use of the buildings library provided some limitations in the simulation time and control domains.
Acknowledgements

This thesis was conducted in partial fulfilment of the requirements for the degree of Master of Science in Sustainable Energy Technology at the Eindhoven University of Technology in collaboration with TNO.

I would like to express my sincere gratitude to my supervisors prof.dr.ir. Jan Hensen, Dr. John Bynum, Ing. Hans Phaff and Dr. Henk Polinder for their support and their valuable input and guidance throughout the duration of this project. I would also like to thank the whole team of the Building Performance Simulation group at TU/e for their help, support and sharing of experience in different matters.

I would also like to thank all my friends for their full support and help during my time in Eindhoven. Last but not least, I would like to express my thankfulness to my family for providing me with the support and love to be where I am. This thesis would not be possible without them.

Eindhoven, August 2014.

Jad Al Koussa
Contents

Abstract i
Acknowledgements iii

1 Introduction 1
1.1 General introduction 1
1.2 Goal 1
1.3 Research questions 2
1.4 Report structure 3

2 Literature 5
2.1 Dymola and Modelica 5
2.2 Modelica in buildings and renewable energy simulations 6
2.2.1 Libraries 6
2.2.2 Applications 8
2.3 Comparison between Dymola/Modelica and other simulation tools 9
2.3.1 Dymola/Modelica vs TRNSYS 9
2.3.2 Dymola/Modelica vs Simulink 10
2.4 Building stock in Europe and the Netherlands 10
2.5 Energy consumption for a Dutch household 11

3 Testbed development 13
3.1 System components 13
3.1.1 Libraries used 13
3.1.2 Building 13
3.1.3 Occupants 14
3.1.4 Infiltration and ventilation 15
3.1.5 Solar collectors and heating 15
3.1.6 Domestic hot water 16
3.1.7 Electric model 16
3.2 Complete system 17

4 Testbed simulation results 21
4.1 Energy demand 21
4.1.1 Space heating and cooling demand 21
4.1.2 Electricity demand 22
4.2 Energy supply 23
4.2.1 Solar collectors production 23
4.2.2 PV production 24
4.3 Supply and demand matching 25

5 Simulation tools discussion 29
Chapter 1

Introduction

1.1 General introduction

With the increased penetration of energy efficiency measures and renewable energy systems in the built environment, buildings are moving from being energy-consumers to being active and complex producers of energy. The increased complexity of building systems imposes the need for flexible and multi-domain simulation models to be applied during the design phases in order to accurately predict the performances [22]. Those models would help in the assessment of different building related technologies on the technical and economical levels leading to the determination of their feasibility.

Also, models can play a role in determining the different building properties such as insulations, orientation and glazing in addition to the sizing of the different system components such as radiators, boilers, storage tanks, pipes and fans. All of these design capabilities are crucial for achieving low/zero energy buildings which is a requirement for the new buildings in Europe according to the EU Energy Performance of Buildings Directive (EPBD) starting December 2020 [30].

1.2 Goal

In this work, a virtual computational testbed for building-integrated renewable energy solutions is developed using Modelica, the flexible and multi-domain object-oriented modeling language. This testbed will help assess the full scale performance potential of existing products based on pilot scale testing results from the Eindhoven University of Technology (TU/e) campus and simulations which are calibrated using the pilot scale testing measurements. The pilot scale testing facility is a cooperation known as SolarBEAT [7] which has been established between the TU/e and the Solar Energy Application Centre (SEAC), an independent research organization. This facility is being constructed on the roof of the Vertigo building, home of the Department of the Built Environment at TU/e and it consists of dummy buildings with building-integrated photovoltaic/thermal (BIPVT) solar collectors as shown in Figures 1.2 and 1.3. Moreover, the work carried out in this thesis contributes to the IEA EBC Annex 60 [1] collaboration whose goal is the development of a new generation of computational tools for building and community energy systems based on the Modelica and Functional Mockup Interface standards.

The end product of this thesis will be a complete building system simulating photovoltaics, solar thermal collectors, occupants, space ventilation, infiltration, heating system, domestic hot water demand and heat storage as shown in Figure 1.1. The testbed will aid in the assessment of different technologies in order to predict their performance. These technologies will include BIPVT systems from different companies which are interested in testing their products. The hardware topic will be
part of the SolarBEAT project installations whereas the software part will be the testbed developed in this work. The combined work will facilitate the calibration and validation of models and will help in the performance prediction of the technologies with different parameters in addition to the design of the different system components.

Figure 1.1: The system to be modeled including BIPVT, heating system, a storage tank, occupants, domestic hot water and ventilation.

Figure 1.2: The dummy building and the PV collectors being installed on the roof of Vertigo as part of SolarBEAT.

Figure 1.3: The equipments inside the dummy building for measurements and data processing.

1.3 Research questions

The goals for this project led to the following research questions which will be answered in this report:

1. How useful is Dymola/Modelica for buildings and building-integrated renewable energy sources simulations and how does it compare to other tools?

2. What is the potential of BIPVT production in the Netherlands?

3. What are the measures required to lead to low/zero energy buildings?

To answer those research questions, multiple sub-questions can be derived:
1.4 Report structure

1. What are the libraries available for the development of the testbed in Modelica?
2. Which is the best library to use for the relevant tasks?
3. How can the testbed be developed to be as flexible as possible in order to adapt it to different technologies and different building models?
4. How far can we go in the design of buildings and systems with this testbed and the available libraries?
5. Which is the best test case to choose in order to assess the testbed?

The report starts with a literature review covering Modelica and Dymola as well as the libraries and previous similar projects related to simulation of buildings and renewable energy systems in the built environment. In addition to that, a small literature review of the Dutch residential building stock is shown along with related energy consumption figures which will provide a reference for later discussion. The literature review continues with a comparison of Dymola/Modelica with other building simulation tools especially TRNSYS and Simulink. After that, a description of the different components of the testbed and the whole system is provided with detailed information regarding the inputs, assumptions and model architecture. Following that, some simulation results that can be produced by the testbed are presented with a discussion of the usefulness of the data in low/zero energy buildings design and other purposes. Moreover, a discussion of the results and the experience of using Dymola and Modelica is provided in addition to a general comparison with TRNSYS, MATLAB-Simulink and ESP-r. The report finishes with a summary of the conclusions and provides recommendations for the future work required to enhance this testbed.
Chapter 2

Literature

The development of the testbed required a literature review concerning the use of Dymola/Modelica, how it compares to other tools, the available libraries, previous applications and similar projects. For this reason, the literature review was carried out as follows. First, Modelica and Dymola were investigated in addition to the different building-related and renewable energy-related libraries implemented in Modelica. Furthermore, a review of the literature related to the comparison between Dymola and other building simulation tools was performed. This comparison included different aspects such as available libraries, model development time, model simulation and flexibility. Another aspect of the literature review included a small overview of the building stock in the Netherlands which led to a decision regarding what reference building to use for the specific case study in addition to a review of the energy demand of Dutch households.

2.1 Dymola and Modelica

Modelica [8] is an open-source, high-level object-oriented modeling language used to model multidomain physical and control systems. It allows the combined modeling and simulation of electrical, mechanical, thermodynamic, hydraulic and other kinds of systems within the same application. It differs from other widely used programming environments by the fact that it uses non-causal modeling. On one hand, causal modeling is based on assignment statements in which the input-output causality is fixed and the model can be described using Ordinary Differential Equations (ODE). On the other hand, non-causal modeling is a declarative modeling style which is based on equations rather than assignment statements [23]. The solution direction of the equations adapts to the data flow context that is defined by declaring which variables are required as output and which are the external inputs, resulting in a more natural way of describing the system [23]. It is possible to described this type of modeling using Differential-Algebraic Equations (DAE) [23]. The modeling in Modelica can be done either through a graphical interface or through modeling by equations allowing the hierarchical inheritance and reusability of the different classes and components [18][23].

There are multiple simulation tools with the capability of reading and executing the Modelica language such as Dymola, OpenModelica and JModelica. OpenModelica and JModelica are open source tools which can be downloaded and used for free. However, there are limitations regarding the compatibility of these tools with some developed libraries such as the Buildings Library (see Section 2.2.1). Currently work is being carried out to fix the compatibility issues [6].

Dymola (Dynamic Modeling Laboratory) [11] is a commercial tool developed by Dassault Systèmes AB that is suitable for modeling various kinds of physical systems. It supports hierarchical model composition, component reusability and composite non-causal connections. Dymola possesses a powerful graphic editor for composing models and a symbolic translator for Modelica equations.
that generate C code for simulation which can be exported to MATLAB-Simulink and hardware-in-the-loop platforms [11].

For the work carried out in this report, Dymola was chosen as the simulation tool since it is compatible with the libraries chosen for the development of the testbed.

## 2.2 Modelica in buildings and renewable energy simulations

The use of Modelica is widespread in multiple industries such as the power and the automotive industries but the use of this language for modeling related to the built environment is not as common. However, current research in the application of Modelica in the built environment is gaining momentum and new libraries and components related to building simulation are being developed. In what follows, an overview of some of these libraries is presented in addition to some application examples.

### 2.2.1 Libraries

#### Buildings library

This library was developed by the Lawrence Berkeley National Laboratory (LBNL) of the United States Department of Energy [26]. It is a free open-source library containing dynamic simulation models for building energy and control systems and it is mainly based on the Modelica.Fluid library [35]. In total, the Buildings library contains about 200 models and blocks, 70 functions and 200 example and test models grouped into 10 packages with a total size of 480 MB. The important packages of the library are [35]:

- **Air flow**: This package contains models for multi-zone air flow and contaminant transport.
- **Boundary Conditions**: This package contains models to read and compute boundary conditions such as TMY3 weather data, solar irradiation and sky temperatures.
- **Controls**: This package contains blocks that model continuous time and discrete time controllers. It also contains blocks that can be used to schedule set points.
- **Fluid**: This is the largest package of the library and contains component models for air-based and water-based HVAC systems.
- **Heat Transfer**: This package contains models for steady-state and dynamic heat transfer through opaque constructions such as multi-layered walls. It also contains models for heat transfer through glazing systems.
- **Media**: This package contains different implementations for various media that can be used for building HVAC applications.
- **Rooms**: This package contains models for heat transfer in rooms and through the building envelope. The room package contains a building envelope model that can be integrated with the Airflow package to consider building dynamics coupled to the indoor environment including infiltration or the air flow rate between different rooms caused by pressure differences.
- **Utilities**: This package contains utility models for computing thermal comfort, for exchanging data with the Building Controls Virtual Test Bed (BCVTB) [28] or for computing psychrometric properties.

Concerning the renewable energy sources, this library contains models used for the simulation of solar thermal systems. Top-level models are available for solar collectors based on the ASHRAE93 and EN12975 test protocols. With those models, concentrating, flat plate and tubular collectors can be modeled. In addition to that, a stratified heat storage tank with a heat exchanger is available which can be used in combination with the solar collectors. Also, a borehole model is present. This library lacks electrical components including PV collectors.
2.2. Modelica in buildings and renewable energy simulations

The Buildings library is the most robust open source library that is well documented and supported and some of the models such as the room model and the window models have been validated [35] [36].

Districts library

The Districts library [3] is another library developed by LBNL. The eventual goal of its development is to be integrated with the Buildings library. However, this library is not as well documented and tested as the Buildings library and it is still in the experimental phase and no new versions have been developed for some time. The advantage of the Districts library is that it contains electrical models related to districts and to buildings. It contains a PV model and a wind turbine model in addition to the possibility of modeling the electrical demand using time series or linear regression [3].

Integrated District Energy Assessment by Simulation (IDEAS) library

IDEAS is a collaboration between different divisions (Electrical Energy, Building Physics, and Applied Mechanics and Energy Conversion) of the Catholic University of Leuven (KU Leuven) [20]. It incorporates the dynamics of hydronic, thermal and electrical processes in addition to networks in buildings and districts, into a single model and solver [20]. It is one of the most holistic approaches toward modeling the different aspects of buildings involving different renewable energy sources, batteries, thermal storage and electric vehicles in addition to smart grids (Figure 2.1). This library differs from the Buildings library by the fact that it actually connects the electrical models with the other thermal and hydronic models but the degree of detail is less than that of the Buildings library. However it is still in the development phase and no official version of it is available yet (the version available now is for the developers).

![IDEAS Library structure](image)

Figure 2.1: IDEAS Library structure [20].

Green buildings Library

TU Dresden together with ITI GmbH and the Honda Research Institute Europe have developed the Modelica-based "Green-Buildings" library [31]. The library offers models in the renewable, thermal, electric, eMobility, cost and user behavior domains. This library is a commercial one and it allows the modeling of charging stations and electric cars [31].
Other libraries

In addition to the libraries described above, there were a couple of others in the literature. RWTH Aachen University developed a Modelica library for HVAC-systems and building models [21] and UdK Berlin issued a buildings library that offers a large number of building and HVAC models [32]. In addition to these, XRG Simulation GmbH published the Modelica Hydronics library which is designed for the modeling of large fluid circuits and contains constant and variable speed pumps, heat exchangers and different models for liquids [32].

All the libraries mentioned in this section have been used in different applications and in what follows, some of these applications are presented.

2.2.2 Applications

Multiple projects were carried out involving buildings and renewable energy modeling and simulations in Dymola/Modelica. Those projects vary from specific applications such as the design of a PV cooling system for a residential building [10] to more general applications such as the simulation of energy conservation measures and implications for a combined heat and power district heating system [24]. The Green Buildings library was used to model a semi-detached single family house with multiple renewable energy technologies as shown in Figure 2.2.

A project carried out in [15], modeled a railway station building through thermal and electrical load calculations. To address this problem, the authors developed the architectures of the thermal and the electrical models using the Dymola/Modelica environment. The thermal model is based on the elementary components of the Modelica Buildings Library associated with other internal (GDF SUEZ) and external libraries [15]. As for the electrical model, it was implemented using a specific-purpose library calibrated on real data provided by measurements of the power consumption of several pieces of equipment inside a railway station. In order to test the capabilities of the model the authors used a reference railway station to assess the thermal and electrical consumption of the building [15].

Moreover Dymola/Modelica was used for the simulation of a smart grid system involving different electrical components. More specifically, the model consists of a DC model of a household and a
medium DC voltage grid. The household model possesses micro production, such as wind power and solar power, and a battery to be able to operate when the external grid fails [10].

The libraries and applications stated in this section form a solid basis to develop the testbed in this thesis. In addition to that, it is important to look through literature to compare Dymola/Modelica with other tools in order to know the benefits and drawbacks arising from using it. This is the topic of the next section.

2.3 Comparison between Dymola/Modelica and other simulation tools

Within the building energy analysis community, a very wide variety of simulation tools has been developed and enhanced. In this work, the focus is on comparing Dymola/Modelica with the following simulation tools: EnergyPlus, ESP-r, TRNSYS and MATLAB-Simulink.

Modelica differs from the traditional building simulation tools mainly by the approach it takes to handle equations and code complexity [27]. The "traditional building simulation programs" are the ones written using an imperative language, such as FORTRAN, C or C++. Examples of traditional building simulation programs include ESP-r and EnergyPlus. Those programs mix the code describing the physical process with the data management code and the numerical solution methods. In those programs the models are written in a way that was motivated by how computers process instructions and not the actual behavior of physical systems [27]. This leads to a semantic gap between the simulation model and the actual component.

On one hand, in the traditional building simulation programs, the program code is hard to maintain and to change since the component models frequently integrate their own numerical solver and mix the equations that simulate the physical behavior with the program flow logic [27]. This also leads to numerical noise and makes it hard to integrate models from different disciplines. Moreover, the majority of the simulation programs do not model the dynamics of HVAC systems and use an idealized controller based on the heating and cooling demand and not a controller based on a thermostat. This makes it difficult to implement and test different control strategies [27].

On the other hand, Modelica is implemented in a way to overcome the limitations of the traditional building simulation programs. The efficient numerical solutions, the handling of complexity, the efficient simulation of dynamic effects and other properties form the basis for the advantages of Modelica over other simulation tools [27].

In what follows, a comparison between Dymola/Modelica and TRNSYS is presented along with a comparison with MATLAB-Simulink.

2.3.1 Dymola/Modelica vs TRNSYS

A comparison between Dymola and TRNSYS has been performed in [25]. The comparison focused mainly on the development time, the simulation results and the simulation time. The outcome of this paper was that the development time using Modelica is lower by a factor of 5 to 10 times. In the specific case of the paper, this translated into about one year of labor savings. This decrease in development time is due to the hierarchical nature of Modelica and the ability to reuse and inherit models and sub-models [25]. Since Modelica is an equation-based object-oriented language which helps in eliminating the need for routine development for input and output data and for the management of the large amount of data involved in building simulations, the code has a smaller size. In this case, the code was 4 times smaller. Concerning the simulation times, the time for the Modelica model was 3 to 4 times longer than that of TRNSYS [25].
2.3.2 Dymola/Modelica vs Simulink

Concerning the comparison between Modelica and Simulink, one paper considered the implementation of an inverted pendulum on a chart case to compare both tools [18]. The first thing noticed is that the MATLAB-Simulink model is more complex than the Dymola/Modelica model and the way it is modeled does not depict the real physical system as opposed to Dymola/Modelica. In addition to that, the development time of the MATLAB-Simulink model was around 4 times larger than that of the Dymola/Modelica model [18]. On the results side, both simulations led to the same results with a small difference from the measured value for the pendulum angle which the authors attributed to the way the Dymola/Modelica sensor used, measures the angle.

The paper concludes that Modelica is a better choice for modeling and simulation but it suggested that the best solution would be to combine both tools, Dymola/Modelica for modeling and MATLAB-Simulink for simulation and subsequent analysis [18, 12].

In the literature found, the comparison between Dymola/Modelica and other specific tools is limited to the comparison with Simulink and TRNSYS. To further elaborate on this topic, in the discussion section of this thesis a table containing a general comparison with Simulink, TRNSYS and ESP-r has been developed. This table is based on our experience with Dymola/Modelica during the course of this work and the experience of other students with other simulation tools within TU/e.

The literature concerning the simulation tool, its application in the building and renewable energy domain and its comparison with other simulation tools has been covered in this section. The next two sections cover the overview of the building stock and the energy consumption of Dutch households which form the basis for choosing the test case and the subsequent energy demand and supply calculations in this report.

2.4 Building stock in Europe and the Netherlands

Residential buildings are mainly used for housing and living. According to the Buildings Performance Institute Europe (BPIE), the residential buildings sector is the most predominant sector in the building stock and it constitutes around 75% of the total floor space (m²) of the buildings in Europe [9]. Those residential buildings can be further divided into single family houses which constitute around 64% of the floor space while the remaining 36% of the floor space are apartment blocks [9].

As shown in Figure 2.3 in the Netherlands single family houses comprise a large share of the residential buildings with around 78% of the total floor space. It is very important to know the age of a building in order to have an idea about the energy consumption of a building since the materials used and the style have changed over the years. The BPIE survey has classified buildings into 3 different age bands (specific chronological periods) for each country:

- Old: typically representing buildings up to 1960.

In the Netherlands, around 35% of the residential buildings are considered "Old" whereas 46% are considered "Modern" and the remaining 19% are considered as "Recent" [9].
2.5 Energy consumption for a Dutch household

In order to develop the testbed and the subsequent test case, it would be useful to know the average electricity demand and gas demand for a Dutch household. This data will be used as an input and as a basis to compare the produced results.

A Dutch household uses on average 3,500 kWh of electricity per year [13] mainly for cleaning, cooling, lighting, heating and ICT as shown in Figure 2.4. In addition to that, Dutch dwellings use on average 1,500 m$^3$ of gas (Figure 2.5), the majority of its use is for heating of the dwellings.

With this section, the literature review chapter is completed. All the data presented here will be used for the next chapter which describes the method and model implementation for the development of the testbed.
Chapter 3  

Testbed development

In this chapter the case study is described in addition to the different aspects taken into account for the modeling and simulation.

3.1 System components

This work focuses on residential dwellings rather than office and commercial buildings. For this reason, after the overview of the Dutch building stock was carried out, it was decided that a reference "Modern" detached Dutch single family house will serve as the basis for the case study [37]. This choice was mainly taken, among others, because it is simpler to model compared to a row house since one does not need to care about heat transfer from other connected buildings. The different aspects of the testbed that are taken into account are described below.

3.1.1 Libraries used

Comparing the different libraries available for building and renewable energy simulations, we decided to use the Buildings library for the thermal part of the model and the Districts library for the electrical part. The Buildings library was chosen because it is an open-source library that contains most of the needed models, it is well documented and supported and some of its models are validated. The other libraries are either not well documented, commercial, not yet complete or they do not contain all the important components. Also, the Districts library was chosen since it represents the electrical part of the Buildings library even though it is not well documented yet but it has the necessary components.

3.1.2 Building

The building is a modern single family home whose elevations and floor plans are shown in Figure 3.1. In order to model the home, it is assumed that the dwelling has 10 thermal zones: 3 zones at the ground floor including the kitchen, living room and the entrance; 5 zones on the second floor including 3 bedrooms; and the second floor/attic which is divided into 2 rooms. Concerning the characteristics of the constructions used, they are shown in Table A.1 in Appendix A.

The different zones are connected to each other thermally through the common walls and through the doors which can be controlled to be open or closed (Figure B.8 in Appendix B). However, the air connection between the different floors was not modeled.

The multi-zone building model is based on the Buildings.rooms.mixedAir model. Ten instances of this model were used for the 10 different rooms in the building. The MixedAir room model
Chapter 3. Testbed development

is a validated model [35] which assumes fully mixed aire inside the room and takes into account multiple processes such as radiation and heat transfer through glazings systems and walls in addition to shading. The doors were modeled using the operable door instance DoorDiscritizedOperable that represents a 2-way air flow model.

3.1.3 Occupants

The building is assumed to have 4 occupants. The way the occupants affect the indoor environment is through the internal heat gains from heat given off by the occupants’ body, lighting and appliances. The heat gains for lighting are the same for the whole building and are taken to be 12 W/m². The heat gains from people differ from room to room and are taken to be 8.3 W/m² in the living room, 100 W in the main bedroom and 6 W/m² in the other bedrooms [14]. The schedule of the heat gains is given in Figure A.1 in Appendix A. Moreover, the heat gains are divided into convective and radiative heat transfer. The convection coefficients are 0.5 for people and appliances and 0.6 for lighting. It should be noted here that the heat gains from the bathrooms were implemented in other rooms (since the bathrooms were not modeled separately) but the heat gains from the kitchen were not included. The heat gains were implemented using schedules which are connected to the built-in "Heat Gains" port of the MixedAir room model (Figure B.9 in Appendix B).
3.1.4 Infiltration and ventilation

Infiltration represents the unintentional flow of unconditioned outside air into the building through the cracks of doors and windows. In this case, the infiltration is considered to have a typical value of 0.2 ACH for each zone [14]. To model it, an air source with a specific air mass flow corresponding to the 0.2 ACH value and with the weather data as input, was used and it was connected to the fluid ports of the room model (Figure B.7 in Appendix B). The infiltration can be implemented based on pressure differences but it was not considered in this work for simplicity purposes. To consider it, the model "Buildings.Fluid.Sources.Outside_Cp" can be used.

Concerning the ventilation system, it was considered to be a scheduled mechanical supply/exhaust ventilation system. The variation of the ventilation rates according to the zones and the time of the day are shown in Tables A.3 and A.2 in Appendix A.

To implement the mechanical supply/exhaust system, fan models with prescribed air mass flow rates were used (Figure B.10 in Appendix B). However, since the fan model solves for the intersection of the pressure drop curve of the duct network and the fan curve, the simulation time has significantly increased. For this reason, the mechanical supply/exhaust model was dropped and the implementation of the ventilation system was done in combination with the infiltration system as follows: the ventilation rates and schedules were implemented as mass flow rates (+ for supply and - for exhaust) added together with the infiltration rate to supply the net air rate to the different rooms through the air source.

3.1.5 Solar collectors and heating

The heating system considered has a two-pipe reverse return configuration connected to a solar collectors system as shown in Figure 3.2. The heat is provided to the rooms through radiators which are sized accordingly. Each of the radiators is automatically controlled through a valve that takes into account the temperature of the room and the heating schedule depending on the occupancy to efficiently control the radiator’s mass flow rate. The circulation in the radiators circuit is provided through a pump that detects the change in pressure in the circuit to assure the efficient circulation of water. The heating circuit also contains a boiler that heats the water to the required temperature as needed.

The radiator circuit withdraws the hot water from a stratified storage tank. This tank is connected through a heat exchanger to an array of solar collectors. The solar collector circuit consists of the solar collector modules and a pump that provides the water mix circulation. This pump is turned on when the amount of irradiation is above a certain threshold and is turned off when the temperature of the water inside the tank is above a certain threshold.

The solar collector model is capable of modeling different types of solar collectors: flat plate, concentrating and tubular solar collectors. In order to protect the collectors from stagnation and overheating, it is possible to control the shading on the collectors. Given that for the simulation in this work, no effects of elevated horizons or possible environmental shading on the tilted surface is taken into account, the collectors were assumed to be completely shaded when the water mix temperature in the collectors gets above a certain critical point. The properties of the storage tank and that of the solar collector circuit are shown in Tables A.4 and A.5 in Appendix A.

The heating set-point can be user specified and is taken by default to be 21 °C whereas the cooling set-point is 25 °C. Concerning the heat losses in the circuit, they are taken into account in the tank, boilers and pumps. Also, the heat losses from the solar collector’s circuit pipes are considered as shown in Table A.4 whereas the pipes of the heating circuit are considered to be ideal.

To control the whole system, a lot of effort has been done to reduce the control system leading to the reduction of the simulation time. Different control blocks were developed specifically to reduce events and to make the transitions in the boolean values as smooth as possible.
3.1.6 Domestic hot water

The Domestic Hot Water (DHW) includes the use of the hot water for domestic use such as food preparation, sanitation and personal hygiene (showers). Usually, the modeling of the DHW demand is difficult due to the unpredictability of the demand unless a very specific case is considered. For this reason, and for simplicity purposes, it was considered that the demand for the DHW is fixed at 200 L/day. The hot water has a temperature of 60°C and the return temperature is set to be 14°C (which is similar to the temperature of tap water).

To model this system, it was first considered to have 2 outputs from the stratified tank: one for the DHW demand which withdraws water from the top of the tank (where the warmest water is found) and the second one from the middle of the tank for the heating system which requires lower temperatures. However, the stratified tank model present in the library does not allow this option.

Effort was made to alter the model into a 2-output tank but the implementation was very difficult to achieve because of the different properties of the tank model. Therefore, the implementation was considered as follows: a second circuit withdraws water from the top of the tank and the water is passed through a second boiler which heats the water to the required temperature and then this water is passed through a sub-model that removes the heat from the water until the DHW return temperature is reached (Figure B.11 in Appendix B). The connection with the solar thermal system is shown in Figure 3.2.

3.1.7 Electric model

The electrical model should consist of an electrical network representing the dwelling’s distribution system to which the PV and all power consuming equipment are connected. However, modeling different components inside the house and their energy consumption requires the availability of the electrical components library and well determined usage and occupancy profiles. For this thesis the decision was made to represent the electrical demand using average measured data varying
3.2 Complete system

Throughout the year. Moreover, the electric model does not involve the building connection to the electrical grid however the analysis done later takes into account the power used from and supplied to the grid. Also, this work does not involve the analysis of voltages and currents but it rather focuses on the power and energy produced and consumed and how they match with the demand.

The PV model used is the one from the Districts library (Section 2.2.1) which is still an unofficial part of the Buildings library. This model takes into account the power production according to the irradiation and a fixed efficiency. It does not consider the variation of the output according to the cell temperature or other factors. However, this model is good enough to approximate the electrical power produced from PV given a set of parameters.

Concerning the electrical demand data, it has a resolution of 15 minutes and it describes the average data of a Dutch house. This data is official data supplied by the Energy Data Services in the Netherlands (EDSN) [4]. The electrical network also contains a resistor representing losses in the wires and a voltage source to set the operating voltage across the PV as shown in Figure 3.3.

Using this model, the orientation, geographic location, tilt and efficiency of the PV modules can be adjusted in addition to the weather data. No effects of elevated horizons or possible environmental shading on the tilted surface of the PV module are taken into account.

3.2 Complete system

Given the different aspects considered for the testbed, the testbed model was split into an electrical model (Figure 3.3) and a thermal model. Furthermore, to implement all the thermal-related aspects, the thermal model was organized according to Figure 3.4. This model consists of 3 interconnected levels as follows:

- Level 1: This is the lowest level which contains the 10 thermal zones, the door models and the shading control.
- Level 2: This level implements the heat gain inputs as well as the infiltration and ventilation.
Chapter 3. Testbed development

Figure 3.4: Thermal model implementation levels.

- Level 3: This level connects the heating system to the building model. It also includes the weather data. From this level all the simulations can be executed without the need to go through the different model levels.

Figure 3.5: Complete thermal system implemented in Dymola.

Figure 3.5 represents level 3. In this model the building is represented by red block on the right that contains levels 1 and 2 which implement the heat gains, infiltration, ventilation and all the construction properties. The second model (left) is the heating system containing the radiators, DHW, solar collectors and the storage tank. All the important parameters of the simulation can be adjusted in this level (3). Also, the top block is the weather data reader which serves as an
input to all the other blocks. It should be noted here that mechanical cooling is not included in
the model. Future work can address this topic.

All the model components used for the development were organized into a single package containing
multiple sub-packages. The sub-packages are as follows:

- InternalHeatGains: Contains the models for the different internal heat gains considered in
  the building.
- AirInfiltration: Contains the models for the air infiltration in addition to the simplified
  ventilation system.
- HeatCoolingDemand: This package contains multiple models that help determine the heating
  and cooling demand of a building.
- ShadingControl: Contains models to control the shading devices of the building.
- HeatingSystem: This package contains the complete heating system used.
- Ventilation: This package contains the mechanical ventilation system of the building.
- Electrical: This package contains the electrical models of the testbed.
- DomesticHotWater: This package contains the DHW model.
- Levels: This package contains the different levels which use the above mentioned sub-
  packages. From this package the simulations can be executed.

Organizing the different models into sub-packages makes it very handy to access models, change
them, or add new models to the packages.

This ends the section in which we present the model properties, model implementation and the
model inputs and assumptions. In what follows, some results concerning the energy demand and
energy supply are presented.
Chapter 4

Testbed simulation results

In this chapter, a demonstration of how the testbed can be used is shown. All the results presented are for the specific case of the single family Dutch house described in the previous section. The results and the subsequent discussion consist of 3 main parts: the energy demand, the energy supply and the matching between the demand and supply. The energy demand consists of the electric power demand and the heating and cooling demand whereas the energy supply consists of the supply of electric power through the PV and the supply of heat through the solar collectors.

4.1 Energy demand

In this section the space heating and cooling demand of the single family house is presented in addition to the electric power demand.

4.1.1 Space heating and cooling demand

For the case study, a heating set-point of 21°C and a cooling set-point of 25°C were considered for the Amsterdam climate resulting in the space heating and cooling demand shown in Figure 4.1. As it can be seen, the space heating demand is high during the winter and very low during the summer whereas the cooling demand has the opposite variation. This is logical since the climate is cold with an average temperature of around 4°C in the winter and 18°C in the summer.

As shown in Table 4.1, the total space heating demand is higher than the cooling demand which is logical as well. It should be noted here that the shading was not considered for this simulation.

Table 4.1: Total annual space heating and space cooling demand for the single family house.

<table>
<thead>
<tr>
<th></th>
<th>Annual space heating demand (kWh)</th>
<th>Annual space heating demand per area (kWh/m²)</th>
<th>Annual space cooling demand (kWh)</th>
<th>Annual space cooling demand per area (kWh/m²)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>13773</td>
<td>81</td>
<td>5222</td>
<td>31</td>
</tr>
</tbody>
</table>

To validate the results, the best option would be to implement the same model in another simulation tool and compare the results. However, since this was not done, we will compare the results with standard data for Dutch houses. Considering there is a lot of variation depending on the house, occupants and construction properties, specific data related to the case studied is not available. However for a "recent" single family house with construction properties close to the ones used for the specific test case, the energy consumption is estimated to be around 84 kWh/m².
(calculation shown in Appendix A) for annual space heating. This value is very close to the results for the case study model. This gives a rough idea that the results are in the logical range but it does not confirm that the model is validated. Future development of the same case study in another simulation tool would help with the validation.

If one includes external shading devices that are closed when the direct sun radiation is above 300 W/m$^2$, the space cooling demand is reduced to 19 kWh/m$^2$ and the space heating demand is slightly increased to 85 kWh/m$^2$. These results are expected since the presence of shading reduces the heat gains from the sun which reduces the cooling demand especially for the rooms with large windows but it also increases the heating demand especially in the winter.

Concerning the different zones inside the building, it can be noticed that most of the zones that do not have a lot of contact with the outside air have low demand for heating and cooling. However, the rooms facing south have a high cooling demand in the summer due to the heat gains from the sun. For instance, the living room which faces south with a large window, requires 2927 kWh of space cooling whereas the kitchen which faces north requires 3 kWh of space cooling. Also, north facing zones have a high space heating demand during winter since they do not receive enough heat from the sun. For example the kitchen needs 223 kWh of space heating whereas the bedroom requires 34 kWh.

4.1.2 Electricity demand

As stated in Section 3.1.7, the electricity demand data used is an average with a resolution of 15 minutes. The original data is expressed in per-unit (p.u) terms and it needs to be translated into power demand. Taking into account that on average a dwelling consumes around 3500 kWh (Section 2.5) of electricity, the electric power demand can be derived and it is shown in Figure 4.3 for a typical summer and winter week. From this figure it can be seen that the winter demand is higher that of the summer. This is due to the fact that there is less need for lighting in the summer and for electricity related to heating. Figure 4.2 shows a more detailed sample of the data. It shows that the demand during the night hours is almost the same for the summer and the winter but during the day there is a higher demand in the winter.
4.2 Energy supply

In this section the amount of thermal and electrical energy supply from the renewable energy systems is shown for different parameters.

4.2.1 Solar collectors production

Concerning the solar collectors, the amount of heat produced depends on the type of the collectors, geographical location, inclination angle, orientation and size. It is considered that the location is Amsterdam, the angle is 45° and the orientation is south. In this section, the size will be fixed to 10 m$^2$. The Buildings library allows the choice among flat plate, concentrating and tubular collectors but here only flat plate and concentrating solar collectors will be considered.

In order to compare the results of the solar collectors production, one needs to compare how much thermal power was transferred to the collectors liquid and then to the storage tank. To get a clearer representation of the differences, we compared the amount of energy used (natural gas in this case) by the boilers to heat the water to the desired temperatures in 3 cases: without solar collectors which forms the base case, with flat plate collectors and with concentrating collectors.
Chapter 4. Testbed simulation results

Figure 4.4 shows the total monthly gross energy used by the DHW boiler and the space heating boiler for the 3 cases. The gross energy means the total energy released from the natural gas used in the boilers. The energy transferred to the water is this gross energy multiplied by the efficiency of the boilers. A 90% efficiency is considered since this is roughly the efficiency (it can get up to 95%) of the common condensing boilers used in the Netherlands. The figure shows that the contribution of the solar collectors is noticeable most of the months except in November, December and January. In the summer the solar collectors, especially the concentrating collectors, cover the DHW demand and the very small amount of space heating demand. Comparing the 2 types of collectors, it can be seen that the concentrating solar collectors have the higher yield (since they result in the lowest gas energy use). These results are logical because the concentrating solar cells concentrate the radiation to achieve higher heat transfer to the flowing liquid compared to the flat plate collectors.

Table 4.2 shows the annual gross energy use in the 3 cases in more detail. Here $Q_{SH}$ is the gross annual heat released by the gas in the space heating boiler, $Q_{DHW}$ is the gross annual heat released by the gas in the DHW boiler and $Q_{TOT}$ is the total of the 2 boilers. Taking into account that the natural gas from the Netherlands has a Gross Calorific Value (or Higher Heating Value) of 35.17 MJ/m$^3$ [29], the DHW demand and the space heating require 1261 m$^3$ of natural gas for the house without collectors. With flat plate collectors, the amount becomes 1020 m$^3$ and for concentrating solar collectors the consumption is reduced to 862 m$^3$ of natural gas.

Table 4.2: Energy used by the boilers according to the 3 types of solar collectors.

<table>
<thead>
<tr>
<th>Solar collector type</th>
<th>No Collector</th>
<th>Flat plate</th>
<th>Concentrating</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Q_{SH}$(kWh)</td>
<td>12007</td>
<td>9803</td>
<td>8290</td>
</tr>
<tr>
<td>$Q_{DHW}$(kWh)</td>
<td>319</td>
<td>164</td>
<td>128</td>
</tr>
<tr>
<td>$Q_{TOT}$(kWh)</td>
<td>12326</td>
<td>9967</td>
<td>8418</td>
</tr>
<tr>
<td>$f_{sav,therm}$</td>
<td>-</td>
<td>0.19</td>
<td>0.32</td>
</tr>
<tr>
<td>$V_{NaturalGas}$(m$^3$)</td>
<td>1261</td>
<td>1020</td>
<td>862</td>
</tr>
</tbody>
</table>

One indicator that helps compare the different results is the fractional thermal energy saving indicator ($f_{sav,therm}$) [34]. The formula is shown below:

$$f_{sav,therm} = 1 - \frac{\text{gross gas consumption with the solar heating system}}{\text{gross gas consumption without the solar heating system}} = 1 - \frac{Q_{TOT, Coll}}{Q_{TOT, NoColl}}$$  \hspace{1cm} (4.1)

As shown in the table, the highest fractional savings are for the concentrating solar collectors with a value of 0.32. This means that the use of 10 m$^2$ of concentrating solar collectors helps reduce the energy consumption of the boiler by 32% and subsequently the gas consumption by 32%.

Taking into account only the DHW, the fractional saving for flat plate collectors amounts to 49% whereas it amounts to 60% for the concentrating solar collectors. These figures can change if one considers detailed DHW profiles rather than a constant DHW demand. This will be discussed further in Section 4.3.

4.2.2 PV production

The electric power produced by PV’s is a result of multiple factors just as in the case for the solar collectors. The size of the PV module, the inclination angle, the orientation, the type of the PV (efficiency) and the geographical location play an important role in determining the output. In this work, the fixed and the variable parameters are the same as the solar collectors: the geographic location is Amsterdam, the orientation is south and the inclination angle is 45°. The parameters that will be changed are the efficiency of the PV panel and its size/area.
Currently in the market, different types of PV panels are available that differ in multiple aspects, most importantly in their efficiency which is mainly based on the type of solar cells used. Here, 3 types of solar panels were considered: polycrystalline silicon panels which are the most popular with efficiencies of around 12%, monocrystalline silicon panels which have higher efficiency (around 18%) but are the most expensive, and the thin film solar panels that have a lower efficiency (around 7%) but are inexpensive and flexible.

Figure 4.5: Electricity production on a monthly basis by the 3 types of solar panels.

Figure 4.5 shows the production on a monthly basis for the 3 different types of solar collectors. In this graph, the area of the collectors was fixed to 10 m$^2$ to depict the most common PV area used for single family houses. The results across the year are logical (large production in summer and low in winter) since the irradiation in the summer is higher than that in the winter. Also it can be seen that the monocrystalline panels produce the most energy over the year whereas the thin film panels produce the least.

The annual energy production of the 3 types is shown in Table 4.3. Using the monocrystalline solar panels, around 51% of the annual electricity demand can be covered by PV whereas polycrystalline panels cover 34% and thin film panels cover 20%. More on the matching between supply and demand is explained in next section. Note that the area was not changed in this section but it will be considered in the next section.

<table>
<thead>
<tr>
<th>Solar panels type</th>
<th>Thin film</th>
<th>Polycrystalline</th>
<th>Monocrystalline</th>
</tr>
</thead>
<tbody>
<tr>
<td>Annual Energy (kWh)</td>
<td>699</td>
<td>1198</td>
<td>1797</td>
</tr>
</tbody>
</table>

4.3 Supply and demand matching

Supply and demand matching is an important factor in designing Net-Zero Energy Buildings (Net ZEBs). Multiple indicators have been developed for the load matching and the interaction of Net ZEBs with the grid. We distinguish two families of indicators: the load matching indicators which are used to describe the degree of overlap between the generation and the load profiles whereas the grid interaction indicators describe the exchange of energy between a building and the grid [19]. In this work, the focus is on the load matching indicators especially two of them: the On-site Energy Fraction index ($OEF$) and the On-Site Energy Matching index ($OEM$) [33]. $OEF$ indicates the proportion of the local load covered by the on-site energy production while OEM
indicates the proportion of on-site energy generation used by the building and system rather than being dumped or exported. More specifically, the electrical $OEF_e$ ($OEF_e^c$) and $OEM_e$ ($OEM_e^c$) are considered here. The expression for each of the indicators is given below:

\[
OEF_e^c = \frac{\sum_{i=t_1}^{t_2} \min[G_{elec}(i) - ES_{on}(i) - l_e(i); L_{elec}(i)] \Delta t}{\sum_{i=t_1}^{t_2} L_{elec}(i) \Delta t}, 0 \leq OEF_e^c \leq 1 \quad (4.2)
\]

\[
OEM_e^c = \frac{\sum_{i=t_1}^{t_2} \min[G_{elec}(i) - ES_{on}(i) - l_e(i); L_{elec}(i)] \Delta t}{\sum_{i=t_1}^{t_2} G_{elec}(i) \Delta t}, 0 \leq OEM_e^c \leq 1 \quad (4.3)
\]

Where between the time-step interval $i$ and $i+\Delta t$: $G_{elec}(i)$ and $L_{elec}(i)$ are the averaged on-site generated electrical power and electrical load power, respectively; $ES_{on}(i)$ is the averaged on-site part of the electrical power sent to the electrical storage, charge in "+" sign, and discharge in "−" sign and $l_e(i)$ is the averaged electrical loss of on-site electrical power during the distribution process. In this work, electric losses are considered to be 0 and no battery is considered. Values of $OEF_e^c$ and $OEM_e^c$ close to 1 mean a higher matching.

If one looks at the demand/supply matching on an annual basis, for the case where the total energy produced by the PV is equal to the total energy consumed annually, $OEF_e^c$ and $OEM_e^c$ are equal to 1 since the time step is considered to be 365 days. To match the demand of 3500 kWh, the area of the different types of solar panels needs to be enlarged as shown in Table 4.4.

<table>
<thead>
<tr>
<th>Solar cell type</th>
<th>Thin film</th>
<th>Polycrystalline</th>
<th>Monocrystalline</th>
</tr>
</thead>
<tbody>
<tr>
<td>Annual Energy produced (kWh)</td>
<td>699</td>
<td>1198</td>
<td>1797</td>
</tr>
<tr>
<td>Area to match consumption ( m²)</td>
<td>50</td>
<td>29.3</td>
<td>19.5</td>
</tr>
<tr>
<td>Area coefficient</td>
<td>5</td>
<td>2.93</td>
<td>1.95</td>
</tr>
</tbody>
</table>

Table 4.4: Required area to match yearly production with consumption for the different types of solar cells

For the monocrystalline solar panels, the area needs to be almost doubled to 19.5 m² whereas the area should be tripled for the polycrystalline and multiplied by 5 for the thin film cells. The area of one side of the tilted roof is around 54 m². Therefore, it is possible to use any of those 3 technologies since they would fit on the roof. Here, the economies of the cell type choice become important but this topic will not be elaborated further in this work.

For the case of annual electrical zero-energy design, if one looks at more detailed data, i.e. smaller time steps, there is a mismatch between the production and the consumption of power. Figure 4.6 shows the energy mismatch on a monthly basis. This mismatch means that during the winter period the electricity is bought from the grid (in net terms) since the consumption is higher than the production. However, during summer times, when the net production is higher than the consumption, the electricity needs to be sold to the grid or it can be stored locally in batteries for future use. For the case of a time step of 1 month, $OEF_e^c = 0.72$ and $OEM_e^c = 0.70$. Ideally, if there are no losses considered and no battery as the case here, $OEF_e^c$ should equal $OEM_e^c$ but in our simulation we considered an area of 20 m² (rather than 19.5 m²) which leads to a slightly higher production than consumption. This is why $OEF_e^c$ and $OEM_e^c$ are very close but not equal.

If one looks further into daily, hourly or more detailed data, a lot of mismatch can be noticed. For instance, as shown in Figure 4.7 during the day the power demand is relatively low and this is the time where the PV’s produce the most. Taking into account a 15 minutes time-step, $OEF_e^c = 0.39$ and $OEM_e^c = 0.38$. This shows that the more detailed the study becomes, the more noticeable is the mismatch between production and consumption.

The results show that there is a continuous exchange between the PV and the grid to fulfill the demand. This aspect of Net ZEBs is very important since it would have an impact on the grid.
4.3. Supply and demand matching

Figure 4.6: Electricity production for a 20 m² of monocrystalline solar panels and the consumption for a Dutch household. There is a zero net energy consumption on an annual basis.

Performance with high levels of penetration. For instance, a low OEF and OEM for multiple houses at the same instance can contribute in increasing peak loads which require additional generation and transmission capacity from the utilities [19]. For this reason, multiple control techniques, demand side management measures and decision algorithms are used to increase the value of those indicators.

Figure 4.7: Mismatch between production and consumption on a daily basis.

Another important factor that needs to be noted is the mismatch between the weather data resolution and the electricity demand resolution. The weather data used is the Typical Meteorological Year 3 (TMY3) data which is based on a 1 hour time step. This leads to a less accurate representation of the PV production and makes the production look smoother than it actually is [33]. In reality, the PV production varies considerably during one hour due to passing clouds or other factors. The errors in using a 1 hour resolution for the weather and a high resolution for the electrical demand (1 minute) can get up to 60% in some cases compared to a 1-min generation resolution [33]. However, considering the annual energy from PV, it differs by around 3.5 % [17]
which is relatively low but the effect can be large if one is interested in the import/export schemes of electricity and the functioning of any storage device [17].

Concerning the solar collectors, the control of the system and the presence of the storage tank lead to a heating system that is controlled according to the demand. However, the solar collectors can’t provide the complete demand of thermal power around the year. One way of trying to match demand and supply is through demand side management strategies. For instance, the washing machine which requires a temperature of let’s say 60°C, can be turned on during the peak solar radiation to reduce the use of heating power (also concerns regarding the electric power need to be considered). Also, the simulation of actual DHW demand profiles would give a clearer idea about when the demand for hot water is high and allow for more detailed matching with the supply. The matching of the thermal demand is easier in this case because the storage tank is considered. If one wants to make an analogy with the electric demand/supply matching, the storage tank is equivalent to a battery which would provide better matching.

If one tries to increase the area of the collectors in order to increase the fractional savings as much as possible, the size of the tank needs to be increased as well to increase the amount of heat that can be stored and therefore prevent overheating of the water. In this case, the area of the solar collectors was doubled to 20 m² and the volume of the tank also doubled to 1.5 m³ to keep a constant ratio. Here only the concentrating solar collectors are considered since they are the most efficient. This leads to an increase in the fractional savings to around 0.4. It can be seen that the increase in the fractional saving (production from the collectors) is not proportional to the size as is the case for the solar modules. After this point, the increase in the size would give very little increase in the fractional savings.

The mismatch between the demand and supply can be investigated further when actual DHW profiles are considered. If DHW profiles are present, the difference between a 1-hour energy resolution and a 1-minute resolution is about 0.5% which is very low [17]. Concerning the use of hourly data instead of more detailed data for the weather, the energy produced by the solar collectors can vary by 3% [17], with the hourly simulation predicting slightly less solar output. However, those figures may not be the same if the system has less water storage capacity such as instantaneous hot water heating [17]; if less water storage is considered, a large amount of the heat produced needs to be consumed directly since it is not stored, affecting the consumption/production matching and therefore the data resolution would have a larger impact.

As a conclusion, the zero energy building design can be achieved in terms of electricity. The size of the PV can be changed to increase the production accordingly and the demand can be reduced by the use of demand side management strategies and energy efficient appliances. Moreover, the thermal zero energy building design can be achieved if one considers in addition to the solar collectors, energy efficient measures to reduce the heating demand inside the building such as higher insulation, optimization of heat gains from the sun and the use of triple glazed windows. Also, other technologies can be used depending on their availability such as heat pumps for the space heating and Combined Heat and Power (CHP) installations to provide heat in addition to electric power. Those measures and technologies were not simulated in this work but they can be implemented in the future as part of the improvement of the testbed.
Chapter 5

Simulation tools discussion

In this section we will share our experience in using Dymola and Modelica for modeling and simulation and how they compare to other simulation tools. All the experience was acquired during the development of this project without any prior experience.

The first months of this project were spent learning the tool and getting familiar with the language. It should be noted that Dymola’s user manual is well developed and explained and it is very helpful for beginners. Moreover, concerning the buildings library, the user guide is helpful as well and well documented. We can say that the learning curve for Dymola and Modelica is relatively steep. On a more detailed level, it is a bit difficult to learn and to get used to how to connect the different rooms in the Buildings library to each other especially since there is no 3D view of what is being done. However, after getting the idea of how everything works, it becomes easy to manipulate the rooms. Another drawback is the lack of error messages and warnings regarding the right proportions of common walls, sizes and all the geometric properties. For instance, if one declares a room with a volume of 5 m$^3$ that has walls and floor of area of 10 m$^2$, no error messages or warnings are issued. Work must be done in this domain to reduce the development time and to make sure the geometries match. One work-around would be to import the building design from some drawing tools such as Google Sketchup.

Concerning the ease of use of Dymola, we can say that the interface is user-friendly, simple, and efficient. The feature of sub-model reuse and the hierarchical nature were a great advantage during the development. It is very easy to create new models from existing ones, fine tune them to meet the specific application and then run the simulation. Moreover, the ability to group different models into sub-models is very useful; Dymola has a right-click function that automatically groups multiple blocks into one block/model which can be saved in the library for future use. Also, the capability of declaring user specified parameters enhances the flexibility. By doing so, parameters for the different hierarchical models can be propagated to the top-level and their value can be changed by changing the value at this top-level. Moreover, the “drag and drop property” enhances the usability and models can be arranged into packages which contain different sub-packages facilitating the access to the models according to their domain and properties.

Another key point is the direct access to the Modelica code. The code of any model can be accessed by clicking on the scripting editor which allows the user to do virtually anything with the models (Figure B.3 in Appendix B). Direct and easy connections can be created in the diagram tab whereas a bit more complex tasks can be programmed directly in the code tab. This makes Dymola extremely flexible in manipulating components and models. For instance, if conditional connections are required, an if-statement for the connect statement is sufficient.

A third point is the documentation tab and the simple method for documenting your work. The user is allowed to insert pictures, comments, graphics and all types of content that can be used to document the models. This tab allows the user to document and explain all the parameters,
Chapter 5. Simulation tools discussion

the equations and the function of the model. A good practice would be to always document the models in as much detail as possible (Figure B.2 in Appendix B).

Concerning the post-processing capabilities, Dymola provides a good interface for plotting or for acquiring the data in tables which can be exported to other post-processing tools. Any variable used can be plotted and all the values can be reported. The time axis can be easily toggled between seconds, minutes and days which can help in changing the resolution of the graph. Additionally, the user has the ability to zoom in on a specific area of the plot. However, the results interface is limited and enhancements to allow for more direct manipulation of the data need to be added. For instance, integrating functions or the ability to divide the year into months, and some other useful functions need to be available for post-processing.

LBNL has developed a Python package called BuildingSpy [2] for post-processing results based on the Buildings library. This package is useful especially since it allows mapping the variables to their corresponding values. Dymola actually creates *.mat* files which can be read in MATLAB but it is difficult to know which parameter is which and that is mainly what BuildingSpy does: it maps the values to their corresponding variables and provides a couple of useful post-processing functions.

Concerning the simulation time, multiple points can be made. A long simulation time was noticed in accordance with what was mentioned in Section 2.3. The time depends on the required amount of detail in the models. Less detail leads to a shorter simulation time. During the development of our model, it was noticed that the control systems increase the simulation time. Also, the use of conditional statements leads to the creation of events which increases the calculation time. To avoid events, multiple methods can be found in the best-practice section of the Buildings library website [5]. We can say here that the IDEAS (Section 2.2.1) library development team has successfully avoided events in the development of their library.

In this specific application, the ventilation model was changed from the implementation using a mechanical system (fans) to an implementation in combination with the infiltration rates as an air source. This led to a reduction in the simulation time due to the fact that the fan model solves for the intersection of the pressure drop curve of the duct network and the fan curve which is time consuming. Also, the use of constant convective coefficients on the inside and outside of room walls leads to a reduction of the computational time. The linearization of the radiation network also contributes to a faster simulation. Moreover, when shading is taken into account, the simulation time increases due to the different calculations related to the irradiation in the MixedAir room model of the Buildings library and we believe that this issue needs to be addressed.

One topic that was time demanding during development is the calculation of the space heating and cooling demand of the building for which multiple methods were considered. The main factor in determining which method to use was the simulation time since the calculation is being done for 10 rooms. 3 methods were considered:

- 2 PID controllers, one for the heating set-point and one for the cooling set-point. This method proved to be very time consuming (Figure B.4 in Appendix B).
- Demand calculation using conditional statement (if-else) with the avoidance of events. It performs well if it is only implemented for one room but it is time consuming when used for the 10 rooms in our project (Figure B.5 in Appendix B).
- A constant temperature source connected to the convective air-port of the room. In this way, the room is automatically kept at a constant temperature. By connecting a heat sensor, the heat flow can be calculated. However, this method requires 2 simulations: the first one is for the heating set point and the second one is for the cooling set point. This method proved to be the fastest (Figure B.6 in Appendix B).

Another factor that needs to be considered is the solver and its tolerance which have an effect on the simulation time [25]. Dymola provides the capability of choosing between multiple solvers.
depending on the application required. The solvers vary in terms of single and multiple step sizes, ability to solve stiff problems and the order of differential equations that they can handle. By default, the Dymola 2014 version uses DASSL as the solver with a tolerance of 1E-6, but for the Buildings library, the suggested solver is Radau with a tolerance of around 1E-6 [5]. In our case, we used the solver Esdrik34a with a tolerance of 0.001 which provided the fastest simulation time among the solvers considered without a considerable change in the results as compared to the Radau solver.

As a summary, Table 5.1 has been developed to compare Dymola to TRNSYS, ESP-r and Simulink. Concerning Dymola, the evaluation is based on our own experience with the tool whereas for the other tools it is based on the experience of other students and researchers at TU/e. This table takes into account a general overview of different properties of the tools applied to building simulation. It should be noted that some properties of the tools overlap with the programming/modelling language used but no distinction is made in the table. In the following we will summarize the table’s findings.

First, Dymola is the most expensive among the 4 simulation tools and ESP-r is the cheapest since it is an open-source tool. Moreover, for a complete case study of a house with the goal of determining space heating and cooling demand using a tool with which the user has prior experience, the development time is almost the same for all the tools except for Simulink which is slower. The main time consuming part for Modelica if the Buildings library is used is the creation and connecting of the zones. The simulation time for the same case is the highest for Dymola.

Concerning the handling of the tools, Dymola has advantages over the other tools since it allows a lot of flexibility, reusability, ease of documentation, scripting and graphical modeling. It also has a great advantage in representing the physical systems as they actually are in real life. However, Dymola lacks the post-processing capabilities of Simulink. The simulation capabilities are more or less the same with good coupling features for all the tools. Concerning the libraries, most of them require some time to be learned but after that their usage becomes straightforward. A lot of open source libraries related to buildings applications are available in Modelica and they are compatible with Dymola. Similarly, TRNSYS is limited in terms of building-related libraries since they are not openly available. Moreover, Simulink and TRNSYS provide well validated libraries as compared to the libraries of ESP-r and Dymola where many models have yet to be validated.

In addition to that, the Dymola related libraries cover multiple aspects such as mechanics, controls thermodynamics and air flow. Surely, Simulink provides an advantage in control but it lacks building components and geometry related libraries. Also, ESP-r has limitations in hydraulics and electronics.

TRNSYS and ESP-r are well developed for air flow modeling and temperature stratification of air. In this domain, Modelica provides some limitations; the Buildings library for instance does not provide air temperature stratification modeling. This feature is present in some other Modelica libraries but it is not very well developed. One big advantage of Dymola is the availability of libraries that provide a detailed analysis of interconnected thermal and electrical components of a building such as the Green Buildings library and the IDEAS library. Finally, model calibration, optimization, management and export is possible in Dymola and Simulink but it is limited in the other two tools.

As a summary, Dymola and the Modelica language provide a lot of advantages for building-related simulations: flexibility, reusability, ease of use and natural representation of systems. Some limitations are present in terms of controls, simulation times and 3D modeling of buildings which need to be improved. If one needs to get the optimal testbed, a logical way of thinking is to combine the advantages of the different tools together by coupling them. For instance, Modelica can be used to model the systems inside the building, Simulink for control and post processing, and the other tools for the physical building properties.
Table 5.1: Comparison of the building-related simulation tools

<table>
<thead>
<tr>
<th>Properties</th>
<th>Dymola</th>
<th>Simulink</th>
<th>ESP-r</th>
<th>TRNSYS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Programming language</td>
<td>Modelica</td>
<td>MATLAB</td>
<td>Fortran</td>
<td>C/Fortran</td>
</tr>
<tr>
<td>Developer</td>
<td>Dassault System</td>
<td>Mathworks</td>
<td>UoW</td>
<td>Int</td>
</tr>
<tr>
<td>Cost-Commercial</td>
<td>0</td>
<td>+</td>
<td>++</td>
<td>+</td>
</tr>
<tr>
<td>Cost-Academic</td>
<td>+</td>
<td>+</td>
<td>++</td>
<td>+</td>
</tr>
<tr>
<td>Simulation time</td>
<td>0</td>
<td>+</td>
<td>++</td>
<td>+</td>
</tr>
<tr>
<td>Development time</td>
<td>+</td>
<td>0</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Scripting editor</td>
<td>++</td>
<td>++</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Graphical editor</td>
<td>++</td>
<td>0</td>
<td>0</td>
<td>+</td>
</tr>
<tr>
<td>Symbolic manipulation</td>
<td>++</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Hierarchical modeling</td>
<td>++</td>
<td>+</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Reusability of models</td>
<td>++</td>
<td>0</td>
<td>0</td>
<td>+</td>
</tr>
<tr>
<td>Post processing capabilities</td>
<td>0</td>
<td>++</td>
<td>+</td>
<td>0</td>
</tr>
<tr>
<td>Model documentation</td>
<td>++</td>
<td>+</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Software documentation</td>
<td>+</td>
<td>++</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Natural representation</td>
<td>++</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Continuous time systems</td>
<td>++</td>
<td>++</td>
<td>++</td>
<td>++</td>
</tr>
<tr>
<td>Discrete time systems</td>
<td>++</td>
<td>++</td>
<td>++</td>
<td>++</td>
</tr>
<tr>
<td>Debugging facilities</td>
<td>+</td>
<td>+</td>
<td>0</td>
<td>++</td>
</tr>
<tr>
<td>Event logging</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Coupling features (BCVTB/FMI)</td>
<td>++</td>
<td>++</td>
<td>++</td>
<td>++</td>
</tr>
<tr>
<td>Error descriptions</td>
<td>+</td>
<td>0</td>
<td>0</td>
<td>++</td>
</tr>
<tr>
<td>Ease of use</td>
<td>++</td>
<td>+</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>Open source libraries</td>
<td>++</td>
<td>+</td>
<td>+</td>
<td>0</td>
</tr>
<tr>
<td>Validated libraries</td>
<td>+</td>
<td>++</td>
<td>+</td>
<td>++</td>
</tr>
<tr>
<td>Mechanics</td>
<td>++</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Controls</td>
<td>+</td>
<td>++</td>
<td>+</td>
<td>++</td>
</tr>
<tr>
<td>Thermodynamics</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>++</td>
</tr>
<tr>
<td>Hydraulics</td>
<td>++</td>
<td>+</td>
<td>0</td>
<td>++</td>
</tr>
<tr>
<td>Electronics</td>
<td>+</td>
<td>++</td>
<td>0</td>
<td>+</td>
</tr>
<tr>
<td>Buildings</td>
<td>+</td>
<td>0</td>
<td>++</td>
<td>+</td>
</tr>
<tr>
<td>Air flow</td>
<td>+</td>
<td>++</td>
<td>0</td>
<td>++</td>
</tr>
<tr>
<td>Model calibration</td>
<td>+</td>
<td>++</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Design optimization</td>
<td>+</td>
<td>++</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Model management</td>
<td>++</td>
<td>+</td>
<td>0</td>
<td>+</td>
</tr>
<tr>
<td>Code and model export</td>
<td>++</td>
<td>++</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

1 University of Wisconsin-Madison
2 International world-wide development community and its distribution is managed under GitHub source code control

Note
- means that the feature or the capability is not present.
0 means that the feature or the capability is present but it is not good.
+ means that the feature or the capability is present and relatively well developed.
++ means that feature the feature or the capability is very well developed.
Chapter 6

Conclusions and recommendations

6.1 Conclusions

From the development of the testbed, multiple conclusions concerning the testbed, the simulation tools and Modelica can be drawn.

Concerning the testbed:

- This testbed can used for multiple purposes: feasibility of renewable energy sources mounted on the buildings, sizing of electrical and thermal components and design of low/zero energy buildings since it covers all the important aspects of energy interaction inside the building.

- The testbed is flexible enough to serve as a basis for future testing and simulations of different technologies. The capability of grouping the different models into packages and the capability of propagating the parameters to top levels helps in enhancing this flexibility. This version of the testbed helps in changing the important parameters of the building, the heating system, the solar collectors and the PV's in a very simple way despite being implemented on multiple levels.

- The testbed also serves in designing and testing complete heating systems and other components for specific cases.

- The results produced by this testbed are in the range of the reference cases, however more validation is required for a robust model.

Concerning Dymola and Modelica:

- Dymola and Modelica provide a high degree of flexibility which allows models to be used in different setups. The hierarchical properties and the ease of coding and documentation accessibility enhance this property.

- Dymola is a very useful and easy-to-use tool for simulating Modelica based libraries. However, improvements should be made in terms of post-processing capabilities and debugging tools. Also, the price compared to other tools is very high.

- The Buildings library used for the thermal model in addition to the Districts library used for the electrical model are enough for the development of the test-bed. However, one disadvantage is that those libraries are still independent of each other. LBNL’s aim is to connect them which would provide a better picture of the complete interactive building components.
• The natural representation of physical models provides an easy way to understand the interaction among the different components of a model which gives Dymola/Modelica an advantage compared to other tools.

• The Buildings library is very well developed and has good support from the development community. The components of this library are more directed to detailed, specific applications in buildings rather than more general ones. This leads to slow simulation times if one does not try to minimize less important equations. Effort must be made to address errors and warnings concerning the dimensions and the geometric properties of rooms and buildings in order to reduce the simulation time. Also, 3D visualisation of zones would be a great improvement.

• For faster but less detailed simulations, other libraries such as IDEAS may provide better capabilities. However, this library is still in the development phase.

6.2 Recommendations

For future work, effort can be made to enhance the testbed developed during this thesis. The following are some recommendations on how this can be done.

• Validation of the building model by implementing it in another simulation tool and comparing the results.

• Enhancement of the DHW model by including detailed load profiles rather than a constant load.

• Enhancement of the heating system model by including losses due to piping in the space heating circuit.

• Implement a cooling system for the building.

• Implement heat recovery for the ventilation system. This means that the implementation of the ventilation needs to be done through mechanical representation.

• Increase the flexibility by allowing the radiators to be replaced easily by other types of heating schemes such as floor heating. Also, the capability of using Heat pumps and Combined Heat and Power (CHP) models is useful especially when assessing low/zero energy buildings.

• Develop occupancy schedules that automatically set the schedules for space heating according to occupancy by rooms rather than occupancy of the whole building, internal heat gains and electric power demand.

• Work on reducing the simulation time by enhancing the controls of the heating system. This can include programming the control in MATLAB-Simulink and then connecting it to the model through the BCVTB (Section 2.2.1).

• Reduce the number of thermal zones modeled in the building to reduce the simulation time. This can be done by merging multiple rooms into one large room. This is applicable especially if one is not interested in the detailed variation in temperature and air flow in specific rooms.

• Develop testing codes that provide both cost and technical results to manufacturers and people interested in the specific renewable energy technologies. For instance, on the technical part, sensitivity analysis codes concerning the tilt angle, the area, the efficiency, the orientation and the geographic location of the BIPVT can be developed. Also, the size of the storage tank and the size of the radiators can be investigated.
Bibliography


Appendix A

Simulation data

Calculation of space heating demand for a reference building

From the internal documents of the Building Performance simulation team at TU/e, the following data can be accessed. For a "Recent" detached single family house, the annual gross gas consumption is 2093 m$^3$ of which 79% is for space heating. Taking into account that 1 m$^3$ of gas has a HHV of 35.17 MJ and that the boilers have an efficiency of 90%, then the space heating demand would be 14534 kWh.

Input parameters

Table A.1: Thermal properties of the detached single family house used for the case study [37].

<table>
<thead>
<tr>
<th></th>
<th>$R_{value}$ (m².K/W)</th>
<th>$U_{value}$ (W/m².K)</th>
<th>$K$ (W/m.K)</th>
<th>$\rho$ (Kg/m³)</th>
<th>$C_p$ (J/kg.K)</th>
</tr>
</thead>
<tbody>
<tr>
<td>External wall</td>
<td>3.5</td>
<td>-</td>
<td>0.23</td>
<td>1000</td>
<td>840</td>
</tr>
<tr>
<td>Roof</td>
<td>-</td>
<td>-</td>
<td>0.12</td>
<td>300</td>
<td>840</td>
</tr>
<tr>
<td>Ground Floor</td>
<td>3.5</td>
<td>-</td>
<td>0.23</td>
<td>1000</td>
<td>840</td>
</tr>
<tr>
<td>Internal walls</td>
<td>-</td>
<td>-</td>
<td>0.17</td>
<td>500</td>
<td>840</td>
</tr>
<tr>
<td>Windows(Double glazing)</td>
<td>-</td>
<td>1.65</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table A.2: Ventilation flow rate for each room [37].

<table>
<thead>
<tr>
<th>Room</th>
<th>$A_{floor}$ (m²)</th>
<th>$V$ (m³)</th>
<th>supply (dm³/s)</th>
<th>Exhaust (dm³/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Living room</td>
<td>26.5</td>
<td>69</td>
<td>24</td>
<td>-</td>
</tr>
<tr>
<td>Kitchen</td>
<td>18</td>
<td>47</td>
<td>8</td>
<td>43</td>
</tr>
<tr>
<td>Master Bedroom</td>
<td>17</td>
<td>44</td>
<td>15</td>
<td>-</td>
</tr>
<tr>
<td>Bedroom 1</td>
<td>11</td>
<td>29</td>
<td>10</td>
<td>-</td>
</tr>
<tr>
<td>Bedroom 2</td>
<td>15</td>
<td>39</td>
<td>14</td>
<td>-</td>
</tr>
<tr>
<td>Bedroom 3</td>
<td>8</td>
<td>51</td>
<td>7</td>
<td>-</td>
</tr>
<tr>
<td>Attic</td>
<td>12</td>
<td>73</td>
<td>-</td>
<td>7</td>
</tr>
<tr>
<td>Bathroom,toilet and stairs</td>
<td>-</td>
<td>88</td>
<td>28</td>
<td>-</td>
</tr>
<tr>
<td>Total</td>
<td>-</td>
<td>-</td>
<td>78</td>
<td>78</td>
</tr>
</tbody>
</table>
Table A.3: Daily ventilation schedule [14].

<table>
<thead>
<tr>
<th></th>
<th>High (100%)</th>
<th>Medium (75%)</th>
<th>Low (30%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>6:00-11:00</td>
<td>0:00-6:00</td>
<td>11:00-14:00</td>
</tr>
<tr>
<td></td>
<td>14:00-23:00</td>
<td>23:00-24:00</td>
<td></td>
</tr>
</tbody>
</table>

Table A.4: Collector loop properties [34].

<table>
<thead>
<tr>
<th>Collector loop</th>
<th>Length, tank to collector</th>
<th>Diameter</th>
<th>Insulation thickness</th>
<th>Insulation thermal conductivity</th>
<th>Heat transfer media</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>15 m</td>
<td>0.02 m</td>
<td>0.02 m</td>
<td>0.042 W/mK</td>
<td>Glycol (40%)/Water</td>
</tr>
</tbody>
</table>

Table A.5: Storage tank properties [34].

<table>
<thead>
<tr>
<th>Storage Tank</th>
<th>Total volume</th>
<th>Height</th>
<th>Diameter</th>
<th>Insulation thickness</th>
<th>Thermal conductivity of insulation material</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.75 m³</td>
<td>1.67 m</td>
<td>0.76 m</td>
<td>0.15 m</td>
<td>0.042 W/mK</td>
</tr>
</tbody>
</table>

Figure A.1: Heat gains inside the house [13].
Appendix B

Models implementation in Dymola

Interface

The following figures show the diagram tab, the information tab and the Modelica text tab in the interface of Dymola. They show how the user can click on one of the tabs on the top of the page to toggle between the modes. Any change in the Diagram tab is automatically translated in the Modelica text tab and vice-versa.

Figure B.1: Diagram tab of the Modelica model in Dymola allowing the graphical development of the model.
**Figure B.2:** The documentation tab of the same model as the previous Figure. Here the user can explain the model through text and illustrations. For instance, it can be seen how the picture of the building elevations and floor plans is present in addition to other explanations.

**Figure B.3:** The Modelica text tab which allows the direct access to the Modelica code for the same model as the previous Figure. Here the user can insert any Modelica coding text in order to develop the model.
Models implementation

Space heating and cooling demand

Figure B.4 shows the implementation of the heating logic using two PID controllers. The temperature sensor "TRooAir" detects the room temperature which and is used as the reference signal for the 2 PID controllers. One of the controllers compares this temperature to the cooling set-point "TSetCoo" and the other one to the heating set-point "TSetHea".

If the temperature is between the heating and cooling set-points, no action is taken. Otherwise, if it is higher than the cooling set-point or lower than the heating set-point, the difference between the temperatures (TRooAir-Tset) is processed to get the amount of heat required and then used as an input to the heat source "preHea" which provides heat if heating is required and withdraws heat if cooling is required. The "Hea_port" is connected to the air heat_port of the specific room to provide/writhdraw heat. This logic is repeated 10 times for the 10 thermal zones.

Also, two integrators are present to calculate the heating and cooling energy. Those integrators usually are time consuming and an alternative to that would be to post-process the data in BuildingSpy to get the energy through integration.

![Figure B.4: Implementation of the space heating/cooling demand calculation using 2 PID controllers.](image-url)
Appendix B

Figure B.5 shows the implementation of the heating/cooling logic by comparing the temperature with the heating and cooling set-points, the same way as in Figure B.4, but this time using coding rather than standard blocks. This code first initializes "Q_flow" to 0 for all the zones and then compares the temperature to calculate the amount of heat required to be added or subtracted for each room. The implementation is in a for-loop to calculate for the different thermal zones. The heat and cold required is assigned to the "Q_flow" variable of the heat port "port_b1[i]" which is connected to the air heat port of room "i".

```plaintext
model HeatCondMultiZone "Model to determine the heating and cooling"

parameter Modelica.SIunits.Temperature RoomSetPoint = 21.57315;
parameter Modelica.SIunits.Temperature RoomcoolingSetpoint = 16.27315;

// Modelica.SIunits.Volume Vol104""Room volume"
// Modelica.SIunits.HeatFlowRate HeatLoad;
// Modelica.SIunits.HeatFlowRate CoolLoad;

parameter Integer nZones=1 "Number of Zones";
Modelica.SIunits.HeatFlowRate[nZones] HeatLoad;
Modelica.SIunits.HeatFlowRate[nZones] CoolLoad;

initial equation
for  i in 1:nZones loop
    port_M1(i).Q_flow = 0;
end for;

equation
for  i in 1:nZones loop
    HeatLoad(i) = if nEvent(RoomSetPoint - TRoomAir[i] > 0) then -1012*1.204*Vol104*(HeatSetPoint - TRoomAir[i]) else 0;
    CoolLoad(i) = if nEvent(RoomSetPoint - TRoomAir[i] < 0) then -1012*1.204*Vol104*(CoolSetPoint - TRoomAir[i]) else 0;
    //HeatLoad(i) = if nEvent(HeatSetPoint - TRoomAir[i] > 0) then -100000*(HeatSetPoint - TRoomAir[i]) else 0;
    //CoolLoad(i) = if nEvent(CoolSetPoint - TRoomAir[i] < 0) then -100000*(CoolSetPoint - TRoomAir[i]) else 0;
    port_b1[i].Q_flow = (HeatLoad(i) + CoolLoad(i));
end for;
end HeatCondMultiZone;
```

Figure B.5: Implementation of the if-else statement logic.
Appendix B

Figure B.6 shows the implementation of the heating/cooling logic through the use of a fixed temperature source. The temperature "fixedTemp" is connected directly to the heat port "port_NF0" which is connected to the air heat_port of room NF0. This source makes sure that the temperature is constant at the value of the set-point. By inserting a heat sensor between the source and the heat_port, the heat flow required to keep the temperature constant can be measured and then the energy required is calculated using an integrator. This logic is repeated for all the rooms. This implementation requires 2 simulations, one for heating and one for cooling.

![Figure B.6](image1.png)

**Figure B.6:** Implementation of the constant temperature source logic.

**Infiltration**

Figure B.7 shows the implementation of the infiltration. The weather data bus (the yellow line) is connected to an air source "souInf" which is then connected to the port "air_source". This port is then connected to the room fluid port (not shown). Also, the room fluid port is connected to the AirSink port. This port is connected to an AirSink model which is an ideal flow source that produces prescribed mass flow determined by the input connected to its "m" connector. As it can be seen, the density is multiplied by the Infiltration rate. In addition to that, the ventilation rates can be inserted here through the connector "u1". In this way, the outside air enters the room from "air_source" and exits through "AirSink" depending on the flow rates given.

![Figure B.7](image2.png)

**Figure B.7:** The implementation of the infiltration logic.
Thermal zone model

Figure B.8: Implementation of a thermal zone and its connection to other blocks.

Figure B.8 shows the implementation of 2 thermal zones with the connection between them. Each of the square blocks is the MixedAir room model representing one room. Let’s look at the upper room which is "NorthRoomF0". It has as ports q, u, fluid, air, ra...(radiation), bo...(boundary), sur...(surface) and the yellow port which is used for the weather. The "q" port is used for the internal heat gains. Here, the input is not shown but the port is propagated to become an input for the whole model. The propagated input is the large triangle "qGai_flow_NF0" to which heat gain schedules will be connected. The "u" port is the shading control port. It can be seen that 2 shading control blocks are connected to it, each of them for a window. The fluid port is used for the fluid connections which, in this case, is the air. The door model "doorNMF0" is connected from one side to one room through the fluid ports and from the other side to the other room through its fluid ports. This door model is an operable model which can be controlled using the user-specified "DoorStat" parameter. When open, the air flows (2-way possible) through the whole opening of the door and when it is closed, the air flows through the cracks. It should be noted here that this model takes into account the air flow and not the heat transfer and the thermal properties. The later can be done in the room model itself. Also, the fluid ports are propagated through the ports "portsNF0" in order to make a connection with the infiltration and the ventilation systems. The other ports (the red ones) are for the thermal connections and they are connected to the ports of the other rooms such as the second room shown in the figure.
Heat gains

The implementation of the heat gains is shown in Figure B.9. The heat gains are divided between appliances, lighting and persons. The white/yellow blocks on the left contain user specified schedules. This data is then processed according to the user specified heat gains from lights and people in addition to the room area. Also the gains are divided into convective, radiative and latent heat gains (taken to be 0) and then connected to the room model through a multiplexer. Also the model calculates the total energy from the heat gains.

![Figure B.9: The implementation of the heat gains.](image)

Mechanical ventilation

The mechanical ventilation is implemented in Figure B.10. One fan is used for the supply "fanIn"
and another one for the exhaust "fanO". "port_b1" and "port_b2" are connected to the room's fluid ports whereas "port_a1" and "port_a2" are connected to the outside air through an air source connected to the weather data (not shown). The fans are controlled according to the ventilation schedules ("VentSched" and "VentSched1") and the specified mass flow rates ("mass-flow" and "massflow1").

**DHW**

The DHW implementation is shown in Figure B.11. The water comes in from the tank through "port_a1", passes through the heat exchanger "heaDyn" which removes the heat from the water until it reaches the required temperature and then the water leaves back to the tank through "port_b1". The heat exchanger is controlled by a controller that senses the temperature of the output water and compares it to the required one.

![Figure B.11: DHW demand implementation.](image)