A FORMALIZED URBAN PROSUMER MODEL: SUPPORT OF AUTOMATED SIMULATION AND DESIGN OPTIMIZATION

A Dissertation
Presented to
The Academic Faculty

by

YUN JOON JUNG

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Architecture

Georgia Institute of Technology
August 2021

A FORMALIZED URBAN PROSUMER MODEL: SUPPORT OF AUTOMATED SIMULATION AND DESIGN OPTIMIZATION

Approved by:

Dr. Dennis Shelden, Advisor
School of Architecture
Rensselaer Polytechnic Institute

Dr. Tarek Rakha
School of Architecture
Georgia Institute of Technology

Dr. Santiago Grijalva
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Ralph Muehleisen
Energy Systems Division
Argonne National Laboratory

Dr. Cheol-Soo Park
Department of Architecture and Architectural Engineering
Seoul National University

Date Approved: July 8th, 2021

Dedicated to my parents, Su Jeong Lee, and Prof. Godfried Augenbroe

ACKNOWLEDGEMENTS

This thesis can't be finished without a heavy debt of gratitude to many persons. During the Ph.D. study period, I have been privileged to work closely with many talented faculty.

The first and foremost gratitude is owed to Prof. Godfried Aguenbroe. I was very fortunate to be able to learn from him for the last five years. Without his invaluable support and guidance, the realization of this thesis work undoubtedly could not have been possible. I am also profoundly grateful to him for giving me the freedom to pursue various research interests during my Ph.D. study. However, it was extremely saddening news to hear about his passing in May 2021. I sincerely appreciate his advising and will always cherish all the knowledge, life lessons, great memories on my mind. I will also miss the in-depth discussions with him, and he will always be remembered and greatly missed. His philosophy, vision, and knowledge are of great importance to the building technology field, and I promise that I will continue to follow his path to greatness. I will always respect him, and all my condolences are with him. Rest in peace, Prof. Godfried Augenbroe.

I also sincerely appreciate Professor Dennis Shelden, who also co-advised me throughout the Ph.D. study and gave me the opportunity to learn BIM, programming, data modeling, and academic research. Furthermore, I learned/enjoyed a lot from our gbXML, BIM-BEM-BCS, and Perkins and Will's projects. I was able to expand my academic view by also belonging to the Digital Building Lab.

I can't omit and can't appreciate enough to the committee members. Professor Tarek Rakha gave me meaningful feedback and helped me finish the Ph.D. study after Professor Augenbroe's passing. Professor Santiago Grijalva taught the power system analysis and control and electricity markets courses. Professor Grijalva always answered my questions and helped me understand how building-to-grid can be integrated together. Dr. Ralph Muehleisen helped develop the last case study (power outage scenario) and gave me helpful feedback on this thesis. Professor Cheol-Soo Park also pointed out important aspects of the doctoral thesis and gave feedback on the calibration process.

I would like to thank my parents and Su Jeong Lee, who helped me focus on my study and root for me. Finishing Ph.D. study wouldn't have been possible without their supports. I sincerely appreciate it.

Last but not least, I thank the readers for taking the time to review the research.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS **iv**

LIST OF TABLES **viii**

LIST OF FIGURES **x**

LIST OF SYMBOLS AND ABBREVIATIONS **xiv**

SUMMARY **xvi**

CHAPTER 1. Introduction **1**
1.1 Background **1**
1.2 Problem Statement **3**
1.3 Literature Review **6**
1.3.1 Physics-based Energy Simulation of the Building Stock 6
1.3.2 Controls of Building-integrated Microgrids 13
1.4 Importance of an Urban Prosumer Schema in Urban Energy Simulation **17**
1.5 Research Hypotheses **18**
1.6 Organization of the Dissertation **20**

CHAPTER 2. DEVELOPMENT OF AN URBAN PROSUMER SCHEMA **21**
2.1 Background **21**
2.1.1 Metadata in the AEC Industry 21
2.2 Basic Outline of the Urban Prosumer Schema **26**
2.3 Physical Layer **30**
2.3.1 Hierarchies in the Physical Layer Element 31
2.3.2 Relationships in the Physical Layer 32
2.4 Logical Layer **36**
2.4.1 Hierarchies in the Logical Layer Element 36
2.4.2 Relationships in the Logical Layer 52
2.5 Application of the UP Schema **54**

CHAPTER 3. PHYSICS-BASED MODELS EMBEDDED IN THE SIMULATION PLATFORM **59**
3.1 Introduction **59**
3.2 Consumer **61**
3.3 Producer **66**
3.4 District System **68**
3.5 Storage **70**
3.6 Source **70**
3.7 Power Flow **71**
3.8 Interactions between Physical and Logical Layers **71**

CHAPTER 4. SYSTEM SIMULATION AND OPTIMIZATION **76**
4.1 Introduction **76**

4.2 Application Setting: Joint Optimization of Design and Control 80

CHAPTER 5. SOFTWARE DEVELOPMENT 86
5.1 Executable Software 86
5.1.1 Input Module 92
5.1.2 Rule-based Logic Conflict Detection 99

CHAPTER 6. application of the up schema 102
6.1 Ten Selected Buildings on the Georgia Tech Campus 102
6.2 Setting1: Energy Performance Base (As-is) 107
6.3 Setting2: Rule-based Controls 108
6.4 Setting3: Design Optimization + Parameterized Optimal Control 115
6.5 Setting4: Design Optimization + Optimal Control 121

CHAPTER 7. CONCLUSIONS 132
7.1.1 Future Potential Updates on the UP Schema and Developed Software 133
7.1.2 Future Potential Research on Grid-interactive Efficient Buildings 134

APPENDIX A. DESCRIPTION OF DEFAULT SUBHEADING SCHEME 136
A.1 Result Validation 138

APPENDIX B. COMPARISON BETWEEN THE ORIGINAL AND LINEARIZED SOLAR MODELS 142
B.1 Result Comparison between the Original and Linearized Models 144

APPENDIX C. RBC LOGICS IN ENERGY SETTINGS 2 147

APPENDIX D. SOURCE CODE OF THE UP SCHEMA AND uP INSTANCES 149
D.1 Source Code of the UP Schema 149
D.2 UP Instance of Setting 2 215
D.3 UP Instance of Setting 3 226
D.4 UP Instance of Setting 4 237

REFERENCES 243

LIST OF TABLES

Table 1 List of Relationships in the Physical Layer and Their Definitions (M:N: many-to-many relationship, N:1: Many-to-one relationship, 1:N: one-to-many relationship) 34

Table 2 List of the `Time` Sensors 39

Table 3 List of the `Weather` Sensors 40

Table 4 List of the `Consumer` Sensors 40

Table 5 List of the `Producer` Sensors 41

Table 6 List of Storage Sensors 41

Table 7 List of the `PriceSignal` Sensors 41

Table 8 List of the `DistrictSystem` Sensors 42

Table 9 List of Available Actuators in the `Building` Element 47

Table 10 List of Available Actuators in the `Producer` Element 47

Table 11 List of Available Optimal Control Variables in the `Building` Element 49

Table 12 List of Available Design Optimization Variables in the `Building` Element 51

Table 13 `Consumer` Simulation Model Input/Output List 62

Table 14 List of EPC Simulated Outcomes (y) 64

Table 15 Producer Simulation Model Input/Output List 66

Table 16 `DistrictSystem` Model Input/output List 68

Table 17 Indicative Efficiencies for Different Prime Mover Technologies for Building-integrated Cogeneration Installations (EPA-NR 2007) 69

Table 18 `ElectricityBattery` Model Input/output List 70

Table 19 List of Design Parameters and Control Variables of Network Nodes 80

Table 20 Ten Selected Building's Information 103

Table 21 Two Selected EVCS's Information 104

Table 22 Electricity Cost for the Georgia Tech Test Cases 104

Table 23 Community PV Information 108

Table 24 List of Design Variables and Their Costs in Setting 3 108

Table 25 Number of Demand Response Logic Occurrences Comparison 115

Table 26 List of Design Variables and Their Costs in Setting 3 116

Table 27 List of Control Variables and Their Costs in Setting 3 116

Table 28 Additional Simulation Setup in Setting 3 117

Table 29 Optimal Parameter Results in Setting 3 119

Table 30 Building Archetypes in Anatolia Community Energy Model Information, Adapted from (Carneiro 2017) 123

Table 31 Three Selected Residential Building's Information 123

Table 32 Electricity Cost for the Anatolia Community Test Case 124

Table 33 List of Design and Control Variables Used in Setting 4 126

Table 34 Additional Simulation Setup in Setting 4 126

Table 35 Optimal Design Parameter Results in Setting 4 128

Table 36 PV Model Test Setup 144

Table 37 RBC Logic Inputs in Setting 2 147

LIST OF FIGURES

Figure 1 Grid Architecture (Grijalva 2014) 3

Figure 2 Simulation viewed as a (virtual) experiment (Augenbroe 2019) 17

Figure 3 (a) Example of an Urban Energy Setting; Hypothetical Operation Scenario (b) Physical Layer (c) Logical Layer 27

Figure 4 High-level View of UP Schema 28

Figure 5 Elements in the Physical Layer (Other physical elements also contain the "Name" child elements and the "Key" attribute) 31

Figure 6 Urban Energy Network Representation in a Directed Graph 33

Figure 7 Relationships in the Physical Layer (Other relationships also include the "PhysicalLayerRelationship" complex type) 33

Figure 8 Specialized Logical Layer and Relationship with the Physical Layer 36

Figure 9 Structure of the Logical Layer 37

Figure 10 Weather and PriceSignal Sensors (Weather sensors have their unique IDs) 43

Figure 11 Building Sensors (Each Building sensor has `Key` and `TimeStep` attributes.) 44

Figure 12 Logical Layer Input Pseudocode 45

Figure 13 Example of Rule-based Logic in the Controller Element 45

Figure 14 Structure of the Actuator Element 46

Figure 15 Structure of the `OptimalControl` Element 48

Figure 16 List of Available Optimal Control Variables in the `Producer` Element 49

Figure 17 Structure of the `DesignOptimization` Element 50

Figure 18 List of Available Design Optimization Variables in the `Producer` Element 51

Figure 19 List of Available Design Optimization Variables in the `DistrictSystem` Element 51

Figure 20 List of Available Design Optimization Variables in the `Storage` Element 52

Figure 21 Relationships in the Logical Layer 53

Figure 22 UP instance – Physical Layer 55

Figure 23 UP instance – Logical Layer Sensor, Controller and Actuator Elements (Cont'd from Figure 22) 56

Figure 24 UP instance – Logical Layer Relationship Elements (Cont'd from Figure 23) 57

Figure 25 Building Energy Performance Analysis Process (Lee 2012) 63

Figure 26 Schematic of Example Setting 72

Figure 27 Example Urban Setting's UP instance (Physical Layer) 74

Figure 28 Example Urban Setting's UP instance (Logical Layer – Sensor, Controller, and Actuator, Cont'd from Figure 27) 75

Figure 29 Example Urban Setting's UP instance (Logical Layer – Relationship, Cont'd from Figure 28) 75

Figure 30 (a) Example Urban Scenario (b) "Islanded" Physical Components for the Local Scope 79

Figure 31 Software Architecture 86

Figure 32 Principle of RBC Operation 88

Figure 33 Principle of Parameterized Optimal Control and Design Optimization Operation 89

Figure 34 Principle of Optimal Control and Design Optimization Operation (Bi-level Optimization) 91

Figure 35 Physical Layer Input Sheet – Component Definition 92

Figure 36 Physical Layer Input Sheet – Component Relationship Definition 93

Figure 37 Logical Layer Input Sheet – Simulation Type Selection 94

Figure 38 Logical Layer Input Sheet – Design Variable Input 95

Figure 39 Logical Layer Input Sheet – Parameterized Optimal Control Variable Input 95

Figure 40 Logical Layer Input Sheet – Optimal Control Variable Input 95

Figure 41 Logical Layer Input Sheet – Rule-based Control Sensor Input 97

Figure 42 Logical Layer Input Sheet – Rule-based Control Sensor Input 97

Figure 43 Logical Layer Input Sheet – Rule-based Control Controller Input 97

Figure 44 Rule-based Control Logic Conflict Detection Algorithm 100

Figure 45 (a) Instantiated Sensor/Actuator from the Hypothetical Setting in Bipartite Graph (The arrow represents control) (b) Example Logic Matrix 101

Figure 46 Selected Georgia Tech Buildings for the Test Cases 102

Figure 47 Two selected EVCS's Schedule 104

Figure 49 Directed Graph Representation Setting 2&3 106

Figure 50 Performance Base Case Results for Setting 1 107

Figure 51 Electricity Use and the Trigger for DR for Logic (a) 109

Figure 52 Results for Setting 2 if logic (a) and 8,000 m^2 PV are applied 111

Figure 53 Results for Setting 2 if logic (b) and 8,000 m^2 PV are applied 111

Figure 54 Results for Setting 2 if logic (a) and 10,000 m^2 PV are applied 112

Figure 55 Results for Setting 2 if logic (b) and 10,000 m^2 PV are applied 112

Figure 56 Electricity Battery State Duration Curve 114

Figure 57 Net Present Value (a) Without Control parameters (b) With Control Parameters 118

Figure 58 Anatolia Community – Rancho Cordova, Sacramento County, CA (Carneiro 2017) 122

Figure 59 Directed Graph Representation Setting 4 123

Figure 60 Natural Gas Cost for the Anatolia Community Test Case 124

Figure 61 Weekly Average Dry-bulb Temperature of Sacramento, CA in TMY3 125

Figure 62 Cost Function Value Over the Genetic Algorithm Generation 127

Figure 63 Power Flows for Physical Components and Building Indoor Temperature for No Lighting/Appliance Use Reduction (The red line indicates the start of the power outage) 128

Figure 64 PV Electricity Generation and Stored Energy in Electricity Battery and Power Flows for No Lighting/Appliance Use Reduction (The red line indicates the start of the power outage) 129

Figure 65 Power Flows for Physical Components and Building Indoor Temperature for 30% Lighting/Appliance Use Reduction (The red line indicates the 129 start of the power outage)

Figure 66 PV Electricity Generation and Stored Energy in Electricity Battery and Power Flows for 30% Lighting/Appliance Use Reduction (The red line indicates the start of the power outage) 130

Figure 67 5R3C Thermal Model 136

Figure 68 T03 (a) Heating Need (yearly difference : 11.6%) (b) Cooling Need (Yearly difference : 1.6%) 140

Figure 69 T04 (a) Heating Need (yearly difference : 12.0%) (b) Cooling Need (Yearly difference : 1.9%) 140

Figure 70 T05 (a) Heating Need (yearly difference : 0.0%) (b) Cooling Need (Yearly difference : 0.0%) 140

Figure 71 Caddell (a) Heating Need (yearly difference : 0.0%) (b) Cooling Need (Yearly difference : 0.0%) 140

Figure 72 Mason (a) Heating Need (yearly difference : 5.3%) (b) Cooling Need (Yearly difference : 0.7%) 141

Figure 73 Smithgall (a) Heating Need (yearly difference : 0.0%) (b) Cooling Need (Yearly difference : 0.0%) 141

Figure 74 Savant (a) Heating Need (yearly difference : 0.0%) (b) Cooling Need (Yearly difference : 0.0%) 141

Figure 75 Solar Angles for Vertical and Horizontal Surfaces (ASHRAE 2017) 142

Figure 76 Results Comparison between the Original and Linearized Models (Generated power difference = generated electricity from the original model–generated electricity from the linearized model) 146

LIST OF SYMBOLS AND ABBREVIATIONS

AHU	Air Handling Unit
BAS	Building Automation Systems
BOT	Building Topology Ontology
CHP	Combined Heat and Power
CVRMSE	Ccoefficient of Variance of Root Mean Squared Error
DER	Distributed Energy Resources
DG	Distributed Generators
DO	Design Optimization
DR	Demand Response
EA	Evolutionary Algorithm
ECM	Energy Conservation Measures
ESS	Energy Storage Systems
EV	Electric Vehicle
EVCS	Electric Vehicle Charging Station
gbXML	Green Building XML
GEB	Grid-interactive Efficient Buildings
GHG	Greenhouse Gases
GIS	Geographical Information System
GSHP	Ground Source Heat Pump
IFC	Industry Foundation Classes
IoT	Internet of Things
LOD	Levels of Details
MILP	Mixed-integer Linear Programming
MINLP	Mixed-integer Non-linear Programming

obXML	Occupant Behavior XML
PID	Proportional–Integral–Derivative
RBC	Rule-based Control
UBEM	Urban Building Energy Models
UP Schema	Urban Prosumer Schema

SUMMARY

Many global cities have announced ambitious net-zero energy consumption targets or net-zero CO_2 emissions plans. It is well recognized that this can only be realized through a mix of measures such as efficiency improvements at the sites of consumption and decentralized energy generation, storage and delivery mechanisms. This transition will not happen without major changes to energy supply networks, especially in the way they enable frictionless inclusion of renewable energy sources and local supply, for instance through microgrids.

At the urban scale, buildings constitute the major consumers of electricity and their integration through building-to-building and building-to-grid controls is crucial to realize efficient energy sharing in urban energy networks. Over the last decade, the building energy simulation domain has moved its focus from traditional local studies to urban energy studies. The main objective of this thesis is to make a contribution to this growing research domain, especially in enabling the simulation of energy supply networks in a robust manner and at a large scale. It is possible to simulate such networks with customized software but considering that there is no systematic way to specify urban energy models (especially with multiple concurrent control topologies), the simulation software has to be hand-customized which leads to opaque simulations that moreover are hard to use for rapid variant explorations. The thesis argues that this can be overcome by the development of an urban prosumer (UP) schema that facilitates the specification and automated mapping of an urban energy network into simulations, focusing on the effective specification of controls outside the software.

At a high level, the UP schema is comprised of a physical and a logical layer. The physical layer conceptualizes existing urban energy networks using directed graphs for energy transport between nodes. The logical layer conceptualizes how the dynamic processing (reasoning) of sensor data leads to instructions to a set of actuators that execute the control. In doing so, two levels of control are distinguished: (a) “private” (mostly rule-based) control such as the internal HVAC system following temperature setpoints, (b) “public” control that is exposed to the rest of the network and thus within the scope of the UP schema. Public control can be either rule-based or optimal control, the latter driven by an appropriate optimality criterion, defined at a network scale. In design situations, the optimality criterion is not limited to control variables but can also include design parameters, such as building design parameters, solar installation sizes, community battery size, and the number of EV charging stations. Mixed-integer non-linear programming (MINLP) is used to solve optimal control problems. The genetic algorithm is employed to solve design optimization problems.

The case studies using the UP schema for ten Georgia Tech campus buildings are presented. The purpose of the case studies is to prove that the UP schema can facilitate simulations involving different levels of controls. The simulations target optimal energy decisions for the selected campus buildings in the presence of PV and electricity battery. Additionally, three residential buildings in California are chosen to investigate how the design and control parameters act together to avoid the power outage situation with the embedded UP schema in the simulation platform.

CHAPTER 1. INTRODUCTION

1.1 Background

Traditionally, electricity has been provided mostly from central power plants to electricity consumers in a one-way power flow fashion. Power plants have become the distributor of electricity as a widely available commodity in the 20th century. However, this conventional central energy supply model has been shown to have drawbacks for the realization of the new energy sharing economy. Most importantly, it puts limitations on the optimal use of non-renewable fuels, the expansion of existing networks, reducing congestion on existing lines, and optimal response to threats from extreme weather and malicious entities (Fathima and Palanisamy 2015).

Currently, the source of electricity largely relies on fossil fuels, and their use in electricity generation emits greenhouse gases (GHG). Electricity is generated approximately 61% from fossil fuels (aggregation of natural gas, coal, and petroleum) in the U.S. based on 2018 data (LLNL 2019). To fight increasing GHG emission trends, many global cities announced ambitious net-zero energy consumption targets or net-zero CO_2 emissions plans. For instance, 19 cities, including 7 U.S. cities (Los Angeles, New York City, San Francisco, San Jose, Santa Monica, and Washington D.C.) announced in 2018 that they would commit to cut GHG emissions by ensuring that new buildings operate at net zero carbon by 2030 (Cities 2018). It is well understood that this transition will not happen without major changes to energy supply networks, especially in the way they enable frictionless inclusion and optimal allocation of renewable energy sources.

To address emerging energy landscapes, microgrids have gained popularity, especially because of their ability to supply energy in a more resilient manner. A microgrid is defined as “a group of interconnected loads and distributed energy resources within clearly defined electrical boundaries that act as a single controllable entity with respect to the grid” (Ton and Smith 2012). Microgrids can comprise renewable distributed energy resources (DER) and distributed generators (DG), energy storage systems (ESS), and (controllable) loads. Planning microgrids comes with many challenges such as the proper selection of technology, sizing of DERs, executing the optimal operation, investigation of financial feasibility, analyzing the uncertainty in managing risks, etc. Rewards can be substantial, typically in the form of emission reduction credits, distributed generation benefit allocation through net metering, demand response rewards for peak power reduction, feed-in tariffs, and additional tax credits (Husein and Chung 2018). Furthermore, (NAVIGANT 2018) forecasted that there would be at least 14,000 GW of cumulative installed DER capacity deployed globally in the next two decades, assuming all installed assets after 2017 remain in service or are replaced.

The building residential and commercial sectors accounted for a large amount of total U.S. energy consumption, i.e., roughly 40% or about 40 quadrillion British thermal units in 2018 (U.S. EIA 2019). As buildings play such a dominant role on the energy consumer side, the grid-interactive efficient buildings (GEB) concept emerged. The basis of the GEB concept integrates and continuously optimizes DERs for the benefit of building owners and occupants. It is important to realize that building interventions in design or through retrofit of existing buildings should be regarded as just another DER and energy resource that is unlocked by increasing the energy efficiency of a building. The investment in energy efficiency thus becomes part of the mix of DER investments. Realizing this puts parallel focuses on design concepts and optimal operation of energy flows to the place of consumption. End-user energy costs vary with different occupant patterns and preferences, utility price signals, weather forecasts, and transient availability of on-site generation and storage. We are in the early days of development and adoption as most states are in the rudimentary stages of implementing GEB (Northeast Energy Efficiency Partnerships (NEEP) 2020).

1.2 Problem Statement

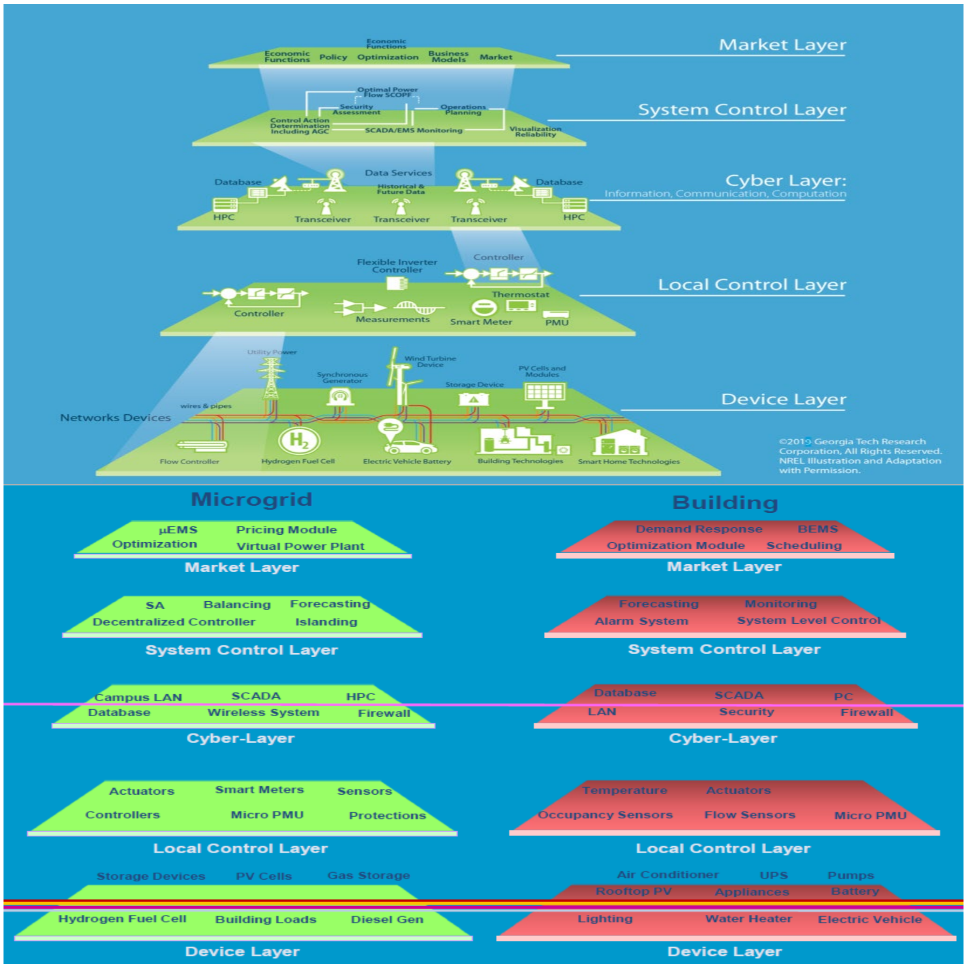


Figure 1. Grid Architecture (Grijalva 2014)

As Figure 1 shows, the electricity grid is a business-cyber-physical system consisting of five layers, device, local control, cyber layer, system control layer, and market layers (Grijalva 2014). The building energy simulation discipline has assisted energy decisions related to the device layer (a.k.a. physical layer). However, the discipline has not successfully run full-blown simulations with orchestrated local control layer and grid level global control in a robust manner at a large scale. This is not surprising because as the number of physical components increases, the number of relationships between physical components increases exponentially in urban-scale simulation. The control layer is informed by “sensors” that measure (observable) states and measure or predict specific outcomes from the physical layer and executes controls through the operation of “actuators.” Sensor and actuator are to be interpreted in the widest sense as a measuring entity and a controlling entity. We will come back to this later in this thesis when we introduce them as generic entities in the schema of the urban energy network. The complexity of urban scenarios increases because sensors and actuators are inter-linked logically but also closely intertwined with the physical layer. In addition, there is no flat control layer; it is in fact stratified into global and local control, the latter being often executed autonomously within a component of the network. A good example of this is a building’s HVAC control that maintains setpoint temperatures in a building. This is largely executed in an autonomous fashion. But at the higher level, the decision can be made to curtail the power consumption of the HVAC system or change the temperature setpoints in certain zones for a specified period of the day to maximize the use of resources elsewhere in the network. This type of control must thus operate at the global network level and its actions are based on forecasting the effect of a control action in future states and outcomes. Only a multi-layered control approach enabled by simulation-based forecasting can support this. For the above reasons, it becomes crucial to flexibly integrate the local control layer into the global control execution or in the design decision-making process. Simulations are thus executed, in the design stage (in vitro simulation) as well as in the operation level (embedded control simulation). This thesis is rooted in the argument that the development of computational models that meet this need have to be supported by a formal specification of an energy network. Such formalization can be based on a conceptualization of the nodes in an urban network and how they interact with each other and are operated by a combination of local and global control systems.

The current lack of such conceptualization results in hard-wired coding inside urban simulation software which makes it impossible to inspect and modify the system and its operational control structure outside the software. A formalized approach is deemed necessary to support a dialogue with different involved stakeholders. The benefits of this approach are i) ability to develop/modify the complex simulation models rapidly ii) to increase the efficiency of embedding the control layer inside simulation.

This dissertation thus aims to enable microgrid control integration in urban-scale simulations by formalizing urban producers and consumers ("prosumers") as network components with embedded, partly private behavior. A prosumer exposes some of its state (by exporting sensor data) and reacts to certain control actions (allowing actuators to be controlled externally).

1.3 Literature Review

1.3.1 Physics-based Energy Simulation of the Building Stock

To conduct an urban or more generally a building stock's energy analysis, there are two approaches, top-down and bottom-up. The top-down approach is used at an aggregated level and typically aims at fitting a historical time series of national energy consumption or CO_2 emissions data to the aggregate set of buildings (Kavgic et al. 2010). On the other hand, the bottom-up approach energy models for individual buildings and determine the aggregated energy consumption, which can then be used to identify areas for improvement (Swan and Ugursal 2009). In many cases buildings are classified into archetypes and the simulation is performed for the set of archetypes and aggregated to urban scale by type-occurrence multiplication factors. In this thesis, we focus on the bottom-up modeling of all connected urban nodes in the energy network. No archetype based aggregation is used as we want to garget the actual control dynamics of each individual node. Hence the literature review will be carried out only for that category, with a special focus on how researchers facilitate controls of each physical component. The current state of the art of urban-scale energy modeling is given in (Kavgic et al. 2010), (Reinhart and Cerezo Davila 2016) and (Johari et al. 2020).

(Lee et al. 2013) develop a methodology to express overall network energy performance (NEP) that can manage multiple relationships between energy consumers and producers in the network. It focuses on commercial buildings as the prime nodes in the network. The NEP model uses a reduced-order hourly calculation method and represents energy relationships between energy consumers/suppliers in directed graphs.

The authors use the model to test seven different retrofit options of the Georgia Tech campus. When applying different campus retrofit technologies, they report each retrofit option's CO_2 emission reductions, e.g., Lighting fixture replacement (15%), district cooling plant efficiency improvement (4.9%), district heating plant efficiency improvement (1.1%), Combined Heat Power (CHP) cogeneration (5.9%), Combined Heat Power (CHP) trigeneration (11.1%), PV station installation for the campus buildings and DC plant chillers (16.6%), BIPV installation for the campus dormitories (1.8%).

(Orehounig, Evins, and Dorer 2015) present a method of integrating energy from decentralized energy systems at the neighborhood scale using the energy hub approach. The model introduced in this research allows to add various types of technologies for distribution, conversion, and storage of energy. The paper takes four different urban energy-hub layouts consisting of PV, wood pellets, electrical grids, hydropower turbines, CHP, solar thermal collectors. The results show that a significant reduction of CO_2 emissions (between 81% - 90%), an increase of the energy efficiency (between 53% and 55%), a curtailment of peaks on the electrical grid (between 73% and 79%) for all scenarios.

(Quan et al. 2015) show the development of a GIS-based urban building energy modeling system called Urban-EPC to support urban planning and conduct a simulation of all buildings in Manhattan as a test-case. The Urban-EPC engine, a modified version of a reduced-order building energy model, based on the ISO 13790:2008 standard, consists of three sub-engines (shading, microclimate, and occupancy engines). The sub engines address external/mutual shading effects on windows, local air temperature/wind patterns, and occupant density based on population and job data, respectively. The

Manhattan results conclude that when urban contexts (e.g., mutual shading, microclimate, and occupant behavior) are taken into consideration, the simulated results are closer to the recorded overall electricity usage (NMBE:0.25, CVRMSE:0.69), compared to when urban spatial proximity aspects and connects were not considered (NMBE: 0.50, CVRMSE: 0.83).

(J. A. Fonseca and Schlueter 2015) show the development of a GIS-based model for characterizing spatiotemporal building energy consumption patterns in neighborhood and city districts. The simulation engine is also based on the simple hourly dynamic method of ISO 13790. EN15316 and EN 15241 with minor manipulation of the HVAC part. The software is validated against the actual energy data of a peer model for a city district in Switzerland.

Based on the past research reported in (J. A. Fonseca and Schlueter 2015), (J. A. Fonseca et al. 2016) show the integration of time-dependent methods for building energy performance simulation, conversion and storage technologies simulation, assessment of local energy potentials, bi-level energy systems optimization and multi-criteria analysis. The test case model for a downtown area in Switzerland shows the integration of 50% to 80% of buildings in thermal micro-grids and exploitation from 50 to 100% of the available solar potential.

(Davila, Reinhart, and Bemis 2016) report a method to construct urban building energy models (UBEM) and apply that to Boston. To develop the models, the authors obtain "2.5D" massing geometric data based on the building geometry GIS dataset and use 52 archetypes to generate non-geometric data for each object. After running the simulation, an average absolute error of 40% in the total energy use for both gas and electricity is reported. Two different scenarios, a large scale deployment of solar PV and a demand response control development, are added in the simulated experiment. The authors conclude that the PV scenario reduces daytime electrical energy peak by 48 MW at 5 p.m. on July 7th, and the DR scenario decreases the energy use by 49 MWh on July 7th.

(Chen et al. 2017) show the development of a web-based tool, called CityBES, which conducts retrofit analysis and visualizes city wide building energy datasets. The software collects necessary data such as weather files, building data in CityGML format, and then applies energy conservation measures (ECM) in comparison to the current situation. A demonstration case of CityBES selects 540 small and medium-sized office and retail buildings for the retrofit analysis. It applies the following five ECM (i) replacement of existing heating system with AFUE 95 gas boilers (ii) replacement of the existing cooling system with a SEER 14 packaged rooftop unit (iii) replacement windows with U-value(0.25)/SHGC(0.18) (iv) installation of an HVAC economizer (v) replacement lighting with LED bulbs (0.6 W/ft^2). It is found that the LED lighting upgrade (option v) is expected to save the most annual site energy consumption (approximately $150 \cdot 10^8$ Btu) among any other isolated ECM application. If the economizer option (iv) and the LED lighting option (v) are applied together, they are expected to save about $170 \cdot 10^8$ Btu annual site energy consumption.

(New et al. 2018) provide an overview of the technical capabilities developed for generally-applicable urban-scale building energy modeling adaptable to any geographical area. More than 130,000 buildings are modeled using EnergyPlus and OpenStudio, and the results are visualized in CesiumJS.

(Nagpal et al. 2018) propose automatic calibration techniques that use data-driven approximation with statistical surrogate models. The authors use random forests and neural network algorithms to develop the surrogate models. The case study conducted for three MIT campus buildings shows that the new methodology is most effective for building types where only a few parameters primarily drive the energy use. On the other hand, when the energy use is affected to a similar degree by a large number of parameters, the accuracy falls sharply.

(Letellier-Duchesne et al. 2018) propose a workflow developed around an existing urban planning tool that supports the simulation potential of district heating scenarios at an early stage. The methodology is comprised of a simple three-step workflow: building energy models are defined and calculated, district energy system topology is determined from revenue optimization while exploring different thermal plant schemes.

(Nagpal and Reinhart 2018) compare two different approaches for modeling campus-scale building energy models. The first approach is based on utilizing a combination of statistical techniques that attribute energy use to the primary programmatic uses on campus before evaluating the effect of energy efficiency measures for those specific program types. The second approach is based on a bottom-up energy modeling methodology to forecast the impact of building-by-building retrofit scenarios. The results from the test case conducted on MIT campus buildings indicate that the overall campus energy use intensity match closely for the two modeling approaches illustrating that both approaches are successfully calibrated to the measured energy use at the campus scale. However, for the individual energy consumption predictions, the second approach outperforms the first approach.

(Remmen et al. 2018) report the development of a TEASER ("Tool for Energy Analysis and Simulation for Efficient Retrofit"), an urban energy modeling software for rapid assessment of energy efficiency potentials of building stocks. The simulation engine is based on a reduced-order building energy model. TEASER enables integrating multiple data sources such as databases, standardized information models (e.g., CityGML), or self-defined file formats to interface with the simulation. Furthermore, TEASER also allows to export ready-to-run Modelica models. The authors conduct three use cases at the building/neighborhood/urban scale, respectively. The results show that the software can be utilized as a fully scalable and adaptable urban energy modeling tool.

(Schiefelbein et al. 2019) present an urban energy modeling approach driven by open-source geographical information system (GIS) datasets to reduce input data uncertainty.

There have been additional efforts to take uncertainty in building stock models into consideration in physics-based simulation approaches. More details about one such effort, a physics-based stochastic modeling approach, are introduced in (Lim and Zhai 2017).

(Booth, Choudhary, and Spiegelhalter 2012) show the development of a stochastic urban scale domestic energy model (SUSDEM) and set up a calibration framework capable of quantifying the effects of second-order (epistemic) uncertainties. The case study suggests how to expand the framework for retrofit analysis at an urban-scale.

(Heo et al. 2015) develop a generic analysis framework that can incorporate all sources of uncertainty for energy efficiency measures. The authors conduct a case that shows how the use of uncalibrated versus calibrated models impacts large-scale retrofit decisions. The results demonstrate that calibration reduces the uncertainty of model predictions to the level that the resulting accuracy can indeed support decision making with more confidence.

(J. N. B. Fonseca and Oliveira Panão 2017) report the implementation of a bottom-up physics-based housing stock model to predict the energy demand using the Monte Carlo method. The overall performance of the model in the prediction of energy indicators used in benchmarking and, more specifically in Energy Performance Certification is satisfactory.

Summarizing above sources, it can be concluded that building stock simulation has been successfully applied for large-scale building assessments and the study of the use of DER at city scale. The models are shown to successfully assist various urban energy-related decisions. Given the fact that uncertainty is especially relevant in building stock simulation where only limited information about each urban object is available, some researches have started taking uncertainties in urban areas into consideration as apparent from the citations above. However, all of the past efforts use an ad-hoc software architecture to achieve their goals. Moreover, hardly any attempt is made to develop a formalized representation of the objects and their relationships in the urban energy network and typically lack a clear specification of how different layers of control are operated and orchestrated. As a result, all efforts have led to rather narrowly applicable software architectures and as such do not provide a basis for a common approach to an extendable set of re-usable modules in a generic software architecture. The objective of this thesis is to lay the groundwork of this ambitious goal. It targets the development of a common conceptualization of the physical and logical relationships at the urban scale. This is accomplished by capturing the semantics in a conceptual data model, formalized in the targeted UP schema. It is expected that the schema will enhance future developments, in particular in capturing how different layers of controls can be defined in a generic fashion (i.e., outside the actual code), and consequently be executed from the external specification (an instance of the schema) inside a simulation platform. The latter is especially relevant for designing urban building-integrated microgrids with multiple local and global control variables where high flexibility and, efficient data management are crucial. The next section dives more in-depth into this aspect.

1.3.2 Controls of Building-integrated Microgrids

The increasing interest in integrating DERs with power supply to buildings presents major challenges from the viewpoints of reliable and effective operation and control. (Fontenot and Dong 2019) categorize hierarchical microgrid control structures. They highlight several control execution types, such as rule-based control, agent-based control, or multi-agent control systems and model predictive control. In all types, except rule-based, some optimality criterion has to be defined to drive the real-time control. The following review details how previous researches include controls in their studies

(Gupta and Gupta 2015) propose optimal microgrid operation in a deregulated environment considering renewable power uncertainties using a robust optimization-based approach. The authors conduct two case studies consisting of grid-connected and grid-isolated modes, and their test cases involve a wind power uncertainty model designed using ARIMA modeling. The proposed robust optimization approach is compared with deterministic and stochastic approaches. The results show that the operation cost decreases by 5.02% in the grid-connected mode compared to deterministic optimization, and by 3.07% in the grid-connected mode compared to stochastic optimization. Besides, for the grid-isolated mode, 17.25% and 11.02% of operation cost reductions are found compared to the deterministic optimization and stochastic optimization, respectively.

(Li, Roche, and Miraoui 2017) propose a methodology to determine the optimal sizing for a stand-alone microgrid. The methodology combines an evolutionary algorithm (EA) for sizing and mixed-integer linear programming (MILP) for scheduling. The authors show that their approach is better suited to consider advanced energy management strategies, anticipating decisions compared to rule-based approaches.

(Ayodele et al. 2017) investigate an approach for maximizing the benefits of a Stand-Alone Photovoltaic-Battery (SAPVB) system via techniques that provide optimum energy management. A rule-based load management scheme is also demonstrated. The test case is comprised of a residential building in Ibadan, Nigeria, which is used in four different scenarios to verify the effectiveness of rule-based control. The results reveal that in case 1 (no energy management deployment, base case), the percentage satisfaction of the critical (e.g., fridge, lighting, TV, etc.) and uncritical (e.g., Water pump, washing machine, etc.) loads by the PV system are 49.8% and 23.7%. However, with the implementation of the energy management scheme in three different scenarios, the lowest satisfaction percentage of critical and uncritical loads among the three scenarios are 87.2% and 65.4%, respectively, which is much higher compared to the no energy management case.

(Mazzola et al. 2017) develop a framework to assess operation cost reduction potential under a range of assumptions about the quality of load and photovoltaic generation forecasts. The test case consisting of a village of 600 households reveals that cost savings ranging between 2% and 7% are obtainable depending on the forecast quality and the composition of the microgrid.

(Alimohammadisagvand, Jokisalo, and Sirén 2018) study the effect of demand response (DR) actions on energy use and energy cost with two alternative heat generation systems (a ground source heat pump (GSHP) heating system and a water-based electric heating system) in a detached house in a cold climate. The authors use two rule-based DR control algorithms presented in recent work and develop two new rule-based DR control algorithms based on the trend of future hourly electricity price (sliding-maximum subarray and moving average methods). The results show that the maximum annual savings in the heating energy and cost occur when the rule-based DR control algorithm based on the trend of future hourly price. For the GSHP heating system, the heating energy and costs savings are about 10% and 15%, respectively, for the GSHP heating system; and about 1% and 8%, respectively, for the water-based electric heating system.

(Thomas, Deblecker, and Ioakimidis 2018) investigate the cooperative evaluation of an energy management systems (EMS) operation in a building taking into account (i) bidirectional energy trading capabilities of an EV fleet arriving at an office building under a stochastic EV's driving schedule, (ii) the impact of PV uncertainty on EMS operation based on real smart-metering data and comparing it with a deterministic PV production approach and (iii) the effect of setting different prioritization factors in selling energy back to the grid from the resources on total system's cost. All cases are studied with linearized models and MILP based determination of control actions. The authors conclude that the total expected daily cost for the system in the stochastic cases is much lower than their corresponding deterministic cases.

(Nelson and Johnson 2020) develop two model predictive control approaches to optimize microgrid dispatch, one with participation in real-time ancillary service markets and the other without participation. The test case which consists of an office building with a grid-tied solar PV, electricity battery, and diesel generator demonstrates that the model predictive control algorithms can reduce operating expenses by up to 13.7% compared to logic-based controls and can reduce net operating expenses by up to 23.5% through participation in ancillary service markets.

From the studies mentioned above, it is apparent that a majority of efforts have focused on finding or applying various types of controls (e.g., rule-based, optimal control, model predictive control) in specific settings. None of the reported studies sought to conceptually define and facilitate diverse controls for rapid iterations with the simulation platform.

1.4 Importance of an Urban Prosumer Schema in Urban Energy Simulation

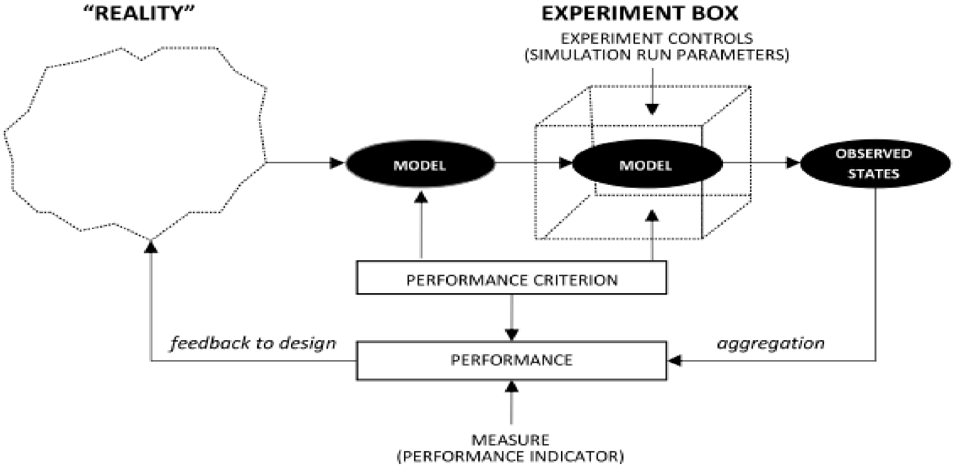


Figure 2. Simulation viewed as a (virtual) experiment (Augenbroe 2019)

The scope of this thesis is the simulation-based design/operation of urban energy systems. This results in a multi-domain and multi-criterion parameter optimization problem with embedded optimal control. To get a better understanding of the role of simulation, it is worthwhile to have a closer look at Figure 2, which suggests the simulation is a process that converts a "piece of reality" into a model. Different scenarios can be experimented with by putting the model inside the "experiment box" (de facto the simulation tool). The simulation generates observable output states that can be aggregated/translated into performance measures whose values determine how well the proposed design meets the performance requirements of the stakeholders (Augenbroe 2019).

In terms of urban-scale energy models, their development must include several (i.e., private and public) control layers each operating on different "sub-topologies", i.e., a group potentially dispersed urban components. Considering the fact that there is no systematic way to develop urban energy models that can implement the topology of different control variants, it prevents energy modelers from investigating variants in the experiment box since it is too time-consuming to model each possible variant. It should be noted that each variant may be characterized by a different set of sensors, placed at different locations, exposed at different scales, and monitoring different outcomes at different granularities. In addition, each scenario is characterized by a set of actuators with specific functions and governed by localized (often sub-optimal) and/or globally optimal control.

This reinforces the claim that it is imperative to develop an urban prosumer (UP) schema that captures the network-relevant energy concepts and relationships at the semantic level in order to expedite the deployment of the control layer in urban-scale energy simulation.

1.5 Research Hypotheses

This thesis targets the definition of a semantic urban energy model with the integration of private (local) and public (global) controls. Achieving this will enable a transparent and largely automatic implementation of a specification in an executable simulation. In this thesis, its development is undertaken in a proof of concept mode. It focuses on the demonstration of the approach in a campus case study which serves to verify the following two hypotheses.

- *Hypothesis 1*: Based on known urban prosumer settings, a generic conceptualization can be developed that captures the system semantics of collections of buildings with integrated energy flow control logic.

- *Hypothesis 2*: The conceptualization in the form of an urban prosumer schema facilitates the rapid specification of proposed or existing urban energy networks, including control topologies, in a form that can be automatically mapped to their execution in a simulation platform, enabling rapid iteration of design and control variants.

The first hypothesis examines the completeness of the proposed UP schema. We posit that a large variety of energy scenarios are conceptualized in the generic UP schema. However, this thesis does not claim that the UP schema will cover all kinds of physical network components or include all control elements and control layers. Instead, the test case intends to show the usefulness of the developed schema and how sophisticated controls can be captured in it. The plausibility of this hypothesis is thus examined with the test case.

The second hypothesis tests the usability of the proposed UP Schema. We posit that the developed schema contains the generic specification of realistic controls in such a way that its external definition can be read into the simulation platform and executed without additional coding. This hypothesis is examined by simulating various energy control scenarios in the test case.

In this thesis, the focus is on developing a campus test case and hypothesis 1 and verification of hypothesis 2. The test cases prove the UP schema's usability with an emphasis on more complicated control settings. This is accompanied by showing how the simulation platform can support the simultaneous optimization of design and control variables.

1.6 Organization of the Dissertation

This thesis is outlined as follows.

- Chapter 1 presents the background of traditional and current energy trends, motivating the need for a UP schema at the urban, and more specifically at the neighborhood/campus scale.
- Chapter 2 lays out the metadata for the early development of the UP schema.
- Chapter 3 describes the physics-based models for the nodes in the energy network as far as considered in this thesis.
- Chapter 4 states mathematical formulations of controls in the physics-based models introduced in Chapter 3.
- Chapter 5 introduces the developed executable software.
- Chapter 6 presents the application of the UP schema by exhibiting the three case studies.
- Chapter 7 summarizes the thesis with conclusions and states about the future work.

CHAPTER 2. DEVELOPMENT OF AN URBAN PROSUMER SCHEMA

2.1 Background

2.1.1 Metadata in the AEC Industry

In this section, the current existing metadata in the AEC industry is introduced. The current state of the art of schemas in the AEC industry is also well-described in (Pritoni et al. 2021).

Industry foundation classes (IFC) is the current repository of a standardized, digital description of the built environment (buildingSMART n.d.). The IFC schema can define physical components of buildings, manufactured products, mechanical/electrical systems, as well as more abstract structural analysis models, energy analysis models, cost breakdowns, work schedules, etc. (buildingSMART n.d.). The IFC aims to provide a universal basis for process improvement and information sharing in the construction and facilities management industries (Moon et al. 2011). Although the IFC has started by focusing on "static" product definitions, multiple additions have been made over the years to capture transient (time-dependent) information. This has been mostly limited to implementing simple control strategies. There have been multiple efforts to extend some of the classes in IFC to capture transient logic and certain types of dynamic rules that define the control sequences during the operation of the building. A recent study shows, for example, a method to specify HVAC controls in IFC. The example shows a rudimentary approach by including three HVAC control functions comprised of the definition of a heating curve, the setup of time schedules for the air handling unit, and the flow temperature control for the heating circuit (Benndorf et al. 2017). This and other studies rely mostly on linking external snippets of control logic (in a formal control language) to the building conceptualization. They come to the conclusion that this is generally hard to generalize. However, the more significant drawback of these efforts is that these representations for 10s or 100s of buildings would not only be impractical but be incongruent with the (in most cases) available low resolution of information that is available for buildings in an existing urban setting.

The Green Building XML (gbXML) schema was developed to facilitate the transfer of CAD-generated building information models in a simplified form and resolution as accepted as input by building simulation models (Roth 2020). The schema is small and well suited to capture a building at a low resolution. The current gbXML is mainly utilized for data transformation from BIM to some engineering analysis tools, especially building energy modeling (BEM) software. The transformation is handled inside current CAD software. gbXML is lightweight compared to IFC, which supports a comprehensive and generic representation of an entire building project (Gao, Koch, and Wu 2019). As this is not the focus at the urban scale, gbXML is better suited to model the building nodes in the larger urban network. Unfortunately, in the current form (gbXML version 6.01), the schema has many limitations and drawbacks to model the connection to other nodes and add the specific features that are crucial for the control definition. For example, the gbXML schema only has rudimentary elements to model sensors and no dedicated taxonomy of semantic types for sensors (Bhattacharya, Ploennigs, and Culler 2015). For this thesis, gbXML will be used (where relevant) as a handy conceptualization of part of the information about building nodes in the network with the proviso that other information will have to be added ad hoc due to the incompleteness of gbXML.

Project Haystack aims to standardize semantic data models and web services with the goal of unlocking value from the vast quantity of data being generated by smart devices. Applications include automation, control, energy, HVAC, lighting, and other environmental systems. (Project Haystack n.d.). Project Haystack is designed to help tag entities involved in a Building Automation Systems (BAS) environment. Entities can be automation devices (such as sensors, actuators, or controllers) or building equipment, including HVAC assets, lighting, or energy meters. Project Haystack lacks several features in the context of the Internet of Things (IoT). Its model exists as text documentation for now and is not available in a formal representation (Charpenay et al. 2015). This thesis has no direct use for the Haystack data model, but its concepts drive part of the conceptualizations undertaken here.

The Brick schema defines a concrete ontology of sensors, subsystems, and relationships (Balaji et al. 2018). In (Balaji et al. 2018), the authors use the brick schema to represent the entire spectrum of vendor-specific sensor metadata of six diverse buildings across university campuses, comprising 17,700 data points, while the percentage of mapping tagsets ranges from 96% to 99% (Balaji et al. 2018). For this thesis, the Brick schema is referenced where it conceptualizes each HVAC system component as a node and each relationship as an arc. The UP schema takes this approach and expands to physical components in urban areas.

CityGML is an open data model and XML-based format for virtual 3D city models and aims to provide a standard definition of the basic entities, attributes, and relations in a 3D city model (CityGML n.d.). CityGML covers all relevant features in urban areas and provides a semantic definition, attributes, relationships, and a 3D spatial representation (Gröger and Plümer 2012). CityGML can be applied to large areas and small regions and can represent the terrain and 3D objects at different levels of detail simultaneously. CityGML defines four levels of details (LOD) to represent city objects (Isikdag and Zlatanova 2009). In this thesis, there is no direct use for the CityGML data model, but its concepts drive part of the conceptualizations undertaken here. One of the reasons is that the scope of the proposed UP schema does at this time not include the urban context (e.g., coordinates, physical surroundings, address, etc.) of the energy network. This will need to be included when the creation of UP instances is supported by mappings from urban scene information such as stored in GIS and many other sources. This mapping and its execution in a GUI-based front-end are not considered within the scope of the thesis.

Building Topology Ontology (BOT) provides a high-level description of the topology of buildings, including stories and spaces, building elements, and 3D mesh geometry of spaces and elements. The BOT is comprised of seven classes, 14 object properties, and one datatype property. The classes in the BOT can be used not only for existing buildings but also for creating an abstract requirements model of a future building (Rasmussen et al. 2019). The BOT is not directly used in the development of the UP schema.

The occupant behavior XML schema (obXML) is developed to be used for the practical implementation of DNAS (Drivers, Needs, Actions and Systems) framework into building simulation tools and is designed to provide enough flexibility for both existing and future occupant behavior, building energy and system models to be captured consistently (Hong et al. 2015). The obXML schema will not be directly used in this thesis because the UP schema does not require occupants' information because this information is private to the building and hence not escalated to the level of the UP schema. A deeper discussion of the separation of information that is "exposed" or "hidden" inside network node models will be provided later. At this stage, it is important to note that all local "private" information of a network node is not exposed and therefore not treated in the UP schema. The obXML schema may be used in the private building node definition. As will be shown later, the modules that are used to represent the building behavior for each node in the network use native occupancy schedules that do not make explicit use of the obXML schema.

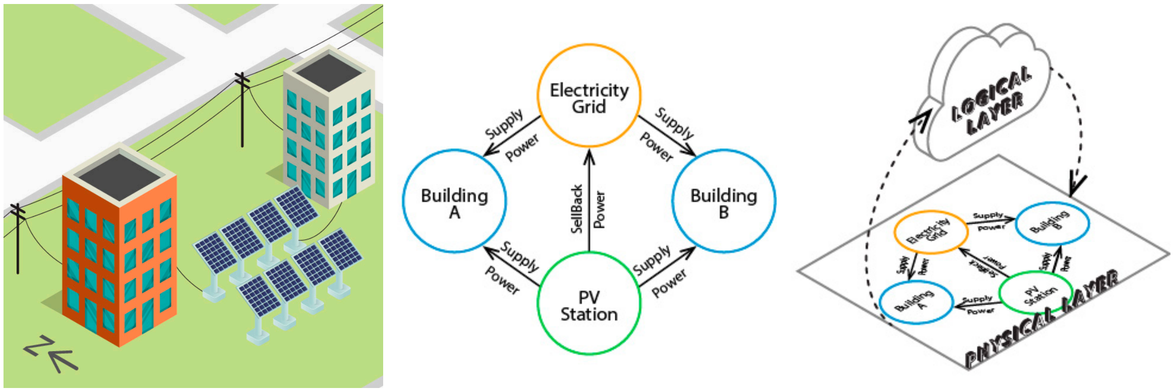
CTRLont is ontology-based modeling of the control logic of building automation systems (BAS) and allows to specify the domain of control logic in BAS formally. The authors successfully apply the developed ontology to an air handling unit (AHU) control test case (Schneider, Pauwels, and Steiger 2017). This thesis has no direct use for the CTRLont, but its concepts about how HVAC controls are implemented drive part of the conceptualizations undertaken in the UP schema.

In summary, it can be concluded that the AEC industry has made use of data models for a variety of different purposes. Many data schemas such as IFC, gbXML, BOT conceptualize the building itself to provide and effectively exchange necessary information regarding buildings. Some schemas such as IFC, Brick, CTRLont attempt to represent building systems or their control logics to effectively exchange their information or implement their controls by conceptualizing sensor, actuator, and control logic. Further, the CityGML schema conceptualizes objects in urban areas, and the obXML schema conceptualizes occupant behaviors and interactions with buildings. However, few if any attempts have been made to conceptualize energy networks at coarse resolution with buildings, energy generation and storage as main nodes and with embedded global and local control. Thus, the goal of the schema development is to conceptualize physical and logical elements and their relationships so that energy-related controls in urban areas can be implemented. The following section shows the schema development approach in detail.

2.2 Basic Outline of the Urban Prosumer Schema

The main goal of the urban prosumer (UP) schema development is to facilitate the rapid specification of energy networks and their automated mapping to a simulation platform. In this, the particular focus is on the effective handling of controls in urban energy networks. The translation to a simulation model puts emphasis on the following requirements (1) a model of the energy flows in the nodes and between nodes i.e., through network branches, and (2) a model of the way specific system and control parameters can be set so that local and global energy targets are met. As Figure 3 shows, the dual emphasis is conceptually reflected in the schema by two separate layers: a physical and a logical layer. To implement the controls in the logical layer effectively, it is vital to capture their internal logic (time precedence, causality, conflict resolution, etc.) as well as their links to the physical elements and their attached sensors and actuators.

This is expressed in multiple relationships that aim to cover conceivable urban energy control strategies.



(a) (b) (c)

Figure 3. (a) Example of an Urban Energy Setting; Hypothetical Operation Scenario (b) Physical Layer (c) Logical Layer

In order to better understand to what extent we should conceptualize urban energy scenarios, we start with a hypothetical case, as shown in Figure 3a. The example scenario includes two buildings and one PV system. The PV system and distributed electricity grid supply electricity to the two buildings, and the PV system can sell back the surplus electricity (when generation is larger than the sum of the building usages) to the electricity grid. As Figure 3b illustrates, we borrow from the directed graph theory to represent this type of energy scenario effectively. Each node represents a physical network component (e.g., building, PV), and each arc represents the direction of energy flow. On top of the physical layer, it can be inferred that another type of node/arc model, i.e., a logical layer, is required to conceptualize the control dependencies. Figure 3c depicts the logical layer, which senses the existing states in the physical layer and "actuates" physical components in the existing physical layer based on the results from the logical layer.

At the very minimum, the UP schema should be able to capture the physical components (nodes) of the energy network and represent their energy flow directions to translate an existing setting into a simulation model. Notably, the UP schema contains a particular subclass of physical components, sensors, actuators that are connected in the logical layer through logical statements, optimal control, and design optimization elements.

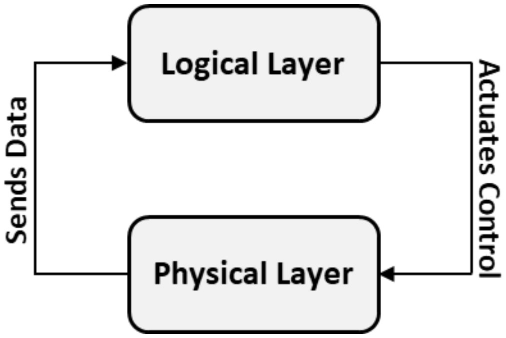


Figure 4. High-level View of UP Schema

Figure 4 illustrates how this high-level view is reflected in the UP schema. The physical layer conceptualizes existing urban energy networks making use of directed graphs for energy transport between nodes.

The logical layer conceptualizes that, based on data obtained from a set of the physical layer, processing that data and involved “controlling” takes place to send instructions to actuators.

At the initial version of development, the rule-based controlling object is kept at a low level of complexity. It is, in fact, limited to an ordered list of simple unitary control expressions, which take the form: “if condition1 and condition and … then change actuator state 1 to X and actuator state 2 to Y and …”. It is clear that this is a simplified version of the actual control that needs to take place in real networks. However, for now, we assume that such simple control based on only current observed states and driven by an ordered list of causal logic expressions (that exhibit no internal conflicts) is a form of control that can make sense. Furthermore, the logical layer also contains optimal control and design optimization elements by exchanging necessary data with the physical layer.

Note that at the schema level, this can be modeled at a high conceptual level, as shown in Figure 4. At instantiation time, the physical layer is populated with components, including attached user-specified sensors and actuators. The rule-based control logic can now be specified as a set of logical expressions that link actuator actions to one or more sensor signals from one or more instantiated components. The population of the expressions is supported by an Excel input file that lets the modeler pick sensors and actuators anywhere in the system and embedding links to them into a list of logical expressions. At this version of the schema, the modeler decides which sensors and actuators play a role in executing a certain operational action. The action itself will be taken by "reasoning" modules embedded in the software platform. The current version of the software is limited to closed-loop instantaneous feedback controls as limited by their current definition in the UP schema. This limitation is reflected in the reasoning modules that read and execute the logic. In addition, the optimal control and design optimization elements conceptualize required input data for the optimizations. However, this version of the schema doesn't incorporate diverse objective functions. Thus this part of the work is left for future schema development work. The current version of developed software conducts the optimization to minimize total costs. The way the schema and software work jointly will be explained more in detail in CHAPTER 5.

The design philosophy of the UP schema is as follows.

- **Scope**: the UP schema should be able to conceptualize the urban energy network comprehensively and should be able to capture urban control strategies as needed.
- **Broad Applicability**: the UP schema should be general enough to map a UP instance to native representations in third-party urban energy simulation or visualization software.
- **Extensibility**: the modular structure of the UP schema should ensure easy extension for future new technologies and more advanced control logic.

In terms of the selection of a schema modeling language, it was decided to develop the UP schema in eXensible Markup Language (XML). An XML schema developed by the W3C is intended to describe the structure and constrain the content of "documents" written in XML by providing rich structuring capabilities. An XML schema can specify the exact element hierarchy and various types of constraints placed on the XML data definitions (Bikakis et al. 2013).

2.3 Physical Layer

This section is dedicated to explaining how the physical layer conceptualizes urban energy networks, e.g., their physical components and their relationships.

2.3.1 Hierarchies in the Physical Layer Element

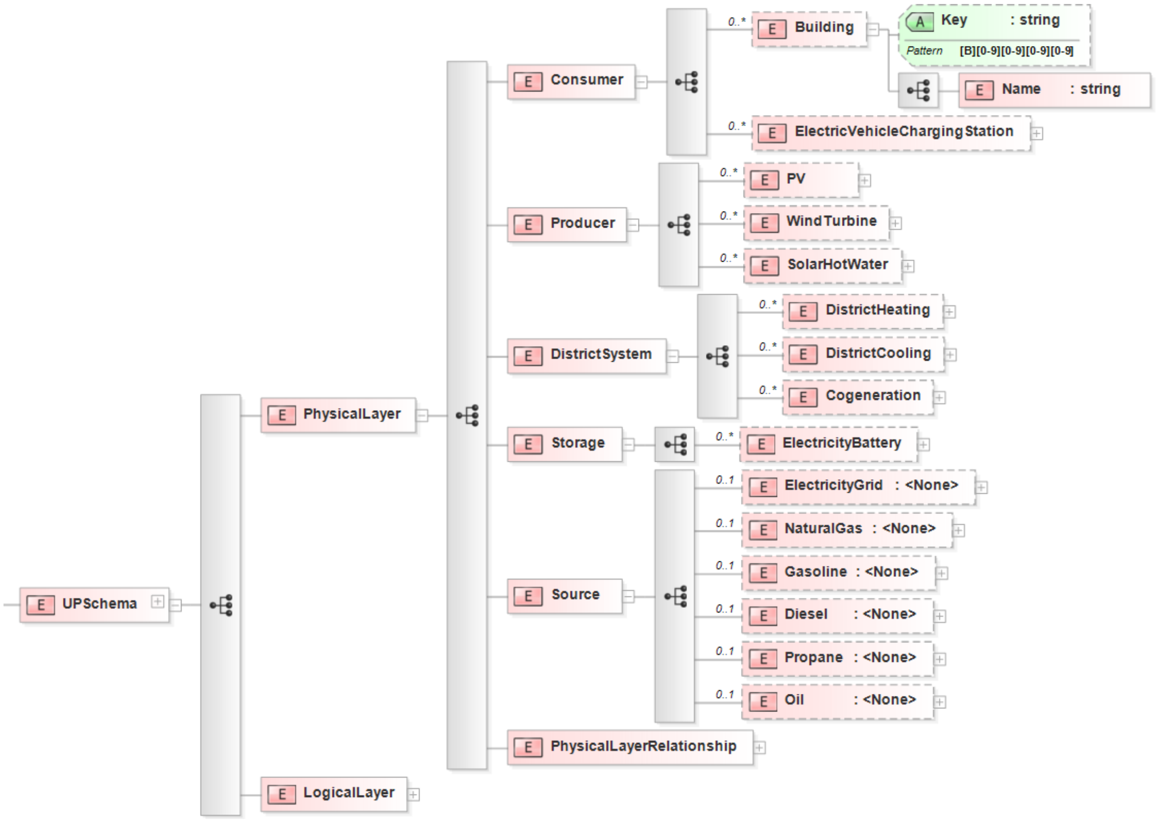


Figure 5. Elements in the Physical Layer (Other physical elements also contain the "Name" child elements and the "Key" attribute)

As Figure 5 illustrates, the root element is `UPSchema`, and the root element has two child elements, `PhysicalLayer` and `LogicalLayer` elements. The `PhysicalLayer` element has the following child elements.

- **`Consumer`**: is the element that consumes different forms of energies supplied from other elements such as the `producer`, `DistrictSystem`, `storage`, and `source` elements. The `Consumer` element has the `Building` and `ElectricVehicleChargingStation` as child elements.
- **`Producer`**: is the element that produces different types of energies from renewable/non-renewable sources. The `Producer` element involves the

`PV`, `WindTurbine`, and `SolarHotWater` elements. The `PV` and `SolarHotWater` elements represent not an individual PV or solar collector modules but a group of PV or solar collector modules.

- **`DistrictSystem`**: is the element that provides thermal energy sources for multiple `Building` elements in the proximity of energy consumers. The `DistrictSystem` element includes `DistrictHeating`, `DistrictCooling`, and `Cogeneration` child elements.
- **`Storage`**: is the element that captures energy produced at one time for use at a later time. The `Storage` elements have the `ElectricityBattery` child element.
- **`Source`**: represents source energy which provides energy to other physical components such as `DistrictHeating`, `DistrictCooling`, etc. The `source` element contains `ElectricityGrid`, `NaturalGas`, `Gasoline`, `Diesel`, `Propane`, and `Oil`. The `ElectricityGrid` element refers to distributed power grids.

Each physical component instance has a terse alphanumeric key (e.g., B1234) to identify a unique physical component. Its key is used for specifying relationships in the instantiation of the physical layer. Further details will be explained in Section 2.3.2.

It should be noted that the UP schema is easily extendable. For instance, if an urban energy analyst would like to add a fuel cell technology, the UP schema can simply be extended to add the technology in the `Producer` element. The current schema contains the structures to define the necessary linkages with other nodes

2.3.2 Relationships in the Physical Layer

The relationships in the `PhysicalLayer` conceptualize energy flows (the arcs in Figure 6) between the physical components in the urban energy network. It is essential to represent energy flows in the UP schema because energy consumers such as `Building`, `ElectricVehicleChargingStation`, etc., can operate only if the right amount and right type of energy are satisfied by the global or local suppliers. For example, Figure 6 illustrates that the suppliers (S1 and S2) provide power to the consumers (C1 and C2). The electricity battery (EB1) is charged from S2 and is discharged to C2.

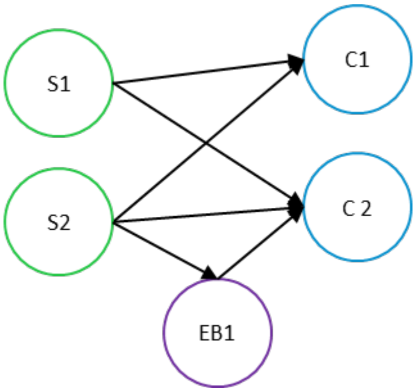


Figure 6. Urban Energy Network Representation in a Directed Graph

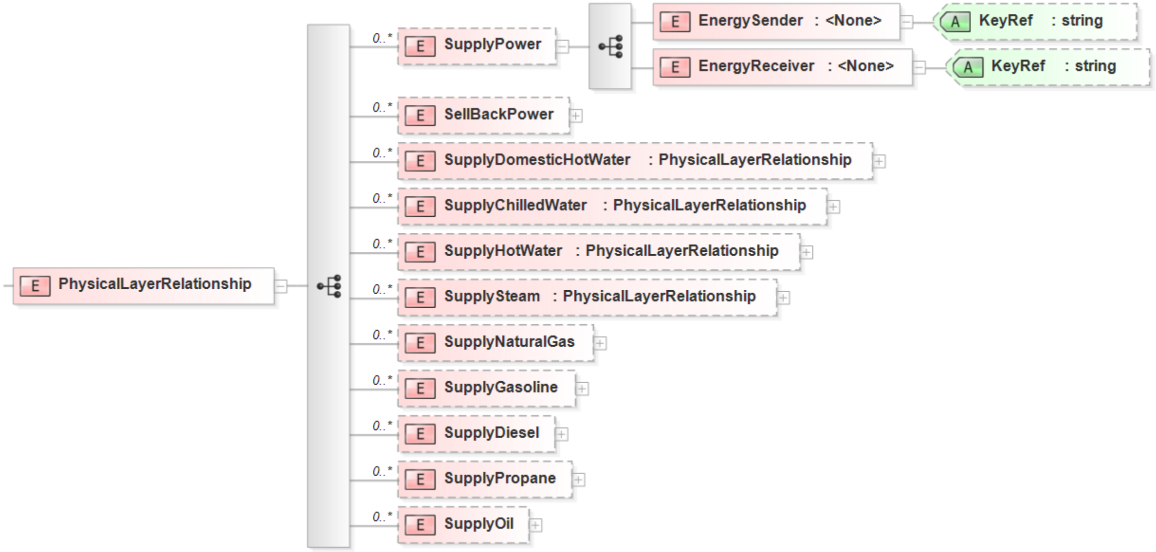


Figure 7. Relationships in the Physical Layer (Other relationships also include the "PhysicalLayerRelationship" complex type)

As Figure 7 depicts, we construct essential relationships based on the variety of physical energy carriers (e.g., power, domestic hot water, chilled water, hot water, steam, natural gas, etc.) in the energy flows. The relationship is represented by specifying `EnergySender` and `EnergyReceiver` by referencing the alphanumeric key from each physical component.

Table 1. List of Relationships in the Physical Layer and Their Definitions (M:N: many-to-many relationship, N:1: Many-to-one relationship, 1:N: one-to-many relationship)

Relationship	Available Power Sender Element(s)	Available Power Receiver Element(s)	Relationship Cardinality
SupplyPower	`PV, WindTurbine, Cogeneration, ElectricityBattery, ElectricityGrid`	`Building, ElectricVehicleChargingStation, DistrictHeating, DistrictCooling, Cogeneration, ElectricityBattery`	M:N
SellBackPower	`PV, WindTurbine`	`ElectricityGrid`	N:1
SupplyDomesticHot Water	`SolarHotWater`	`Building`	M:N
SupplyHotWater	`DistrictHeating, Cogeneration`	`Building`	M:N
SupplySteam	`DistrictHeating, Cogeneration`	`Building`	M:N
SupplyChilledWater	`DistrictCooling`	`Building`	M:N
SupplyNaturalGas	`NaturalGas`	`Building, DistricHeating, DistrictCooling, Cogeneration`	1:N
SupplyGasoline	`Gasoline`	`DistricHeating, DistrictCooling, Cogeneration`	1:N
SupplyDiesel	`Diesel`	`DistricHeating, DistrictCooling, Cogeneration`	1:N

SupplyPropane	`Propane`	`DistricHeating, DistrictCooling, Cogeneration`	1:N
SupplyOil	`Oil`	`DistricHeating, DistrictCooling, Cogeneration`	1:N

Table 1 tabulates the list of the physical layer relationships. As the name of the individual relationship indicates, it is straightforward to know what each relationship means. For instance, the `SupplyPower` relationship suggests that electrical power is supplied from one element (e.g., `PV`) to another element (e.g., `Building`). Table 1 also tabulates which element(s) can be located in `EnergySender` or `EnergyReceiver`.

`SupplyPower`, `SupplyDomesticHotWater`, `SupplyHotWater`, and `SupplySteam` relationships have a many-to-many cardinal relationship because these relationships can supply their electrical or thermal power to multiple receivers, and receivers also can take corresponding power from various suppliers. For example, for a `PV` element and a `Building` element relationship, a building element can be supplied electrical power from many `PV` elements. Similarly, a `PV` element can provide electrical power to many building elements. On the other hand, `SupplyNaturalGas`, `SupplyGasoline`, `SupplyDiesel`, `SupplyPropane`, `SupplyOil` relationships have one-to-many cardinality because these energy sources are represented as one node in the UP schema. The `SellBackPower` relationship has only a one-to-many cardinal relationship because many electrical power suppliers can sell surplus electrical power to the electricity grid, and the electricity grid is conceptualized as one node.

2.4 Logical Layer

The goal of the logical layer is to enable control actions over "controllable elements" of the instantiated physical components or the relationships between them.

2.4.1 Hierarchies in the Logical Layer Element

The logical layer of the schema conceptualizes controls in microgrids.

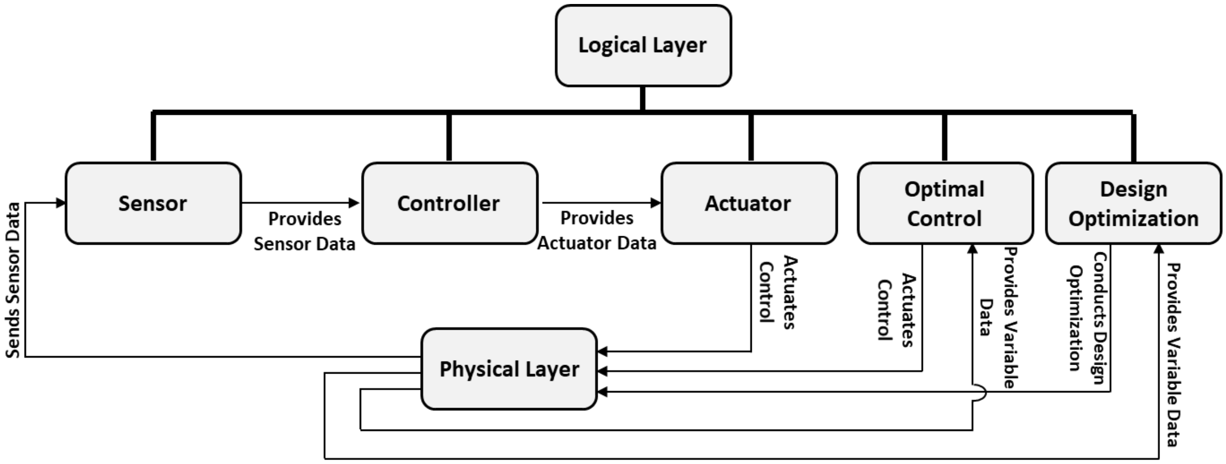


Figure 8. Specialized Logical Layer and Relationship with the Physical Layer

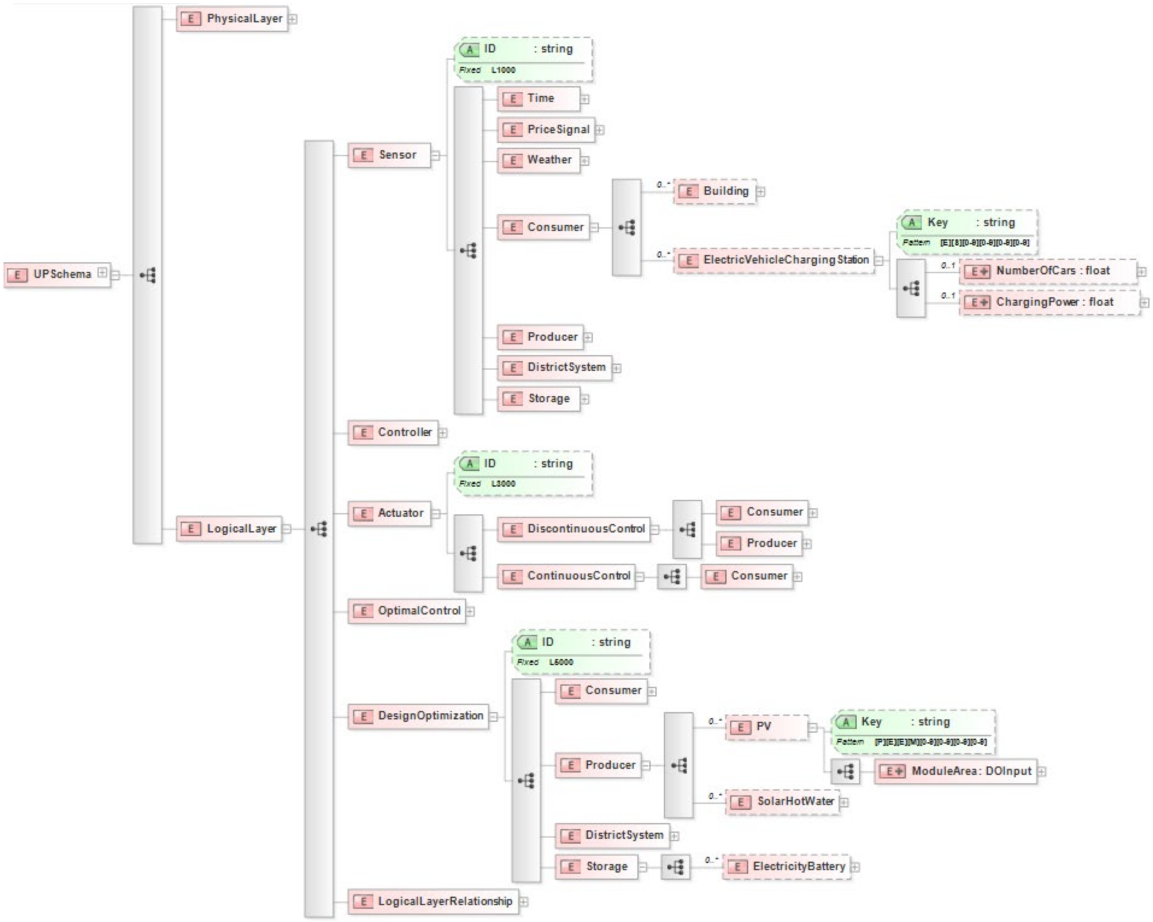


Figure 9. Structure of the Logical Layer

We further specialize inside the logical layer to define `Sensor`, `Controller`, `Actuator`, `OptimalControl`, and `DesignOptimization` elements, as shown in Figure 8 and Figure 9. The `Sensor` element measures the value of a specific observable state variable, such as temperature, solar radiation, etc., of a specific component at a given location in the physical layer. The `Controller` element receives input data from `Sensor` elements, processes the input data, and then produces an actuator signal that activates the actuator's physical components. The `Actuator` element acts to modify controlled variables as directed by the `Controller` element. It should be noted that the `Sensor` element includes both observable states measured from actual sensors as well as simulation outputs such as electricity use or electricity generation. As it can be inferred, we use the term ‘sensor’ in the broad sense and hence do not distinguish between sensor and meter. The `OptimalControl` element refers to finding an optimal control input for a dynamical system over a period of time to optimize the objective function. The way the `OptimalControl` element is conceptualized in the `UPSchema` is that the `OptimalControl` element firstly receives the optimization variable data from the corresponding element in the `PhysicalLayer`.Then, the optimal variable results are transferred to the corresponding element in the `PhysicalLayer` after the optimization is conducted. The `DesignOptimization` element indicates the selection of the optimal design among many alternatives using a mathematical formulation. The `DesignOptimization` element works similarly to the `OptimalControl` element. The `DesignOptimization` element also receives the optimization variable data from the `PhysicalLayer`. Then, the optimization result is given back to the `PhysicalLayer` to reflect the best design decision in the given energy scenario.

Furthermore, it should be noted that given the (in most cases) low resolution of the executable component models (packaged as software modules), the “physical position and meaning” of a particular sensor or actuator inside the component models in the `PhysicalLayer` is not always identifiable with an actual physical sensor or actuator. In other words, sensors and actuators in the real setting may, in some cases, only have an approximate association with an outcome that is simulated (and hence sensible) and with what is a model parameter (and hence controllable) in the building. This is the direct result of the mismatch in resolution between the physical world and the simulated world. This is not desirable but beneficial in terms of the reduced complexity resulting from the reduced-order modeling of the nodes in the energy network. So, although we deal with actual physical actuators in control statements, the actuator in a reduced-order simulation is a partly abstract surrogate for this. The following sections 2.4.1.1 and 2.4.1.3 describe available sensors and actuators in the current reduced-order simulation model. Section 2.4.1.2 explains more about the ramifications for the controller side.

2.4.1.1 Sensors

Below, currently available sensors in the urban energy network simulation are enumerated.

- **`Time`**: observes the time of the simulation. The `Time` element has five child elements, (1) Time step, (2) Date (1~365), (3) Hour (1~24), (4) Day Type (Weekday: 0, Weekend:1).

Table 2. List of the `Time` Sensors

Sensor	Note
`Time Step`	Time step of simulation
`Date`	Cumulative day of the year
`Hour`	Hour of the day
`Day Type`	Weekday/Weekend

- **`Weather`**: measures outside conditions pertaining to weather.

Table 3. List of the `Weather` Sensors

Sensor	Units
Dry Bulb Temperature	°C
Wind speed	m/s
Direct Normal Radiation	W/m^2
Diffuse Horizontal Radiation	W/m^2
Horizontal Solar Radiation	W/m^2
Specific Humidity Dry Air	g/kg
Atmospheric Station Pressure	Pa
Relative Humidity	%

- **`Consumer`**: observes measurable states from the `Building` and the `ElectricVehicleChargingStation` elements.

Table 4. List of the `Consumer` Sensors

Building		**Electric Vehicle Charging Station**	
Sensor	Units	Sensor	Units
Occupancy	Persons	Number of Cars	ea.
Thermal Zone Air Temperature	°C	Charging Power	kW
Thermal Heating Need	kWh		
Thermal Cooling Need	kWh		
Delivered Energy Use	kWh		
Heating Delivered Energy Use	kWh		
Cooling Delivered Energy Use	kWh		
Lighting Delivered Energy Use	kWh		
Fan Delivered Energy Use	kWh		
Pump Delivered Energy Use	kWh		
Appliance Delivered Energy Use	kWh		
DHW Delivered Energy Use	kWh		
Electrical Energy Use	kWh		
Natural Gas Use	kWh		

- **`Producer`**: observes measurable states from the child elements of the `Producer` element.

Table 5. List of the `Producer` Sensors

PV		SHW		Wind Turbine	
Sensor	Units	Sensor	Units	Sensor	Units
`Electricity Generation`	`kWh`	`Hot Water Generation`	`kWh`	`Electricity Generation`	`kWh`

- **Storage**: observes measurable states from the `ElectricityBattery` element.

Table 6. List of Storage Sensors

Electricity Battery	
Sensor	Units
`State`	`%`
`Charging Power`	`kW`
`Discharging Power`	`kW`

- **`PriceSignal`**: observes marginal energy cost.

Table 7. List of the `PriceSignal` Sensors

Price Signal	
Sensor	Units
`Electricity Price`	`$/kWh`
`Natural Gas Price`	`$/kWh`
`Gasoline Price`	`$/kWh`
`Diesel Price`	`$/kWh`
`Propane Price`	`$/kWh`
`Oil Price`	`$/kWh`

- **`DistrictSystem`**: observes the measurable states from the child elements of the `DistrictSystem` element.

Table 8. List of the `DistrictSystem` Sensors

District Heating		District Cooling		Cogeneration	
Sensor	Units	Sensor	Units	Sensor	Units
`HotWaterGeneration`	kWh	`Chilled Water Generaton`	kWh	`HotWaterGeneration`	kWh
`Steam Generation`	kWh	`Electricity Use`	kWh	`Steam Generation`	kWh
`Electricity Use`	kWh	`Natural Gas Use`	kWh	`Electricity Generation`	kWh
`Natural Gas Use`	kWh	`Gasoline Use`	kWh	`Electricity Use`	kWh
`Gasoline Use`	kWh	`Diesel Use`	kWh	`Natural Gas Use`	kWh
`Diesel Use`	kWh	`Propane Use`	kWh	`Gasoline Use`	kWh
`Propane Use`	kWh	`Oil Use`	kWh	`Diesel Use`	kWh
`Oil Use`	kWh			`Propane Use`	kWh
				`Oil Use`	kWh

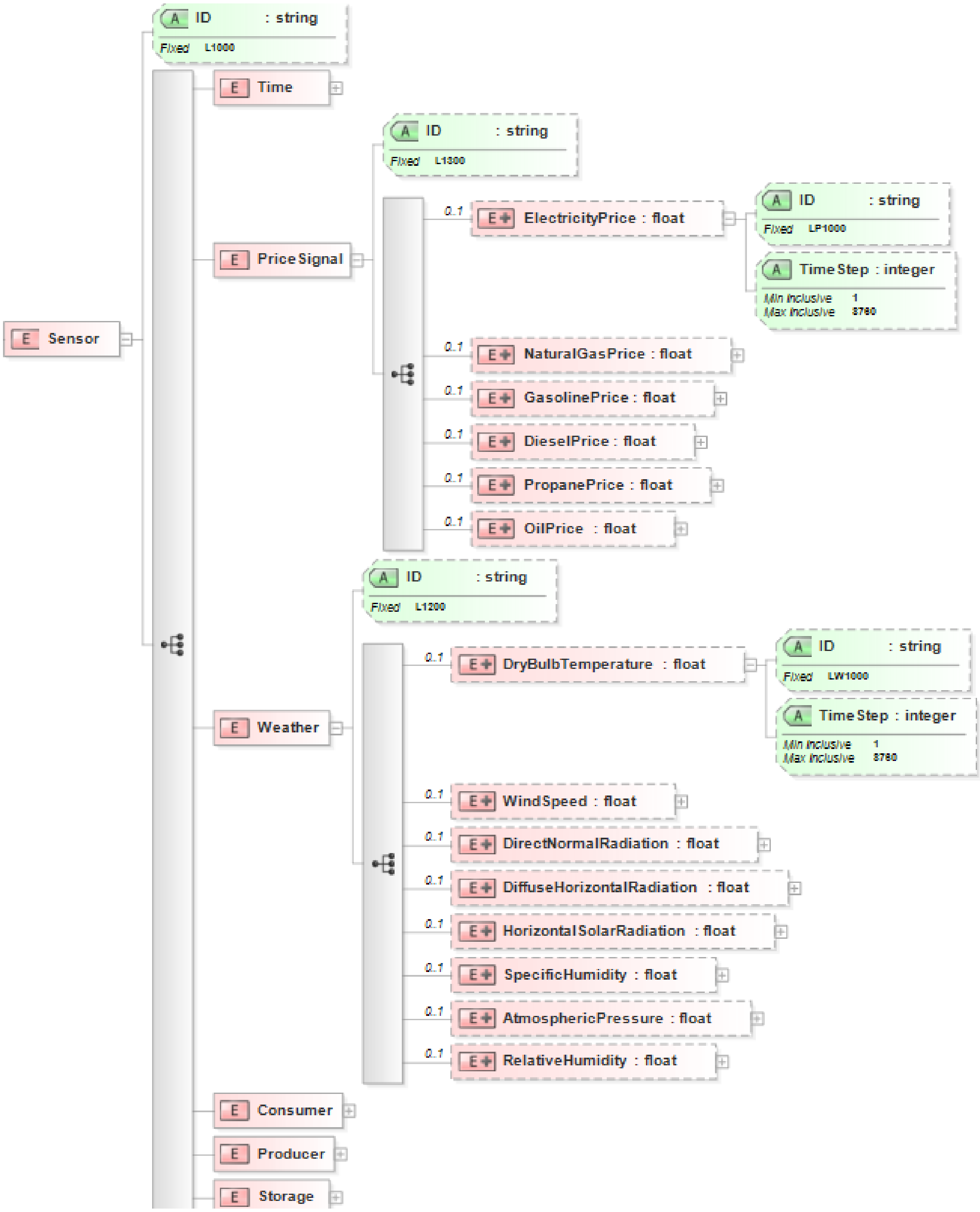


Figure 10. Weather and PriceSignal Sensors (Weather sensors have their unique IDs)

Figure 10 shows the `Weather` and `PriceSignal` elements under the `Sensor` element. The elements under the `Weather` element (e.g., `DryBulbTemperature`, etc.) and the `PriceSignal` element have alphanumeric IDs (e.g., LW1000, LP1000).

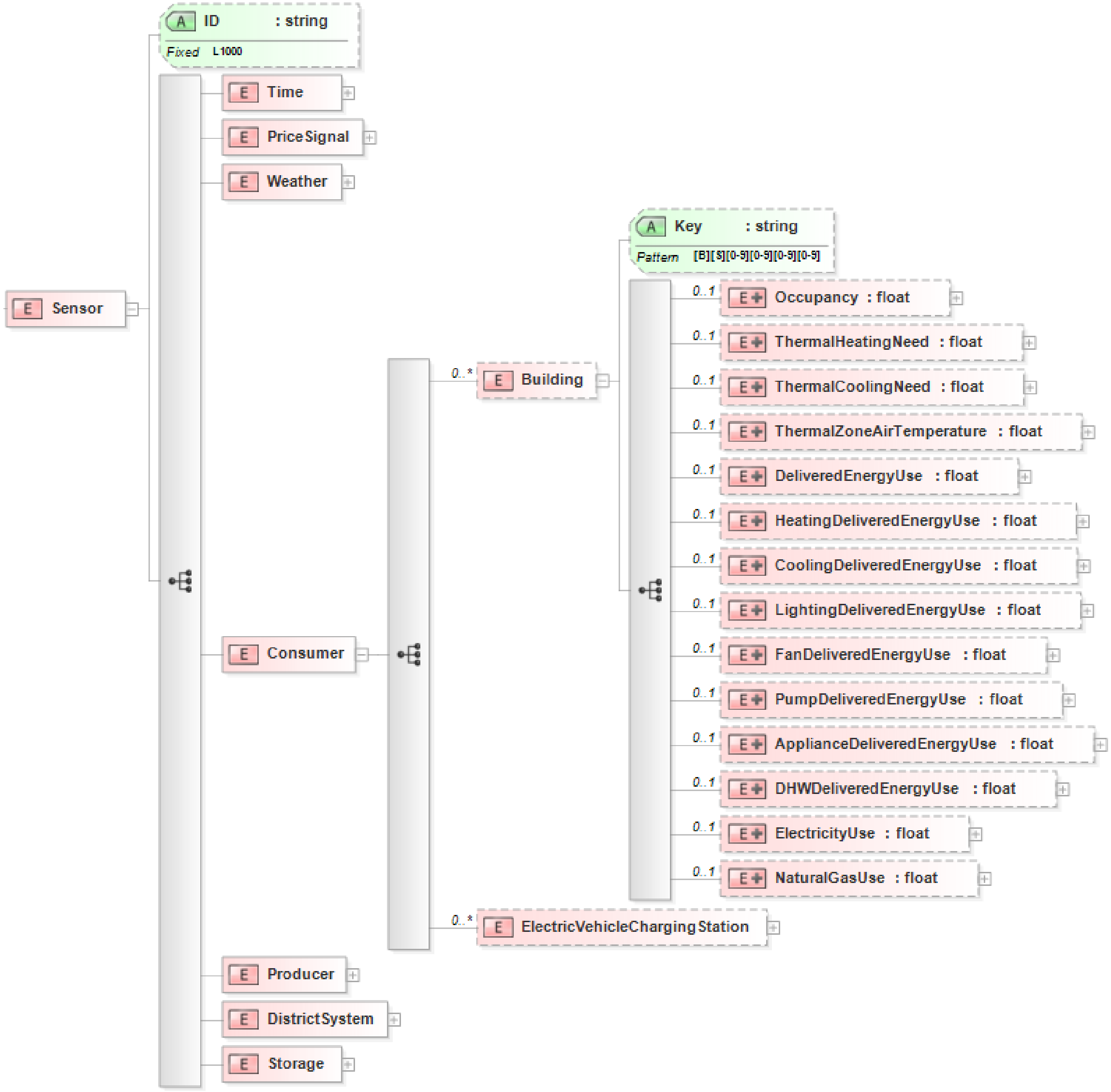


Figure 11. Building Sensors (Each Building sensor has `Key` and `TimeStep` attributes.)

As in Figure 11, each physical element's sensor contains its alphanumeric key for the other sensors. For example, the `Occupancy` sensor has its key consisting of a BSI string value with a 4-digit number.

2.4.1.2 Controller

```
if condition then
    Statement 1
else if condition 1 then
    Statement 2
else
    Statement 3
end if
```

Figure 12. Logical Layer Input Pseudocode

As shown in Figure 12, the Controller element is designed to operate based on rule-based logic. With the rule-based input logics, the UP schema facilitates rapid building stock analyses with different sets of logics by creating UP instances for each set of logic and executing the simulation.

```
1: if 2881 < CumulativeHour < 6552 and 12 <= Hour <= 18 and DayType == 1 and DryBulbTemperature > 27
2:     BuildingA CoolingSetPoint = 27;
3: else
4:     BuildingA CoolingSetPoint = Original;
```

Figure 13. Example of Rule-based Logic in the Controller Element

An example of rule-based logic in Figure 13 describes a `CoolingSetPoint` control in Building A. Line 1 and 3 are conditions, and lines 2 and 4 are statements. To better visualize, the blue color letters indicate the sensors, and the orange color letters suggest the actuators in the UP schema. The first condition (Line 1) and first statement (Line 2) show that if `DryBulbTemperature` is greater than 27 °C (“`DryBulbTemperature` > 27”) on weekdays (“`DayType` == 1”) between 12 and 18 hours (“12 <= `Hour` <= 18”) from May to September (“2881 < `CumulativeHour` < 6552”), then BuildingA’s `CoolingSetPoint` becomes 27 °C (“BuildingA

`CoolingSetPoint = 27;`”). If the first condition is not met, the building’s cooling setpoint is left as-is. The “Original” keyword makes it easier to deal with conditions that have many else-if cases. Without the “Original” keyword, it would have been difficult and confusing to make else-if cases because a modeler has to deal with else-if cases for the `CumulativeHour`, `Hour`, `DayType`, `DryBulbTemperature` sensors in the first condition.

2.4.1.3 Actuators

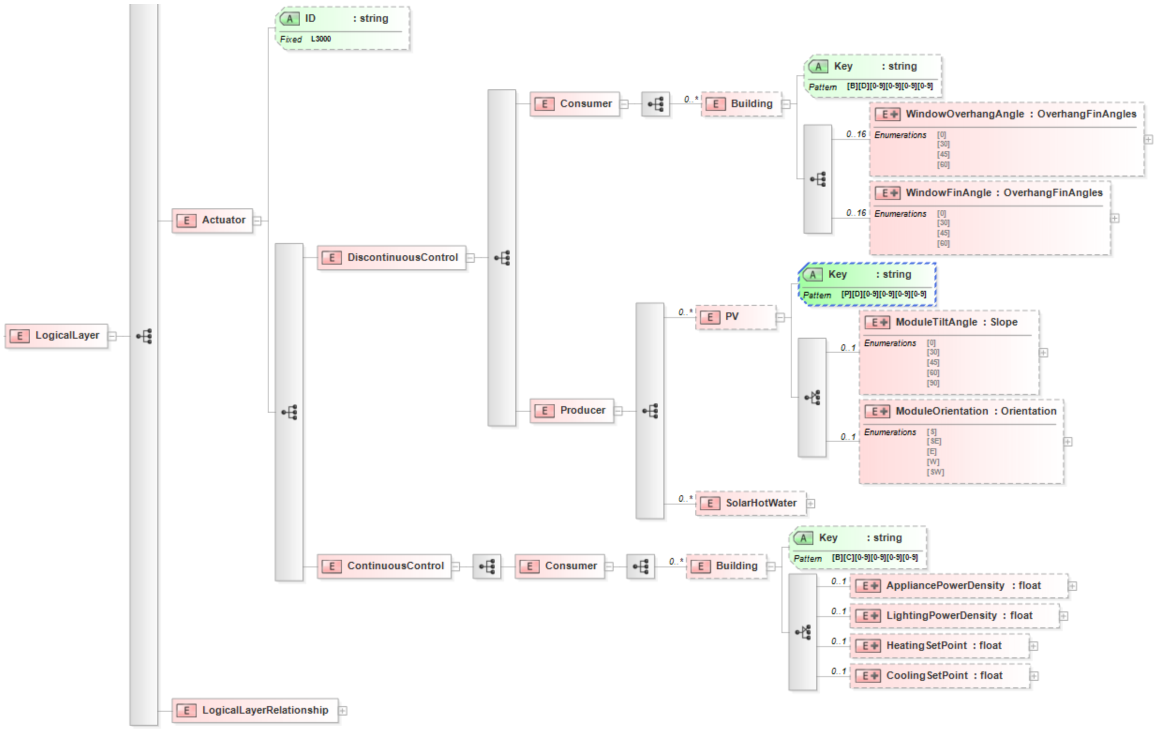


Figure 14. Structure of the Actuator Element

As Figure 14 depicts, the logical layer also contains the `Actuator` element that is further divided into the `DiscontinuousControl` and `ContinuousControl` elements. The `DiscontinuousControl` element modifies controlled variables with discrete values, and the `ContinuousControl` element modulates controllable variables with continuous values. Current available discontinuous and continuous controllable variables are introduced below.

- **`Consumer`**: involves the available actuators from the `Building` and `ElectricVehicleChargingStation` elements.

Table 9. List of Available Actuators in the `Building` Element

Building				
Discontinuous Actuator	Available Values	Units	Continuous Actuator	Units
`Window Overhang Angle`	`[0,30,45,60]`	°	`Appliance Power Density`	`W`
`Window Fin Angle`	`[0,30,45,60]`	°	`Lighting Power Density`	`W`
			`Heating Set Point`	°C
			`Cooling Set Point`	°C

- **`Producer`**: involves available actuators in the `PV` and `SolarHotWater` elements. The `WindTurbine` element does not have an actuator.

Table 10. List of Available Actuators in the `Producer` Element

PV				
Discontinuous Actuator	Available Values	Units	Continuous Actuator	Units
`Module Orientation`	`[W,SW,S,SE,E]`	-	`N/A`	-
`Module Angle`	`[0,30,45,60,90]`	°		
Solar Hot Water				
Discontinuous Actuator	Available Values	Units	Continuous Actuator	Units
`Collector Orientation`	`[W,SW,S,SE,E]`	-	`N/A`	-
`Collector Angle`	`[0,30,45,60,90]`	°		

2.4.1.4 Optimal Control

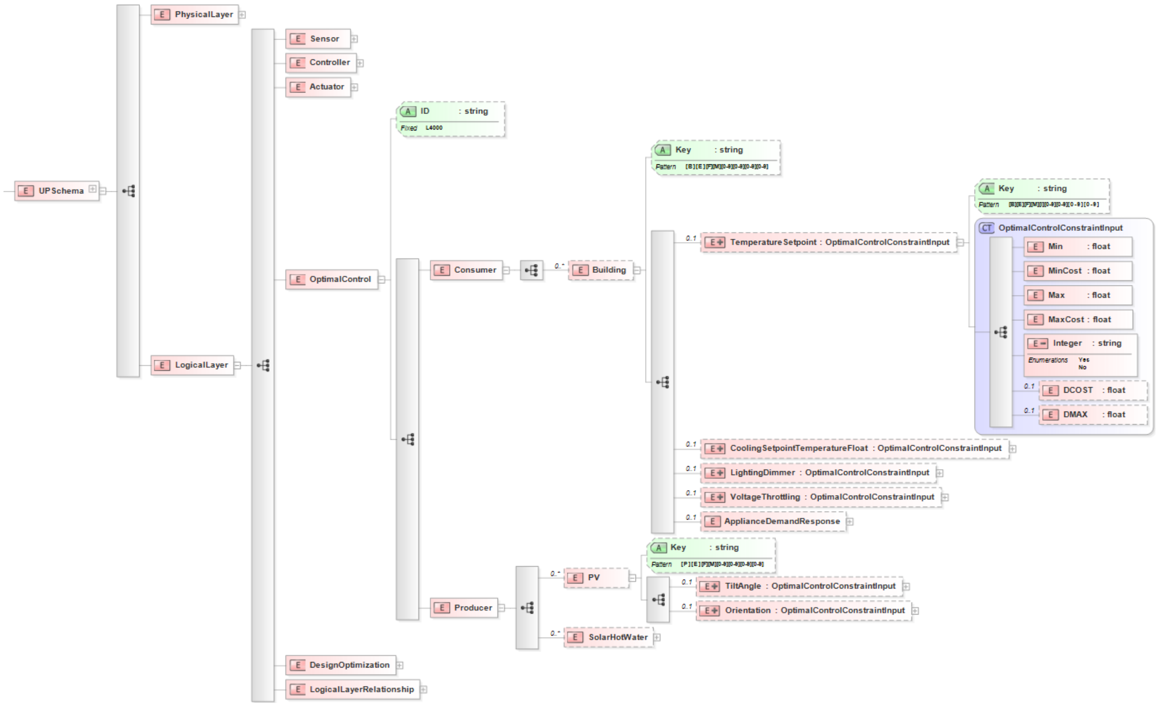


Figure 15. Structure of the `OptimalControl` Element

Figure 15 illustrates the available optimal control variables in each element. The variables under the `OptimalControl` elements include a few sub-elements that are needed for optimal control optimization. More specifically, the `Min/Max` and `MinCost/MaxCost` elements specify the minimum and maximum variable's interval and corresponding minimum and maximum variable's cost, respectively. The Integer element indicates whether the selected variable is an integer type variable or not. The `DCOST` (delta cost) is a minor penalty in the objective function for changing the variable values unnecessarily. This is especially useful if the selected variable is damaged from excessive movement. The `DMAX` (delta max) element gives a hard constraint that prevents the variable from being changed by more than the specified value in a one-time step. The following explains available variables for the optimal control in each physical element.

- **`Consumer`**: involves the available optimal control variables from the `Building` element.

Table 11. List of Available Optimal Control Variables in the `Building` Element

Building	
Variable	Units
`Temperature Setpoint*`	°C
`Cooling Setpoint Temperature Float*`	°C
`LightingDimmer`	-
`VoltageThrottling`	-
`ApplianceDemandResponse`	-

*-Temperature Setpoint is used for the optimal control and Cooling Setpoint Temperature Float is used in the parameterized optimal control

- **`Producer`**: includes available optimal control variables from the `PV` and `SolarHotWater` elements.

Figure 16. List of Available Optimal Control Variables in the `Producer` Element

PV		**Solar Hot Water**	
Variable	Units	Variable	Units
`Tilt Angle`	°	`Tilt Angle`	°
`Orientation`	°	`Orientation`	°

2.4.1.5 Design Optimization

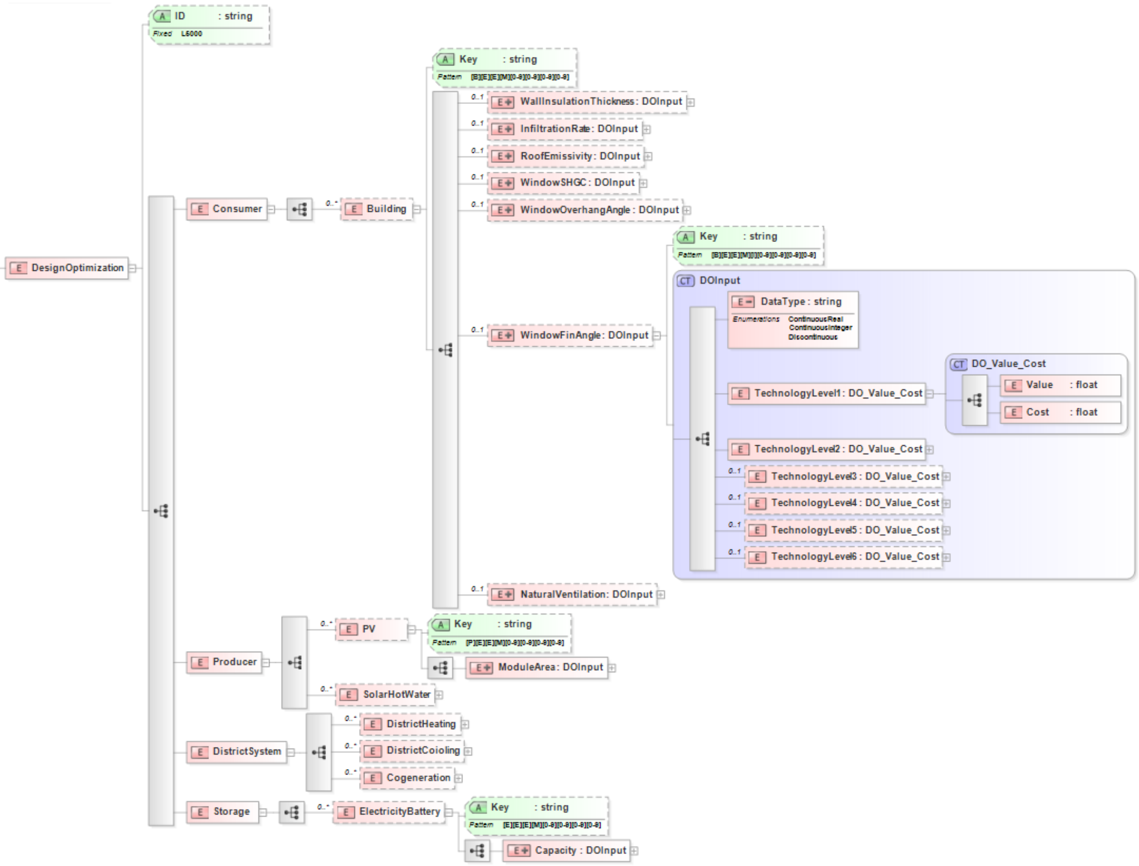


Figure 17. Structure of the `DesignOptimization` Element

Figure 17 shows the `DesignOptimization` element in more detail. The `DesignOptimization` element includes the physical component that includes design optimization variables. The variables are used to find the optimum technology mix that minimizes the total cost within a time horizon. Each design optimization variable is allowed to have 2 technology mixes for a continuous type variable and up to 6 different technology mixes for a discontinuous type variable. This can be set up in the `DataType` element.

- **`Consumer`**: involves the available design optimization variables from the `Building` element.

Table 12. List of Available Design Optimization Variables in the `Building` Element

Building	
Variable	Units
`Wall Insulation Thickness`	cm
`Infiltration Rate`	$m^3/h/m^2$
`RoofEmissivity`	-
`WindowSHGC`	-
`WindowOverhangAngle`	°
`WindowFinAngle`	°
`NaturalVentilation`	m^2

- **`Producer`**: involves the available design optimization variables from the `PV` and `SoalrHotWater` elements.

Figure 18. List of Available Design Optimization Variables in the `Producer` Element

PV		Solar Hot Water	
Variable	Units	Variable	Units
`ModuleArea`	m^2	`ModuleArea`	m^2

- **`DistrictSystem`**: involves the available design optimization variables from the `DistrictHeating`, `DistrictCooling`, and `Cogeneration` elements.

Figure 19. List of Available Design Optimization Variables in the `DistrictSystem` Element

District Heating		District Cooling		Cogeneration	
Variable	Units	Variable	Units	Variable	Units
`Heating COP`	-	`Cooling COP`	-	`Heating COP`	-

- **`Storage`**: involves the available design optimization variables from the `ElectricityBattery` element.

Figure 20. List of Available Design Optimization Variables in the `Storage` Element

Electricity Battery	
Variable	Units
`Capacity`	kWh

2.4.2 Relationships in the Logical Layer

The relationships in the `LogicalLayer` element conceptualize how the measured sensor variable data should be transferred to the `controller` element so that the `controller` element can generate control actions. The `actuator` element should receive the `controller`'s output signal so that the `actuator` element can act to modify the value of controlled variables in the `PhysicalLayer` element. Furthermore, the relationships enable optimal control and design optimization by exchanging necessary data between the `PhysicalLayer` and LogicalLayer.

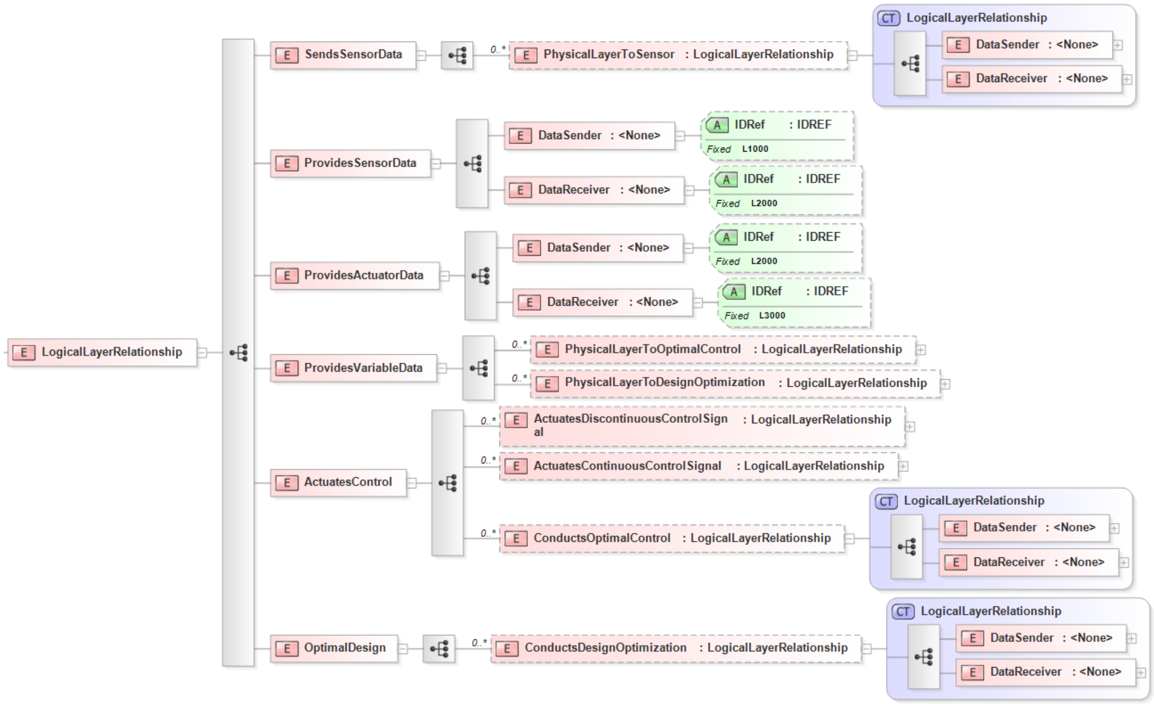


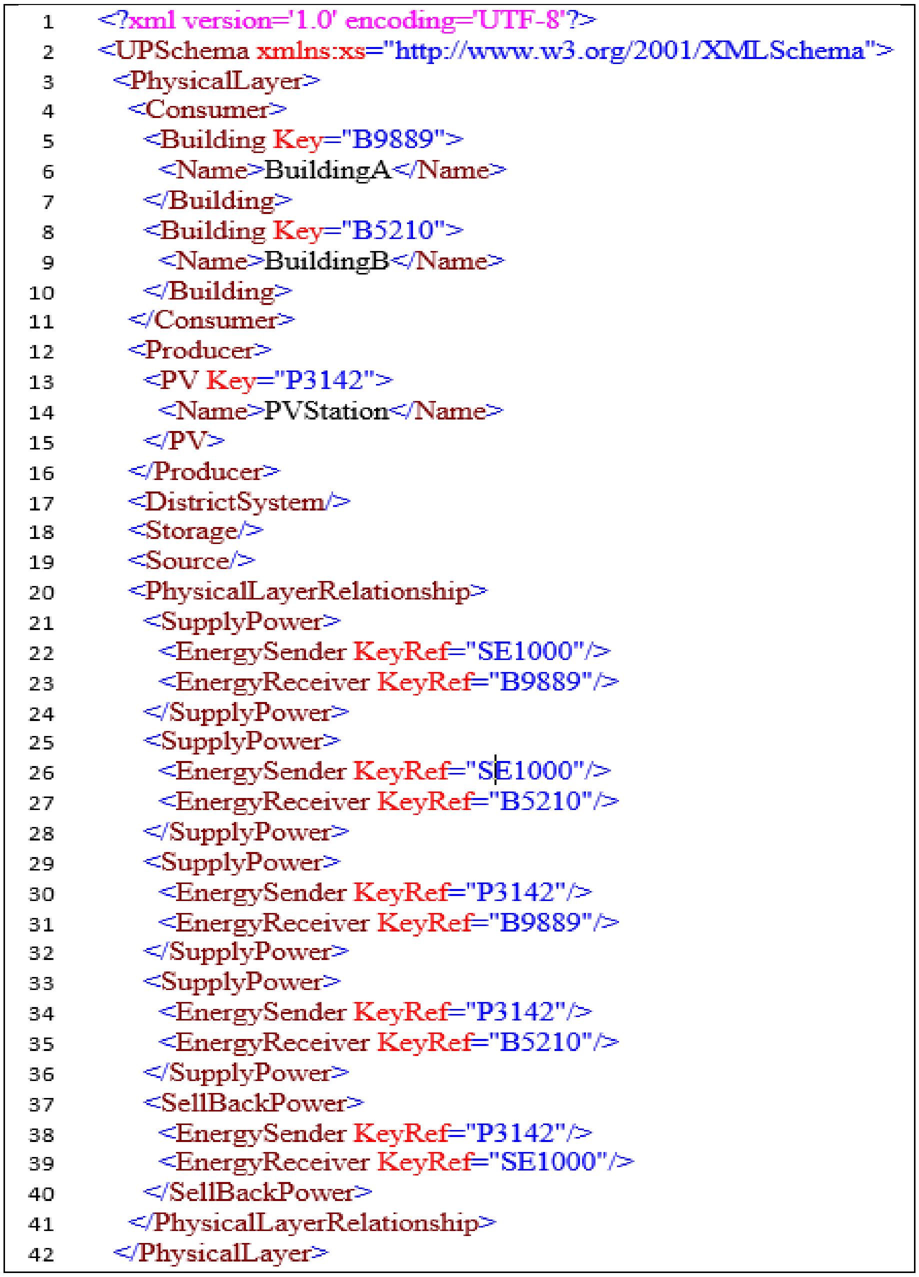
Figure 21. Relationships in the Logical Layer

Figure 21 illustrates the relationships in the `LogicalLayer` element. The `SendsSensorData` relationship matches a physical component in the physical layer and the `Sensor` element in the logical layer by referencing a key in the physical component and the `Sensor` element. As mentioned before, each physical component and the `Sensor` element have its alphanumeric key, respectively. The `ProvidesSensorData` and `ProvidesActuatorData` relationships represent the data transfer from the `Sensor` element to the `Controller` element and from the `Controller` element to the `Actuator` element, respectively. The `ActuatesControl` relationship is further specialized into the `ActuatesDiscontinuousControlSignal`, `ActuatesContinuousControlSignal`, and `ConductsOptimalControl` elements. The first two elements act to modify controlled variables as directed by the `Controller` element. The `ProvidesVariableData` element plays a role in transferring a physical component's instantiated optimal control variable data to the `OptimalControl` and `DesignOptimization` elements. The optimized variable results are sent back to the physical component using the `ConductsOptimalControl` relationship. The `ProvidesVariableData` element also transfers the design optimization variables' data to the `DesignOptimization` element. Similar to the optimal control case, the `ConductsDesignOptimization` relationship under the `OptimalDesign` element gives the optimal design optimization values back to the physical component.

2.5 Application of the UP Schema

In this subsection, a UP instance that describes the hypothetical operation introduced in Figure 3a is generated. The setting consists of two buildings and a PV station.

In Figure 22, two `Building` elements (BuildingA and BuildingB) and one `PV` element (PVStation) are instances of the `PhysicalLayer` element. Every physical element includes its alphanumeric key. The relationships in the `PhysicalLayer` element represent that power flow causality and control in this hypothetical scenario. In this case, the `SupplyPower` and `SellBackPower` relationships represent the energy flows in the given scenario.



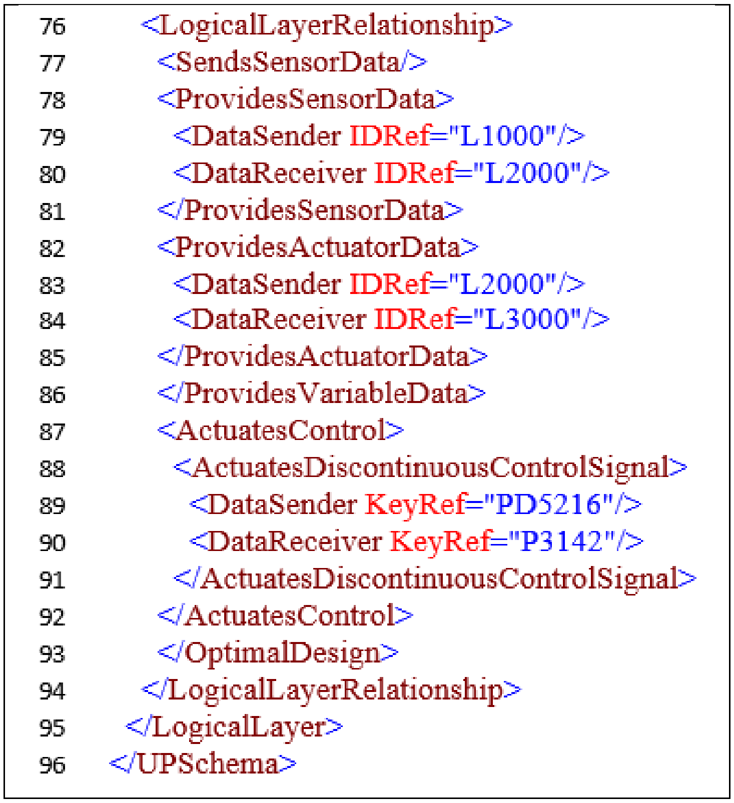
```
<?xml version='1.0' encoding='UTF-8'?>
<UPSchema xmlns:xs="http://www.w3.org/2001/XMLSchema">
 <PhysicalLayer>
  <Consumer>
   <Building Key="B9889">
    <Name>BuildingA</Name>
   </Building>
   <Building Key="B5210">
    <Name>BuildingB</Name>
   </Building>
  </Consumer>
  <Producer>
   <PV Key="P3142">
    <Name>PVStation</Name>
   </PV>
  </Producer>
  <DistrictSystem/>
  <Storage/>
  <Source/>
  <PhysicalLayerRelationship>
   <SupplyPower>
    <EnergySender KeyRef="SE1000"/>
    <EnergyReceiver KeyRef="B9889"/>
   </SupplyPower>
   <SupplyPower>
    <EnergySender KeyRef="SE1000"/>
    <EnergyReceiver KeyRef="B5210"/>
   </SupplyPower>
   <SupplyPower>
    <EnergySender KeyRef="P3142"/>
    <EnergyReceiver KeyRef="B9889"/>
   </SupplyPower>
   <SupplyPower>
    <EnergySender KeyRef="P3142"/>
    <EnergyReceiver KeyRef="B5210"/>
   </SupplyPower>
   <SellBackPower>
    <EnergySender KeyRef="P3142"/>
    <EnergyReceiver KeyRef="SE1000"/>
   </SellBackPower>
  </PhysicalLayerRelationship>
 </PhysicalLayer>
```

Figure 22. UP instance – Physical Layer

```
43  <LogicalLayer>
44    <Sensor ID="L1000">
45      <Time>
46        <Hour>1</Hour>
47      </Time>
48      <PriceSignal/>
49      <Weather>
50        <HorizontalSolarRadiation ID="LW1400" TimeStep="1">0</HorizontalSolarRadiation>
51      </Weather>
52      <Consumer/>
53      <Producer/>
54    </Sensor>
55    <Controller ID="L2000">
56      <Condition>if 7 &lt;= Hour &lt;= 11 and HorizontalSolarRadiation &gt; 1500</Condition>
57      <Statement>PVStation ModuleOrientation = W;</Statement>
58      <Condition>elseif 7 &lt;= Hour &lt;= 11 and 500 &lt; HorizontalSolarRadiation &lt; 1500</Condition>
59      <Statement>PVStation ModuleOrientation = W;</Statement>
60      <Condition>else</Condition>
61      <Statement>PVStation ModuleOrientation = S;</Statement>
62    </Controller>
63    <Actuator ID="L3000">
64      <DiscontinuousControl>
65        <Consumer/>
66        <Producer>
67          <PV Key="PD5216">
68            <ModuleOrientation Key="PDI1002">S</ModuleOrientation>
69          </PV>
70        </Producer>
71      </DiscontinuousControl>
72      <ContinuousControl>
73        <Consumer/>
74      </ContinuousControl>
75    </Actuator>
```

Figure 23. UP instance – Logical Layer Sensor, Controller and Actuator Elements (Cont'd from Figure 22)

Figure 23 depicts both physical and logical layer instances. The logical layer contains `Sensor`, `Controller`, and `Actuator` elements. The `Sensor` element (Line: 44) is comprised of a user-instantiated weather sensor, `HorizontalSolarRadiation`. The `Controller` element (Line: 55) includes the user-input logic, as shown in Figure 13. In the `Condition` element, the input condition and logical statements are instantiated. Since the open and close angle brackets (< and >) are reserved for markup, '<' is represented as "<" and '>' is represented as ">" respectively. The `Actuator` element (Line: 67) involves the user-specified actuator, `ModuleOrientation`.



```
  <LogicalLayerRelationship>
   <SendsSensorData/>
   <ProvidesSensorData>
    <DataSender IDRef="L1000"/>
    <DataReceiver IDRef="L2000"/>
   </ProvidesSensorData>
   <ProvidesActuatorData>
    <DataSender IDRef="L2000"/>
    <DataReceiver IDRef="L3000"/>
   </ProvidesActuatorData>
   </ProvidesVariableData>
   <ActuatesControl>
    <ActuatesDiscontinuousControlSignal>
     <DataSender KeyRef="PD5216"/>
     <DataReceiver KeyRef="P3142"/>
    </ActuatesDiscontinuousControlSignal>
   </ActuatesControl>
   </OptimalDesign>
  </LogicalLayerRelationship>
 </LogicalLayer>
</UPSchema>
```

Figure 24. UP instance – Logical Layer Relationship Elements (Cont'd from Figure 23)

Figure 24 shows the relationships in the logical layer. The `Sensor` (ID: L1000) data is supposed to be sent to the `Controller` (ID: L2000) element through the `ProvidesSensorData` relationship. Similarly, the `ProvidesActuatorData` sends the control signal data from the `Controller` (ID: L2000) to the `Actuator` (ID: L3000) elements.

At this point, it is good to recall that the above instance contains all the needed information to run a simulation for the urban energy network and do this in a completely automated fashion. This is, of course, based on the premise that component modules are provided for all network nodes in the network. The component modules run autonomously while sending sensor data from onboard sensors and receiving controller signals for their onboard actuators. This makes optimal use of previous developments in component module development, which is indeed one of the prime purposes of schema development. So, before we can pursue this further and show the developed software in action, we need to embed the behavioral component models of all nodes in the network. In our example, this is limited to two building modules, and one PV station module as these are the only component types in the network. The two building nodes use the same software module but populated with different data, reflecting the specifics of Building A and Building B. The underlying idea is that in the initial modeling stage, building data can be gathered and parsed and a module for each building is thus instantiated. In the next stage, the energy network is defined as a UP instance, where at a given node of the network, the already applicable premade building module is chosen as the instance.

Each node in the network is represented by a state-space model realized in a computational model and hence software module. Based on the current state, it predicts the state at the next time step. After each time step, the control logic is executed to potentially change the actuators' values (in the instantiated building and PVStation component models and their corresponding simulation modules) before the simulation moves on to the next time step. The next chapter explains the principles and implementation of the component models in the simulation modules that reside in the UP simulation platform.

CHAPTER 3. PHYSICS-BASED MODELS EMBEDDED IN THE SIMULATION PLATFORM

3.1 Introduction

This chapter is dedicated to describing the physics-based simulation models utilized in the simulation platform. They are implemented as component software modules. The choice of component models for the network nodes is driven by the following considerations:

- Order-reduced parsimonious modeling, following established and published approaches. These guarantee run-time efficiency when the number of nodes in the network increases. It is expected that higher fidelity models will not lead to increased fidelity at the global level, although this thesis will not test this assumption but there is enough evidence in the literature that urban energy simulations require only a reasonably coarse granularity of the dynamic response of buildings, which are the main dynamic components in the type of network studies in the scope of this thesis.
- We employ state-space modeling, casting every component behavior as a set of ordinary differential equations:

  $\frac{dx}{dt} = A \cdot x + B \cdot u,\ y = g(x,u);$

  with state x, observation y, and control u.
- x, y, and u are partitioned in (1) a subset that is private to the component and not shared at the global level, and hence not considered within the

scope of the UP schema, (2) a public subset of states and controls that are known and accessible at the global level. The latter is linked to sensors and actuators that are specified in the UP instance.

- Sensors are associated with measurements of an individual or combined elements of the public subset of y at a certain time.
- Actuators are associated with the public subset of u; in some cases, a unique physical actuator (as defined in the UP instance) has a direct match with an element of u that closely represents the actuator in the component model. In other cases, a physical actuator is matched with a control element u, which performs a surrogate control action used inside the component module.
- Actuators receive signals from global and local controllers; if they only receive signals from local controllers, they are not exposed to the global level (remember that local controllers are handled autonomously inside the module). If a controller receives signals from a global controller, it is visible at the global level and hence part of the UP instance. In some cases, an actuator receives a global signal, which is preprocessed at the local component level before an action of the actuator is decided. The actual action that results is based on predefined logic which resides within the component. An illustrative example of this is when the HVAC system receives a DR signal to suspend cooling. A controller device such as an intelligent thermostat decides based on current zone temperatures and other local states whether this in fact leads to an allowed action (i.e. a full or part shutdown of the cooling system for a limited time).

- Component behavior is only visible at the urban level through its public subset of y. Internal behavior is not published and not within the scope of the UP schema.

It should be noted that this choice of order-reduced component models could easily be changed in a future extension, for instance, because a higher resolution is desired. The current modules can be swapped with a higher-order model inside the software architecture, as all local simulations are delegated to component models that "live underwater." A prerequisite is that the new model publishes the same y and reacts to the same actuator signals.

The coverage of component types in the urban energy network has been introduced in CHAPTER 2. The following sections describe the associated component energy models. Every instance of a network component is represented by a simulation module that runs concurrently with all other modules in the platform.

3.2 Consumer

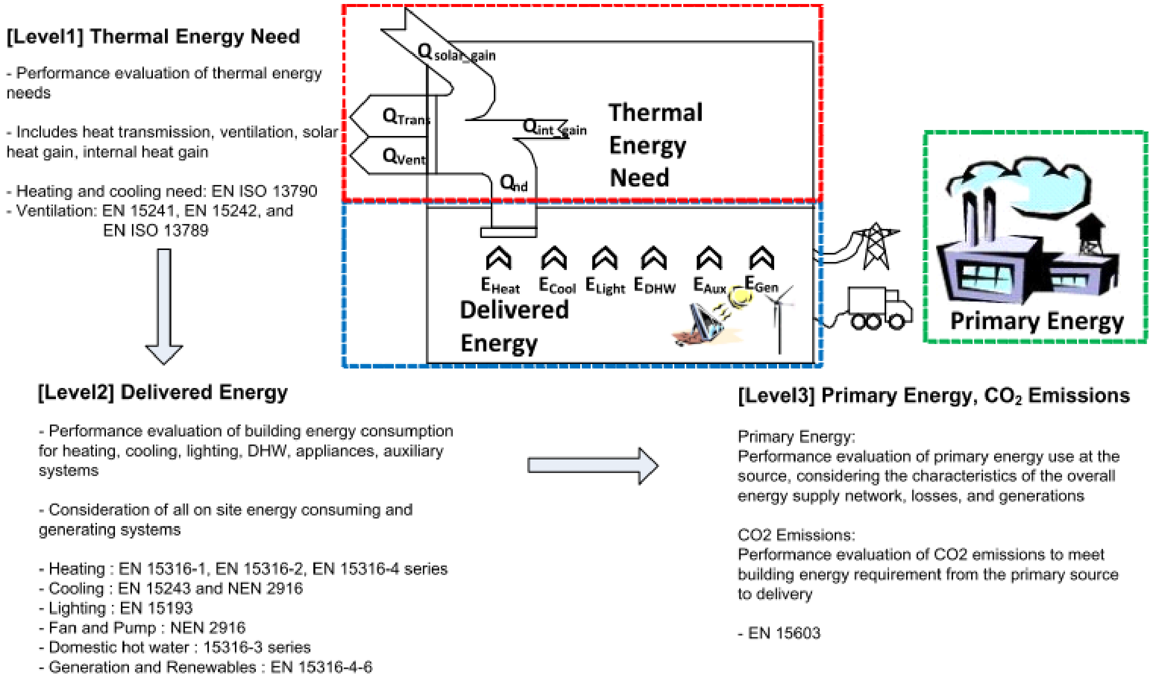
Table 15 tabulates the `Consumer` child elements' input/output simulation parameters and calculation equations. The input parameters are divided into two parts, depending on whether actuators control the parameter or not.

Table 13. `Consumer` Simulation Model Input/Output List

`Consumer` Element	Input Parameter		Output Parameter	Calculation Equation
	Controllable	Not Controllable	Sensor	
`Building`	Table 9	Building attributes*	Table 14	-
`Electric Vehicle Charging Station`	-	f	$P_{Charger}, N$	(1)

*Building attributes – they are private and include all building input parameters that do not belong to Table 9, such as building geometry, envelope (area, material), zone schedule, HVAC/DHW system efficiency, etc.

For the `Building` element, we utilize an adapted hourly normative calculation method, referred to as the energy performance calculator (EPC), based on the ISO 13790:2008 standard (ISO 2008). It calculates the hourly thermal heating/cooling energy need. In addition to the thermal energy needs, seven delivered energy uses, heating, cooling, fan, pump, appliance, lighting, DHW energy uses, are calculated based on a set of standards, as introduced in Figure 25.



[Level1] Thermal Energy Need

- Performance evaluation of thermal energy needs

- Includes heat transmission, ventilation, solar heat gain, internal heat gain

- Heating and cooling need: EN ISO 13790
- Ventilation: EN 15241, EN 15242, and EN ISO 13789

[Level2] Delivered Energy

- Performance evaluation of building energy consumption for heating, cooling, lighting, DHW, appliances, auxiliary systems

- Consideration of all on site energy consuming and generating systems

- Heating : EN 15316-1, EN 15316-2, EN 15316-4 series
- Cooling : EN 15243 and NEN 2916
- Lighting : EN 15193
- Fan and Pump : NEN 2916
- Domestic hot water : 15316-3 series
- Generation and Renewables : EN 15316-4-6

[Level3] Primary Energy, CO$_2$ Emissions

Primary Energy:
Performance evaluation of primary energy use at the source, considering the characteristics of the overall energy supply network, losses, and generations

CO2 Emissions:
Performance evaluation of CO2 emissions to meet building energy requirement from the primary source to delivery

- EN 15603

Figure 25. Building Energy Performance Analysis Process (Lee 2012)

As Figure 25 depicts, the calculation procedures comprise thermal energy needs, delivered energy, and primary/CO$_2$ emissions calculations. For thermal heating/cooling need calculation (level 1), the EPC engine calculates the thermal heating/cooling need based on the simplified heat transfer between the internal and external environment using an equivalent resistance-capacitance model which has five resistances ($\mathrm{H_{ve}}$, $H_{tr,w}$, $H_{tr,op}$, $H_{tr,is}$), five temperature nodes (θ_{sup}, θ_e, θ_{air}, θ_s, θ_m) and one capacitance (C_m). Heat transfer by ventilation ($\mathrm{H_{ve}}$) is directly connected to the air temperature node (θ_{air}) and the supply temperature (θ_{sup}). Heat transfer by transmission is split into the window ($H_{tr,w}$) and opaque ($H_{tr,op}$) parts. The opaque part only contains the thermal mass (C_m) and is split into two parts, $H_{tr,em}$ and $H_{tr,ms}$. The internal and solar heat gains are distributed over the air node (θ_{air}), central node (θ_s), and the node representing the mass of the building zone (θ_m). The heating and cooling need are calculated the need for heating and cooling power that needs to be supplied to or extracted from the air node (θ_{air}) to maintain a certain minimum or maximum setpoint temperature at each hour. For the delivered energy calculation (level 2), the required energy for heating, cooling, fan, pump, lighting, appliance, and DHW is calculated based on the designed system. For the HVAC system, heating and cooling energy losses via water or air delivery are considered. For primary/CO_2 emissions calculation (level 3), the primary energy use and CO_2 emissions are calculated based on the calculated delivered energy by multiplying the primary energy factor ($f_{primary}$) and CO_2 emission factor (f_{CO2}).

Table 14 shows the list of simulation outcomes from EPC and their corresponding `Building` sensors.

Table 14. List of EPC Simulated Outcomes (y)

EPC Simulated Outcomes	`Building` Sensor in the UP Schema
Total Number of Occupants	`Occupancy`
Heating Need	`ThermalHeatingNeed`
Cooling Need	`ThermalCoolingNeed`
Delivered Energy Use	`DeliveredEnergyUse`
Heating Energy Use	`HeatingDeliveredEnergyUse`
Cooling Energy Use	`CoolingDeliveredEnergyUse`
Lighting Energy Use	`LightingDeliveredEnergyUse`
Fan Energy Use	`FanDeliveredEnergyUse`
Pump Energy Use	`PumpDeliveredEnergyUse`

Appliance Energy Use	`ApplianceDeliveredEnergyUse`
DHW Energy Use	`DHWDeliveredEnergyUse`
Electricity Use	`ElectricityUse`
Natural Gas Use	`NaturalGasUse`

For the `ElectricVehicleChargingStation` element, we model an electric vehicle charging station (EVCS) instead of modeling a fleet of individual electric vehicles (EV). More specifically, we estimate the total power usage by electric vehicle chargers at each simulation time step. The total charging power usage at a given time is generated by an external traffic and usage model, which is supposed to deliver the charging needs of a given charging station. Equation (1) calculates the EVCS energy consumption as a function of the power of EV chargers $P_{charger}$, total number of EV chargers at the charging station *N*, the overall-schedule fraction of EV chargers *f*, and simulation time step Δt.

$$E_{EVCS} = P_{charger} \cdot N \cdot f \cdot \Delta t \qquad \textbf{(1)}$$

It should be noted that $P_{charger}$ is not a deterministic function as it will only be known approximately with significant variations. It then becomes necessary that the urban simulation is performed for many different samples drawn from the population of $P_{charger}$ functions that represent the variability. This can be easily accomplished with the standard Monte Carlo approach. In the next stage of this thesis research, an application of an uncertainty analysis will be shown.

3.3 Producer

Table 15 tabulates the `Producer` child elements' input/output simulation parameters and calculation equations. The input parameters are divided into two parts, depending on whether actuators control the parameter or not.

Table 15. Producer Simulation Model Input/Output List

`Producer` Element	Input Parameter		Output Parameter	Calculation Equation
	Controllable	Not Controllable	Sensor	
`PV`	$E_{sol,pv}$	K_{pk}, A_{PV}, f_{perf}	$E_{el;pv}$	(2)
`Wind Turbine`	-	l, η	$E_{el;wt}$	(3)
`SolarHot Water`	$E_{sol,shw}$	$A_{shw}, \eta_{collector}, \eta_{DHW,gen}$	$E_{DHW;SHW}$	(4)

For the `PV` element calculation, the amount of solar radiation cast on the PV module $E_{sol,pv}$, is controlled by PV actuators, `ModuleTiltAngle` and `ModuleOrientation` actuators. Other input parameters such as peak power K_{pk}, total PV module area A and performance factor f_{perf} are not controlled by the `PV` actuators. The PV's output, electricity generation $E_{el,PV}$, is the `PV` element's sensor. The PV calculation equation is derived from Equation (2) (ISO 2007). The reference solar irradiance, I_{ref}, is equal to 1 kW/m^2 to correct the output unit.

$$E_{el;pv} = \left(E_{sol,pv} \cdot K_{pk} \cdot A_{PV} \cdot f_{perf}\right) / I_{ref} \tag{2}$$

For the `WindTurbine` calculation, it does not have any input parameter controlled by the actuator and requires wind turbine diameter and efficiency. The electricity generation output $E_{el;wt}$, is the `WindTurbine` element's sensor, and the wind turbine's electricity output calculation is referred from Equation (3) (Tong 2010). Other than the `WindTurbine` element's input parameters, the gas constant R is assumed to equal 287 (J/(kg·K)). Furthermore, atmospheric pressure p, local temperature T and wind speed v are provided from weather data.

$$E_{el;wt} = 0.5 \cdot \frac{p}{R \cdot T} \cdot \eta \cdot 0.25 \cdot \pi \cdot l^2 \cdot v^3 \tag{3}$$

For the `SolarHotWater` calculation, the amount of solar radiation cast on the collector $E_{sol,shw}$, is affected by `SolarHotWater` actuators, `CollectorTiltAngle`, and `CollectorOrientation`. The domestic hot water energy generation is also a function of the area of solar collector A_{shw}, the efficiency of solar collector $\eta_{collector}$, and efficiency of DHW heat generation $\eta_{DHW,gen}$. The `SolarHotWater` element's calculation equation is shown in Equation (4) (Netherlands Standardization Institute 1999).

$$E_{DHW;SHW} = E_{sol,shw} \cdot A_{shw} \cdot \eta_{collector} / \eta_{DHW,gen} \tag{4}$$

3.4 District System

Table 16 shows the `DistrictHeating`, `DistrictCooling`, and `Cogeneration`'s input/output simulation parameters.

Table 16. `DistrictSystem` Model Input/output List

`DistrictSystem` Element	Input Parameters	Output Parameter	Calculation Equation
`DistrictHeating`	$\sum Q_{heating\_need}, \eta_{gen}, \eta_{dist}$	E_{heat}	(5)
`DistrictCooling`	$\sum Q_{cooling\_need}, \eta_{gen}, \eta_{dist}$	E_{cold}	(6)
`Cogeneration`	$\sum Q_{heating\_need}, \eta_{gen}, \eta_{dist}, \eta_{elec,gen}$	E_{heat}, E_{elec}	(7)

For the `DistrictHeating` and calculations, the generated heating delivered energy (E_{heat}) is calculated based on the summation of connected buildings' heating needs ($\sum Q_{heating\_need}$), heat generation efficiency (η_{gen}), and distribution heat loss efficiency (η_{dist}). The `DistrictHeating` equation is shown in Equation (5) (CEN 2007b).

$$E_{heat} = \sum Q_{heating\_need} / (\eta_{gen} \cdot \eta_{dist}) \tag{5}$$

For the `DistrictCooling` calculation, the generated cooling delivered energy (E_{cold}) is estimated based on the summation of connected buildings' cooling needs ($\sum Q_{cooling\_need}$), coolth generation efficiency, (η_{gen}), and distribution coolth loss efficiency (η_{dist}). The `DistrictHeating` equation is shown in Equation (6) (CEN 2007b).

$$E_{cool} = \sum Q_{cooling\_need} / (\eta_{gen} \cdot \eta_{dist}) \tag{6}$$

For the `Cogeneration` calculation, The heating delivered energy (E_{heat}) is same as the DistrictHeating element's equation, as shown in Equation (5). The `Cogeneration` element generates electricity from the wasted heat and the electricity generation (E_{elec}) is calculated by multiplying the annually-averaged electricity generation efficiency ($\eta_{elec,gen}$). Table 17 shows the efficiencies for different heat generation technologies (EPA-NR 2007).

$$\begin{gathered} E_{heat} = \sum Q_{heating\_need} / (\eta_{gen} \cdot \eta_{dist}) \\ E_{elec} = E_{heat} \cdot \eta_{elec,gen} \end{gathered} \tag{7}$$

Table 17. Indicative Efficiencies for Different Prime Mover Technologies for Building-integrated Cogeneration Installations (EPA-NR 2007)

Efficiency at nominal level	Internal Combustion Engine		Micro Turbine	Stirling Engine	Fuel Cell
	Gas	Diesel			
Electricity	0.21 - 0.38	0.30 - 0.40	0.13 – 0.32	0.10 – 0.25	0.25 – 0.50
Heat	0.45 - 0.61	0.50 – 0.60	0.52 – 0.66	0.61 – 0.95	0.35 – 0.70
Total	0.73 - 0.95	0.78 – 0.95	0.70 – 0.90	0.83 – 1.05	0.75 – 0.95

3.5 Storage

For the `ElectricityBattery` element, we use the electricity battery model expressed in terms of its state of charge (SOC) and use the linearized model (Vejdan and Grijalva 2018). The SOC of the electricity battery is determined by the previous time step's SOC SOC_{t-1}, charging power P_t^{chg}, and discharging power P_t^{dis}. The storage self-discharge percentage η_s, charging/discharging efficiencies η_{chg}, η_{dis} affect the current time step's SOC. The electricity battery equation is introduced in Equation (8).

Table 18. `ElectricityBattery` Model Input/output List

`Storage` Element	Input Parameters	Output Parameter	Calculation Equation
`Electricity Battery`	$\eta_s, \eta_{chg}, \eta_{dis}, SOC_{t-1}$	SOC_t	(8)

$$SOC_t = \eta_s \cdot SOC_{t-1} + (\eta_{chg} \cdot P_t^{chg} - P_t^{dis}/\eta_{dis}) \cdot \Delta t \tag{8}$$

3.6 Source

The `Source` elements such as `ElectricityGrid`, `NaturalGas`, etc. are intended to provide the required energy type to energy consumer elements to meet the electrical and thermal loads. The `Source` element is assumed to be an unlimited energy reservoir and can provide unlimited amounts of energy.

3.7 Power Flow

For now, power flows are determined in such a way that the `Consumer` elements' demand is satisfied with the `Producer` and `Source` elements at each simulation time step. When the `ElectricityBattery` elements are instantiated with renewable energy sources such as `PV` or `WindTurbine`, power flows are decided based on a power balanced approach. More explicitly, electricity battery charges from excess renewable energy sources and discharge during all periods, minimizing electricity usage from the electricity grid. Thus, the thermal or electrical power is delivered to the consumer elements in a cost-saving way, whose method aims to minimize the overall utility cost.

3.8 Interactions between Physical and Logical Layers

This sub-section shows how the UP schema facilitates the interactions between instantiated physical components, including the logical layer.

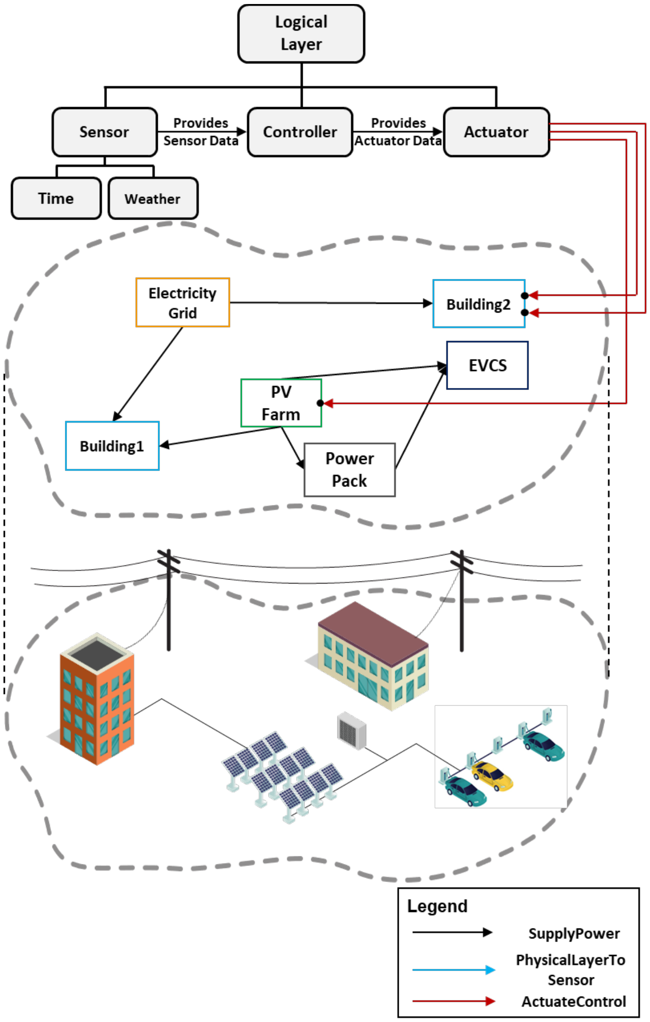
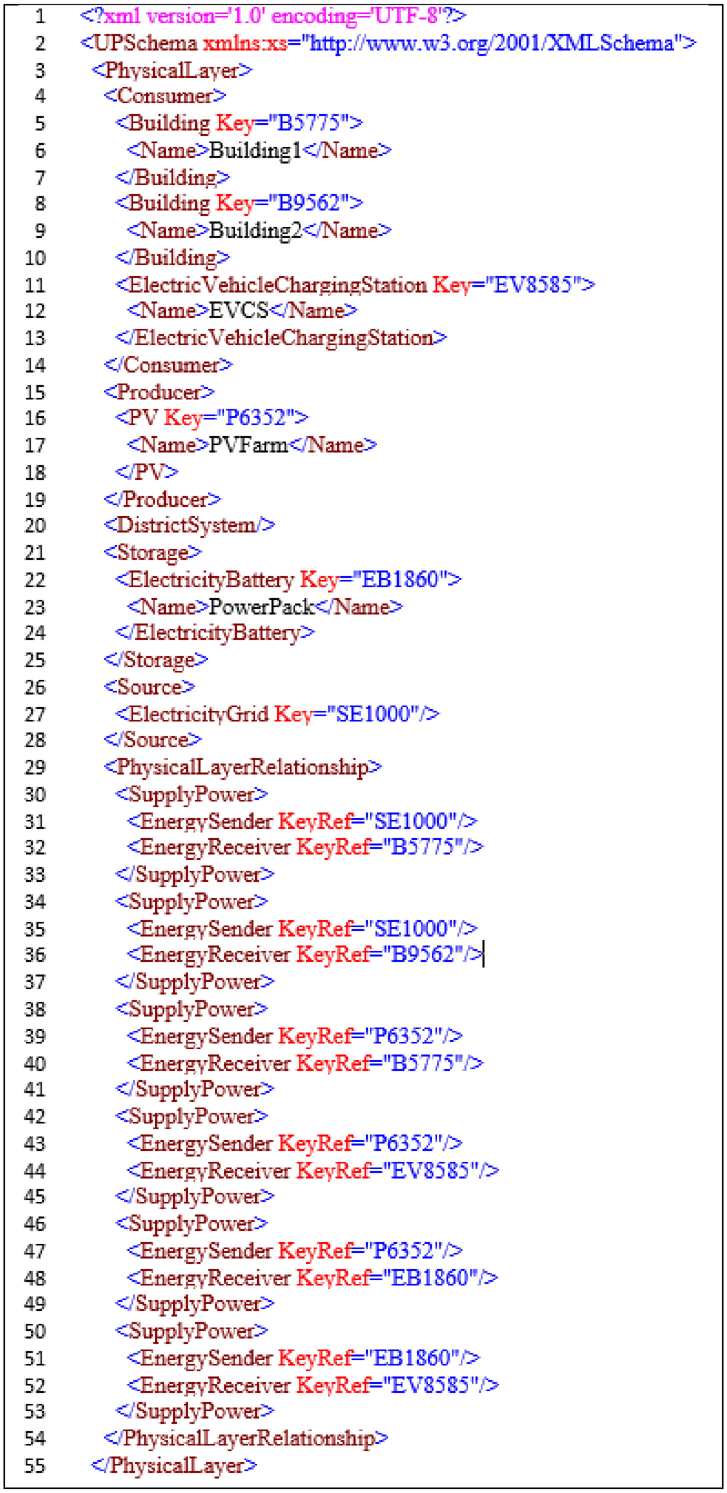


Figure 26. Schematic of Example Setting

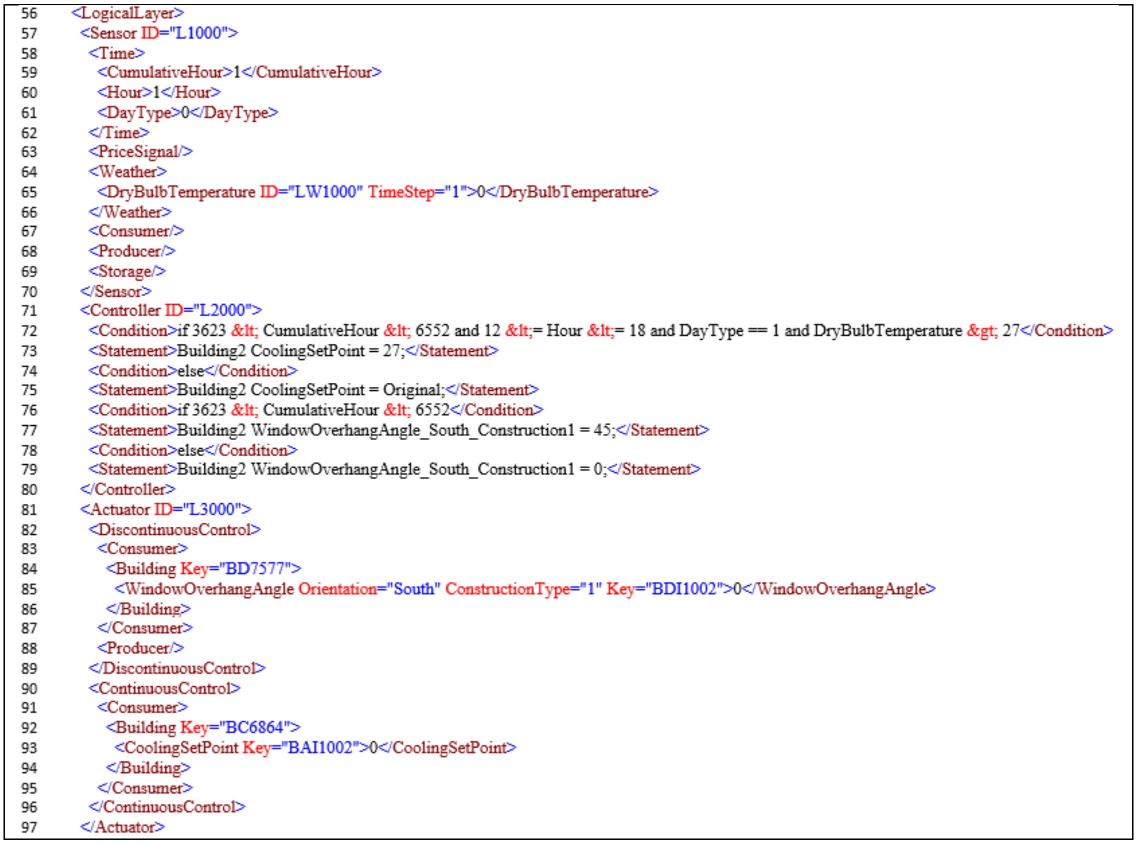
Figure 26 shows an example urban setting. The bottom diagram represents an example setting ("reality") that consists of two buildings (Building1, Building2), PV (PVFarm), electricity battery (PowerPack), EVCS (ChargingStation), and the electricity grid. The middle diagram shows the physical layer that conceptualizes the example setting, and the top diagram depicts the instantiated logical layer in the UP instance. It was assumed that the cooling setpoint and southern-orientation overhang actuators in the Building 2 node are controlled as a function of `Time` and `Weather` sensors. Figure 27, Figure 28, and Figure 29 show the actual UP instance of the example urban setting.

As soon as the physical layer is developed, each physical component is connected to its corresponding simulation model. The rectangle below each node represents the corresponding simulation model. For example, the “Building1” node instantiates the EPC simulation model to calculate its state (x)s and observations y=g(x) over time. Each component simulation model generates the observations that are read by specific sensors, where the UP instance specifies the associations between physical sensor and component observation. During execution, each simulation module reacts to certain changes in actuator values; the UP instance contains the associations between actuators (representing a physical device or abstract control signal) and parameters in the component model. The dots on the right side of a component box represent a discontinuous and continuous actuator port, respectively. If there are sensors that are exported to the global level, they are denoted by dots on the left side of the component boxes (not used in this example). In the example setting, the instantiated “Weather” and “Time” sensors’ data are used to determine the controller element's logic. The value given to an actuator port is the result of the actions in the “Controller.” In the example scenario, the “Actuator” controls “Building2” and “PVFarm” using the “ActuateControl” relationship.



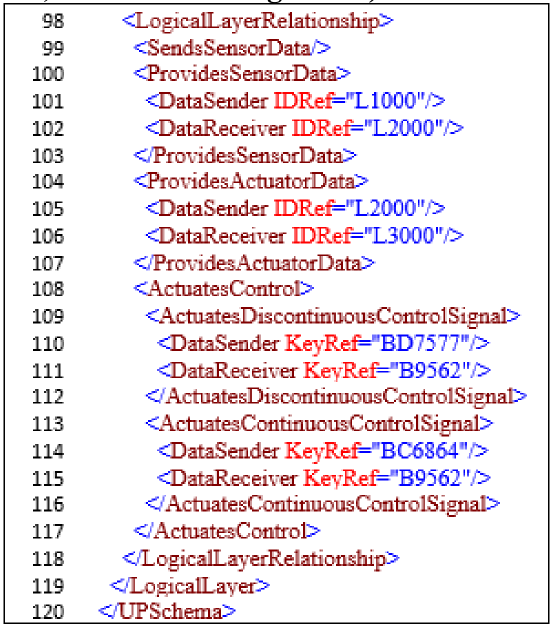
```xml
<?xml version='1.0' encoding='UTF-8'?>
<UPSchema xmlns:xs="http://www.w3.org/2001/XMLSchema">
 <PhysicalLayer>
  <Consumer>
   <Building Key="B5775">
    <Name>Building1</Name>
   </Building>
   <Building Key="B9562">
    <Name>Building2</Name>
   </Building>
   <ElectricVehicleChargingStation Key="EV8585">
    <Name>EVCS</Name>
   </ElectricVehicleChargingStation>
  </Consumer>
  <Producer>
   <PV Key="P6352">
    <Name>PVFarm</Name>
   </PV>
  </Producer>
  <DistrictSystem/>
  <Storage>
   <ElectricityBattery Key="EB1860">
    <Name>PowerPack</Name>
   </ElectricityBattery>
  </Storage>
  <Source>
   <ElectricityGrid Key="SE1000"/>
  </Source>
  <PhysicalLayerRelationship>
   <SupplyPower>
    <EnergySender KeyRef="SE1000"/>
    <EnergyReceiver KeyRef="B5775"/>
   </SupplyPower>
   <SupplyPower>
    <EnergySender KeyRef="SE1000"/>
    <EnergyReceiver KeyRef="B9562"/>
   </SupplyPower>
   <SupplyPower>
    <EnergySender KeyRef="P6352"/>
    <EnergyReceiver KeyRef="B5775"/>
   </SupplyPower>
   <SupplyPower>
    <EnergySender KeyRef="P6352"/>
    <EnergyReceiver KeyRef="EV8585"/>
   </SupplyPower>
   <SupplyPower>
    <EnergySender KeyRef="P6352"/>
    <EnergyReceiver KeyRef="EB1860"/>
   </SupplyPower>
   <SupplyPower>
    <EnergySender KeyRef="EB1860"/>
    <EnergyReceiver KeyRef="EV8585"/>
   </SupplyPower>
  </PhysicalLayerRelationship>
 </PhysicalLayer>
```

Figure 27. Example Urban Setting's UP instance (Physical Layer)



```
 <LogicalLayer>
  <Sensor ID="L1000">
   <Time>
    <CumulativeHour>1</CumulativeHour>
    <Hour>1</Hour>
    <DayType>0</DayType>
   </Time>
   <PriceSignal/>
   <Weather>
    <DryBulbTemperature ID="LW1000" TimeStep="1">0</DryBulbTemperature>
   </Weather>
   <Consumer/>
   <Producer/>
   <Storage/>
  </Sensor>
  <Controller ID="L2000">
   <Condition>if 3623 &lt; CumulativeHour &lt; 6552 and 12 &lt;= Hour &lt;= 18 and DayType == 1 and DryBulbTemperature &gt; 27</Condition>
   <Statement>Building2 CoolingSetPoint = 27;</Statement>
   <Condition>else</Condition>
   <Statement>Building2 CoolingSetPoint = Original;</Statement>
   <Condition>if 3623 &lt; CumulativeHour &lt; 6552</Condition>
   <Statement>Building2 WindowOverhangAngle_South_Construction1 = 45;</Statement>
   <Condition>else</Condition>
   <Statement>Building2 WindowOverhangAngle_South_Construction1 = 0;</Statement>
  </Controller>
  <Actuator ID="L3000">
   <DiscontinuousControl>
    <Consumer>
     <Building Key="BD7577">
      <WindowOverhangAngle Orientation="South" ConstructionType="1" Key="BDI1002">0</WindowOverhangAngle>
     </Building>
    </Consumer>
    <Producer/>
   </DiscontinuousControl>
   <ContinuousControl>
    <Consumer>
     <Building Key="BC6864">
      <CoolingSetPoint Key="BAI1002">0</CoolingSetPoint>
     </Building>
    </Consumer>
   </ContinuousControl>
  </Actuator>
```

Figure 28. Example Urban Setting's UP instance (Logical Layer – Sensor, Controller, and Actuator, Cont'd from Figure 27)



```
  <LogicalLayerRelationship>
   <SendsSensorData/>
   <ProvidesSensorData>
    <DataSender IDRef="L1000"/>
    <DataReceiver IDRef="L2000"/>
   </ProvidesSensorData>
   <ProvidesActuatorData>
    <DataSender IDRef="L2000"/>
    <DataReceiver IDRef="L3000"/>
   </ProvidesActuatorData>
   <ActuatesControl>
    <ActuatesDiscontinuousControlSignal>
     <DataSender KeyRef="BD7577"/>
     <DataReceiver KeyRef="B9562"/>
    </ActuatesDiscontinuousControlSignal>
    <ActuatesContinuousControlSignal>
     <DataSender KeyRef="BC6864"/>
     <DataReceiver KeyRef="B9562"/>
    </ActuatesContinuousControlSignal>
   </ActuatesControl>
  </LogicalLayerRelationship>
 </LogicalLayer>
</UPSchema>
```

Figure 29. Example Urban Setting's UP instance (Logical Layer – Relationship, Cont'd from Figure 28)

CHAPTER 4. SYSTEM SIMULATION AND OPTIMIZATION

4.1 Introduction

$$\frac{dx}{dt} = A(p) \cdot x + B \cdot u \quad \textbf{(9)}$$

$$\mathrm{y} = \mathrm{G}(x,\mathrm{u}) \quad \textbf{(10)}$$

Where,

x: State variable;
y: Output variable;
u: Control variable;
A: System variable;
B: Input variable.

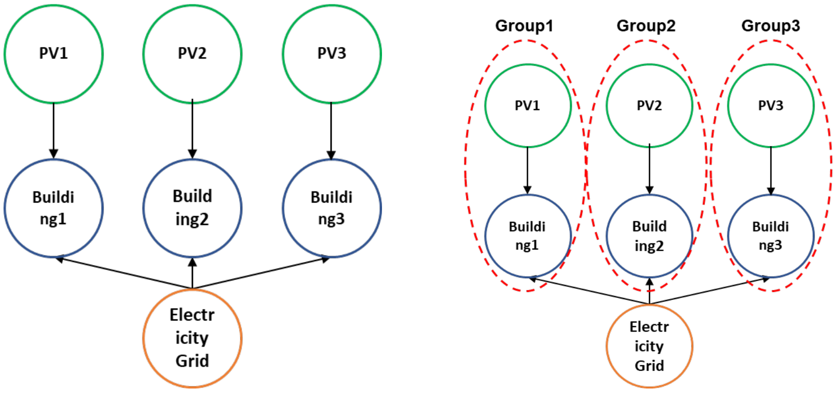
Equation (9) shows the state equation of the physical components, and the state vector x contains the state variables. Equation (10) shows the output equation that provides complete knowledge of all variables of the system. In an existing urban network, the control vector u can be controlled by either rule-based control (RBC) or optimal control. When optimal control is used, the optimization scope is typically limited to operational control (u). However, in the design stage of new urban networks or new extensions to existing urban networks, the optimization scope is a combination of design variables (p) and control variables (u). An example is the study of retrofit interventions in an existing set of buildings under control that achieves optimal operation for any chosen set of interventions. In that case, the choice of retrofits (p) and optimal control (u) are both considered part of the design space of the optimization. Some aspects of the different types of control and or optimization are discussed below.

RBC refers to a static control strategy based on “IF-THEN” commands. Rule-based control is straightforward compared to other control schemes, but it should be recognized that commercial building automation systems typically use RBC (Fontenot and Donsag 2019). We utilize RBC with pre-selected sensors and actuators in each physical component, as introduced in CHAPTER 3. As explained before, some RBCs are internal to the component simulation and are not externally exposed and hence are not specified in the UP instance. A good example of this is how a cooling system operates to satisfy a given temperature setpoint in a zone; the operation and its implicit control execution are embedded in the software. The setpoint itself is, however, a target that is managed at the network level and is in fact treated as an actuator whose value is controlled at the global level. This can be done in a static RBC manner where the setpoint value is solely determined by the current states of components in the network. Where RBC is common for the internal control of devices inside a component, it is not very common at the global level.

Optimal control refers to determining control variable u and state vector x for a dynamic system over a period of time to minimize an optimality criterion (expressed as an objective function). For instance, in a demand response scenario at the urban level, the setpoint in different buildings maybe determined by an optimal control that aims to optimize peak power reduction over an ensemble of buildings that each is connected to DER nodes. The optimal control would then react to the dynamic changes in power consumption, weather, and usage patterns of the different buildings. It would do so, for example, by trying to obtain the minimum value of a cost measure. One such cost measure would be utility cost-saving minus productivity loss cost. There are two types of optimal controls, parameterized optimal control and optimal control. The parameterized optimal control selects the optimal parameter values among a set of predefined discrete sets. For instance, PV module tile angle is selected from the set {0°, 30°, 45°, 60°, 90°} that minimizes the cost function. In contrast, the optimal control finds the optimal value of the control parameters using mixed-integer non-linear programming (MINLP).

Design optimization (DO) refers to an optimal mix of a user-provided set of candidate design variables (*p*). In buildings, the latter formulation is rarely practical as the candidates are technologies rather than parameters. This usually leads to the consideration of a discrete set of technologies where each technology links to a set of component model parameters. The design variables, in that case, are a set of discrete technologies with a choice of achievement levels. They are associated with actual products in the market, with the known cost. DO conducts the optimization by finding the optimal mix of technologies and achievement levels considering a specific criterion, such as minimum total cost. DO can be optionally included in conjunction with RBC and/or optimal control and can also be used independently.

In terms of executing the simulation, there are two simulation levels, i) global ii) local. In the global case, the entire instantiated physical components play their part in the same objective function. Therefore, the concurrent simulation of all nodes in the network is needed to find the optimal solution. In contrast, a local simulation is appropriate when a subset of nodes is “islanded” and has a local optimal solution. Figure 30 shows two identical situations that can be treated as three islanded subsets or one joint global system. The distinction can be used to model different variants of global versus distributed local control.



(a) (b)

Figure 30. (a) Example Urban Scenario (b) "Islanded" Physical Components for the Local Scope

To conduct the optimization, a random-based solution space searching metaheuristic algorithm, i.e., a genetic algorithm (GA), is used in the software platform used in our research. In the next stage of the research, the scope will be enlarged in the sense that GA optimization will be combined with dynamic optimal control solutions. More specifically, physical components will be reformulated in a differential-algebraic equation form, allowing MINLP for optimal control optimization. Its objective function is for now set up to minimize the summation of utility cost and productivity loss cost (if it exists). More about productivity loss cost is explained in the following paragraphs. If design variables are included in the simulation, the design variable's upfront cost is also added to the objective function. The constraints are user-provided min/max of each optimal control variable value and its cost limit.

4.2 Application Setting: Joint Optimization of Design and Control

The application treated in this section is derived from a recent thesis in the High Performance Building lab at Geogia Tech (Zhang 2017). The author performs an optimization to inform a building operator how to find the most optimal building interventions to avoid demand charges. Our goal is to recast the optimization performed in that study for one building to multiple buildings with additional community-level interventions.

Table 19 shows the list of the building design parameters and control variables used in this application. It can be seen that in the current set-up, the `Building` element involves five design parameters and four control variables.

Table 19. List of Design Parameters and Control Variables of Network Nodes

Physical Component	Design variable (p)	Control variable (u)
`Building`	Wall Insulation Thickness [mm]	Lighting Dimmer [ea.]
	Infiltration Rate [$m^3/h/m^2$]	Temperature Control [°C]
	Roof Emissivity [-]	Voltage Throttling [-]
	Window SHGC [-] Natural Ventilation [-]	N/A
`PV`	Module Area [m^2]	Module Tilt Angle [°]
		Module Orientation [-]
`SolarHotWater`	Collector Area [m^2]	Collector Tilt Angle [°]
		Collector Orientation [-]

`Electricity Battery`	Capacity [kWh]	N/A
`DistrictHeating`	Heating COP [-]	N/A
`DistrictCooling`	Cooling COP [-]	N/A
`Cogeneration`	Heating COP [-]	N/A

Wall insulation thickness is related to heat transmission losses and gains. Increasing wall insulation thickness by adding exterior insulation increases the opaque wall system's R-value and decreases both heating and cooling loads by reducing the heat flow through conduction to and from the interior space. For a typical batt insulation layer, the cost includes labor fees for fitting and securing batt insulation between open wall joists, preparation, cleanup, and materials fees. The total cost of installing wall insulation ranges from \$10 to \$17 per 1 m^2 (RS Means 2017). Unwanted infiltration can be reduced by caulking, mostly along line joints. Infiltration rate indicates the flow of outdoor air into a building through leaks, cracks, and other unintentional openings in the envelope. The range of the infiltration rate is from 0.2 to 0.8 $m^3/h/m^2$ (CEN 2007a). The total cost of caulking includes material fees, window trim, door trim, area preparation, and protection. The total cost of caulking ranges from \$4 to \$10 per 1 m (RS Means 2017).

Roof emissivity refers to the ability to re-radiate the absorbed heat back to the sky. The roof emissivity is measured on a scale ranging from 0 to 1. Most roofing materials have emittance values above 0.85 except for those with metallic surface or those treated with special coatings, typically white paint with some additional added compound. The cost to paint a dark roof white includes the material fees, labor fees of removing previous paint, brushing paint, and adding the waterproof layer. The total cost of painting a roof white ranges from \$10 to \$22 per 1 m^2 (RS Means 2017).

Window SHGC refers to the fraction of solar radiation admitted through a window, door, or skylight – either transmitted directly and/or absorbed and subsequently released as heat inside a home. SHGC is expressed as a number between 0 and 1. As the SHGC value is closer to 1, more solar heat is transmitted through a window. The cost to replace existing windows includes labor fees, removing old windows, installing new windows, and windows' costs. The total cost of replacing a window ranges from \$400 to \$650 (RS Means 2017).

Natural ventilation refers to the flow of air through open windows primarily driven by natural induced pressure differences. The natural ventilation design variable operates with a binary signal parameter. Whenever there is natural ventilation potential and a cooling need, the windows are opened to let in enough air such the cooling need is met by natural ventilation. The equipment to implement natural ventilation includes linear actuators, a controller, and labor fees. The cost ranges from \$145 to \$160 per window (Soon Industrial Co. Ltd. 2020) (RS Means 2020).

In this application, the focus is on load reduction (peak shaving) in the afternoon, which we assume is triggered by a demand response (DR) signal. For this reason, three control variables of the `Building` element are considered to operate from 12 p.m. to 4 p.m. in the summer months, i.e.:

- Lighting dimmer controls lights in certain areas of the building. The lighting dimmer cost includes labor and material fees, and the cost ranges from \$200 to \$300.
- Temperature control refers to a thermostat setting that can be adjusted upward in the afternoon between 12 and 4 p.m. This is treated as a control design variable, i.e., the delta T by which the setpoint is increased for the full four hours. There is no dynamic control to adjust the setpoint in the meantime. Note that private RBC inside a building object regulates the HVAC system such that the temperature floats freely up to the setpoint (cooling energy is zero), where it is then kept until 4p.m., whereas after 4 p.m., the HVAC system cools down the building to a preset setpoint, staying within its maximum capacity. There is no other cost involved with the temperature control cost than the potential cost of productivity loss of the occupants due to some form of discomfort when the temperature deviates from the optimal comfort temperature.
- Voltage throttling refers to the coincident peak demand and energy usage over time, regulating the voltage output of high power-consuming equipment, i.e., chillers. The voltage throttling is a binary parameter, and the building manager can determine whether or not the voltage throttling is considered an allowed intervention. The voltage throttling is realized by implementing a 20% capacity reduction on the chiller from 12 to 4 p.m. The cost of the voltage throttling is again calculated based on the potential productivity loss of occupants due to the temperature increase from 12 to 4

p.m. on the hottest days of the year when the power use of the HVAC system is larger than 0.8 of capacity.

To calculate productivity loss cost, we employ (Seppänen, Fisk, and Faulkner 2003)'s study, which claims the upper boundary of the thermal neutrality is 25 °C. Above this neutrality temperature, the productivity loss cost is calculated using Equation (11) at every hour. The productivity loss rate (L_{rate}) and average occupants' salary (S_{occ}) are input parameters reflecting each building's characteristics. The total number of occupants (N_{occ}) is considered based on the occupant density and occupancy schedules.

$$\begin{gathered} if\ T_{thermal\ zone} > 25\,°C \\ productivity\ loss\ cost\ (\$) = (T_{thermal\ zone} - 25) \cdot L_{rate} \cdot S_{occ} \cdot N_{occ} \end{gathered} \tag{11}$$

Where,

L_{rate}: Occupants' average productivity loss rate, [-];
S_{occ}: Occupants' hourly average salary, [$];
N_{occ}: Total number of occupants at time step t, [ea.];

For the `PV` and `SolarHotWater` elements, the module and collector areas refer to the size of the module and collector area. The installation costs for PV modules are \$3.1-4.5/W for residential systems and \$2.4-4.0/W for small non-residential systems and \$1.8-3.3/W for large non-residential systems (Barbose and Naïm 2019). The installation costs of solar hot water collectors include solar collector panels, frame, copper absorber plate, and labor fees and range from \$380-\$480 per m^2 (RS Means 2020).

The `PV` and `SolarHotWater`'s design variables are solar trackers (tilt angle, orientation). They are automated systems that continuously adjust to maximize their exposure to solar radiation and thus maximize the power/solar hot water output. If only one design variable is selected, single-axis solar tracker is applied. Similarly, if both tilt and orientation angles are chosen as control variables, a double-axis solar tracker is applied. The optimal module and collector tilt angle are chosen among 0°, 30°, 45°, 60°, and 90° that produce the highest energy production at every time step. Furthermore, the optimal module and collector tilt angles are selected from east, southeast, south, southwest, and west orientations. The choice is determined based on the highest energy production. The cost of installing a single-axis solar tracker is approximately $500 per module. The cost of installing a dual-axis solar tracker is approximately $1,000 per module (Lane 2020).

The `ElectricityBattery` element has a capacity as a design parameter that refers to the maximum amount of energy that the battery can contain. The electricity battery installation cost includes battery cost, labor fees, electrical and structural balance of systems (Fu et al. 2018). The cost of a utility-scale 60-MW U.S. Li-ion standalone system ranges from $380/kWh (4-hour duration, 240 MWh) to $895/kWh (Fu et al. 2018).

CHAPTER 5. SOFTWARE DEVELOPMENT

5.1 Executable Software

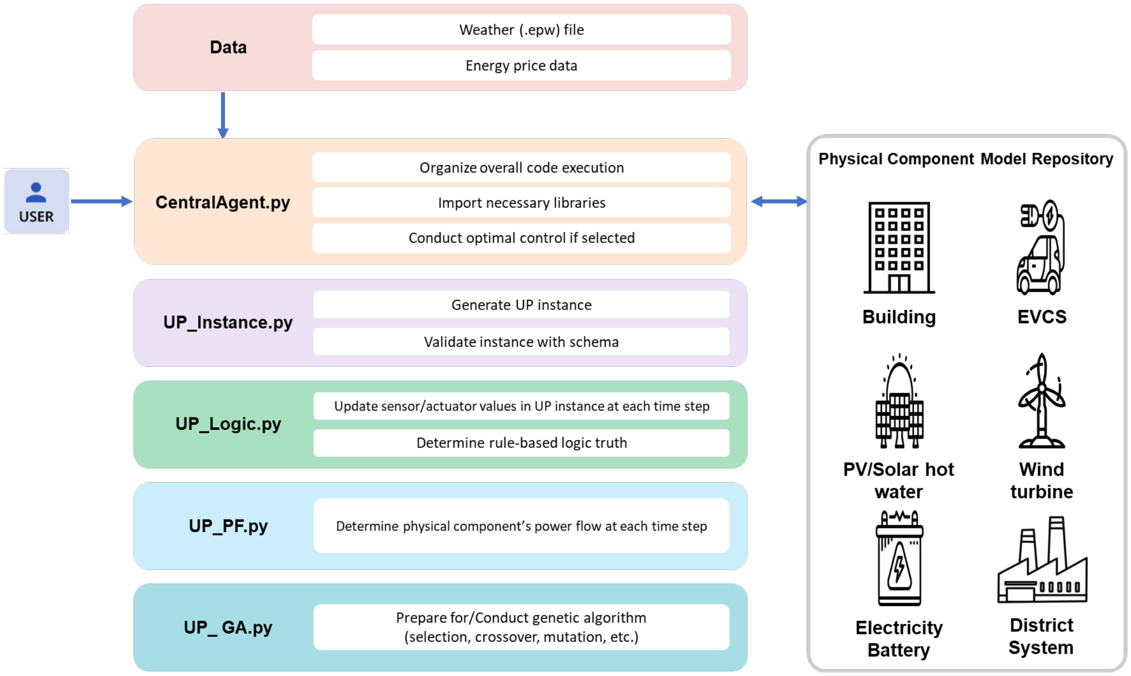
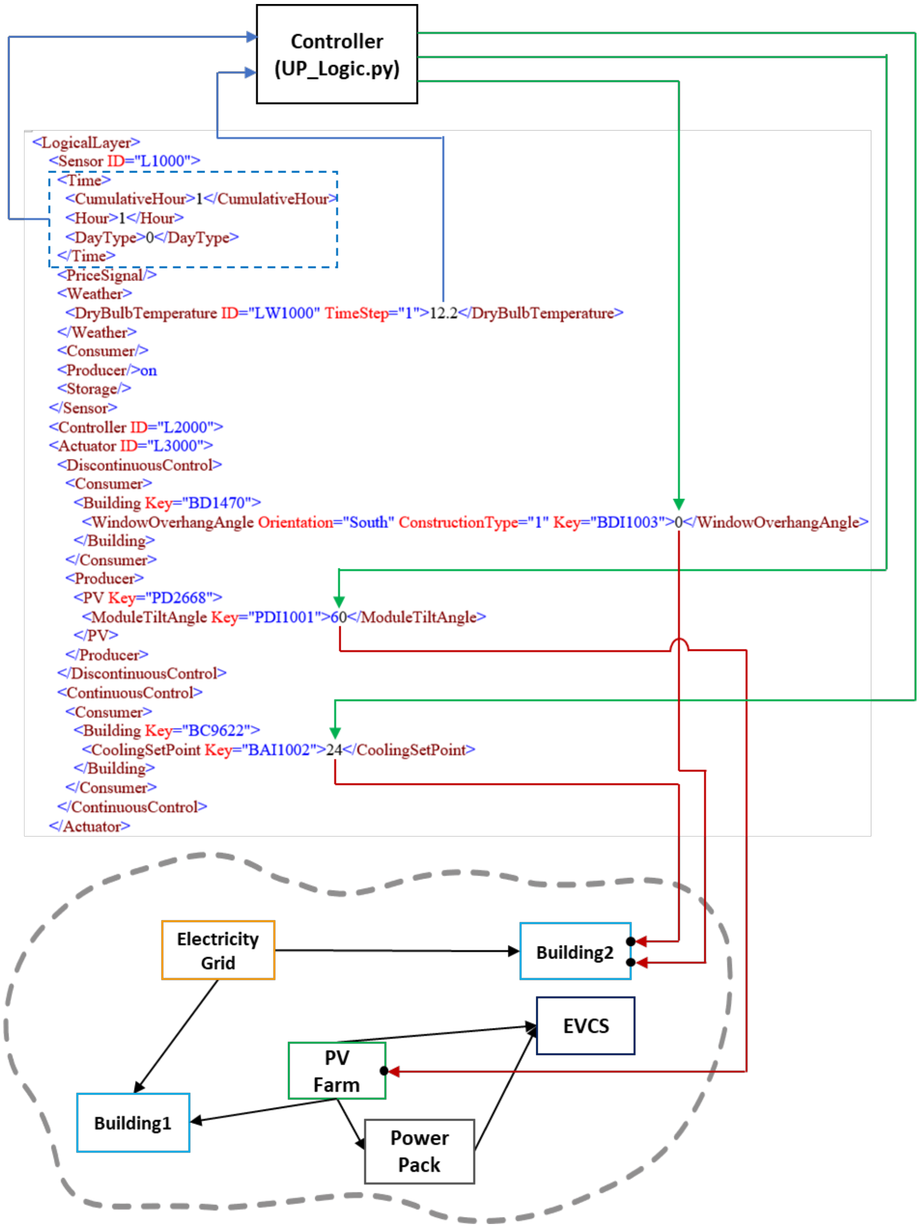


Figure 31. Software Architecture

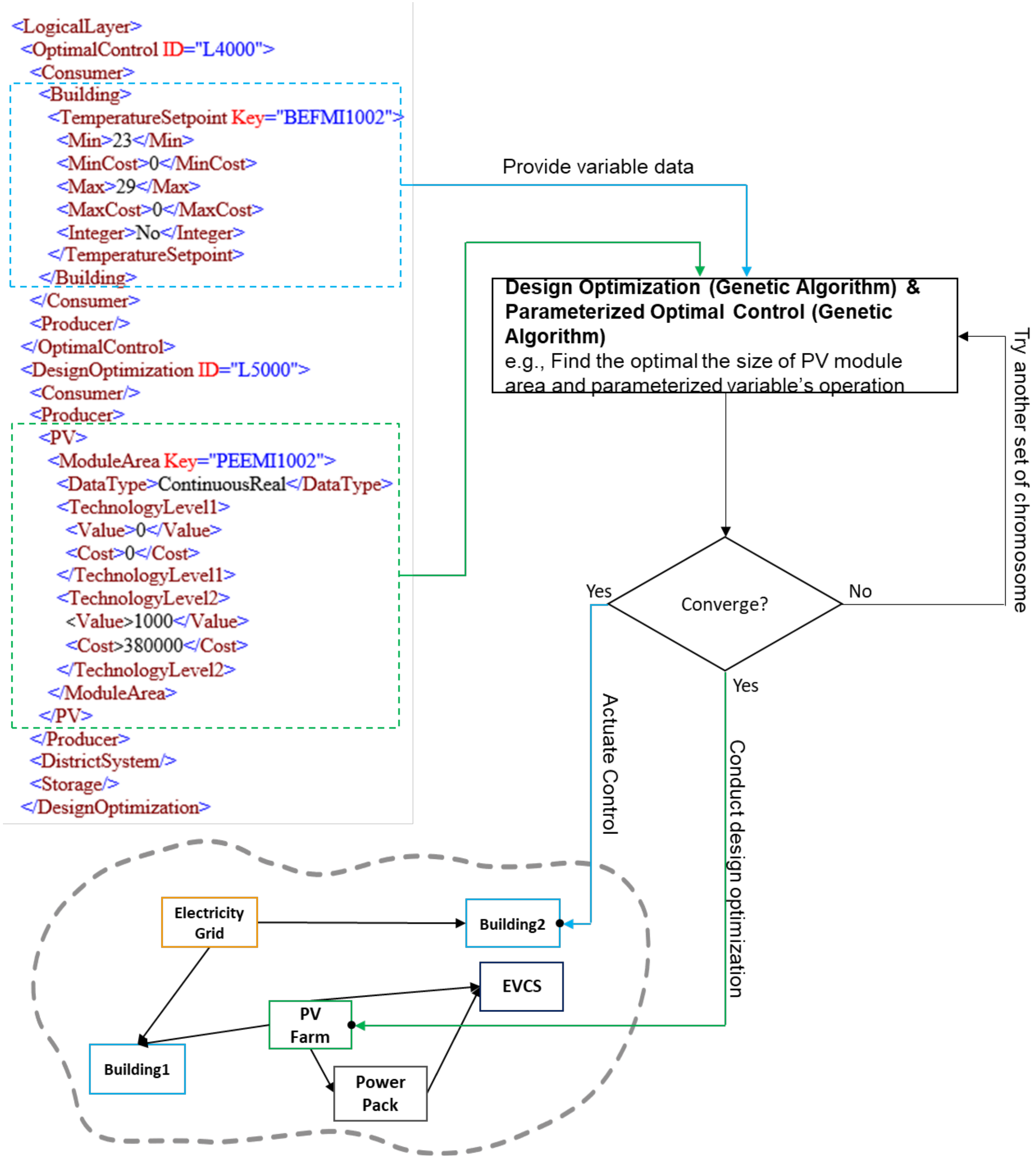
Figure 31 shows the software architecture, as coded in Python 3. The software consists of five Python scripts, “ControlAgent.py,” “UP_ Instance.py,” “UP_PF.py,” “UP_Logic.py,” and “UP_GA.py” and also includes the physical component simulation models.

The “ControlAgent.py” script is the script that the network model executes to organize the overall software operation. The “UP_Instance.py” script generates a UP instance based on the input data from the input Excel file and validates the UP instance against the UP schema. Accessing and manipulating UP instances are processed using the XML document object model (DOM) in Python. If the UP instance is determined as an invalid instance because of the modeler's mistakes, the software is terminated with a warning message. The "UP_Logic.py" script firstly tests whether there is any logic conflict. If any actuators are controlled by more than one input RBC logic, the logic is deemed to be potentially conflicted, and the software is terminated. More details about the logic conflict detection algorithm are explained in subsection 5.1.2. Furthermore, the way the RBC logic works is explained with one example urban setting in Figure 32. The lower part shows the same example urban setting used in Figure 26, and the upper part shows the logical part of the example urban setting's UP instance. At each simulation step, all the instantiated `Sensor` elements in the UP instance are updated by measuring sensor values from corresponding elements. The sensor values are sent to the controller part in the "UP_Logic.py." The code determines each condition's truth and determines which statement is executed. The actuator values in the executed statement are then updated, leading the associated physical component to apply the corresponding actuator values for the next simulation step. The "UP_PF.py" script determines power flows at each simulation time step, as explained in subsection 3.7. The "UP_GA.py" script is added to find the optimal parameters in optimal control and combined design-control optimization. The script implements the GA for the optimization and performs the usual GA steps such as crossover, mutation, etc. Some of which are modeler controlled and thus require specific inputs.



```xml
<LogicalLayer>
  <Sensor ID="L1000">
   <Time>
    <CumulativeHour>1</CumulativeHour>
    <Hour>1</Hour>
    <DayType>0</DayType>
   </Time>
   <PriceSignal/>
   <Weather>
    <DryBulbTemperature ID="LW1000" TimeStep="1">12.2</DryBulbTemperature>
   </Weather>
   <Consumer/>
   <Producer/>on
   <Storage/>
  </Sensor>
  <Controller ID="L2000">
  <Actuator ID="L3000">
   <DiscontinuousControl>
    <Consumer>
     <Building Key="BD1470">
      <WindowOverhangAngle Orientation="South" ConstructionType="1" Key="BDI1003">0</WindowOverhangAngle>
     </Building>
    </Consumer>
    <Producer>
     <PV Key="PD2668">
      <ModuleTiltAngle Key="PDI1001">60</ModuleTiltAngle>
     </PV>
    </Producer>
   </DiscontinuousControl>
   <ContinuousControl>
    <Consumer>
     <Building Key="BC9622">
      <CoolingSetPoint Key="BAI1002">24</CoolingSetPoint>
     </Building>
    </Consumer>
   </ContinuousControl>
  </Actuator>
```

Figure 32. Principle of RBC Operation



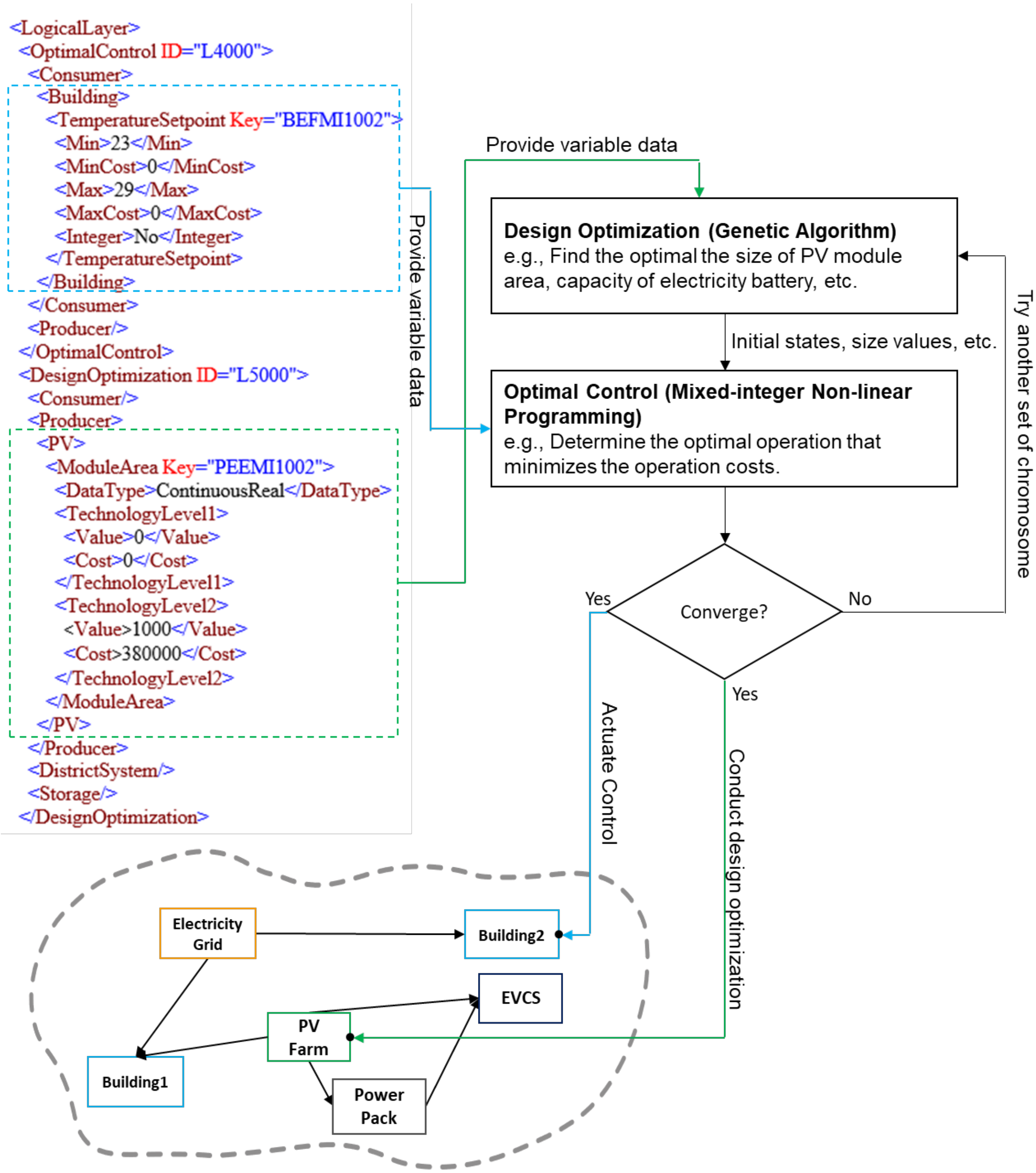
```
<LogicalLayer>
 <OptimalControl ID="L4000">
  <Consumer>
   <Building>
    <TemperatureSetpoint Key="BEFMI1002">
     <Min>23</Min>
     <MinCost>0</MinCost>
     <Max>29</Max>
     <MaxCost>0</MaxCost>
     <Integer>No</Integer>
    </TemperatureSetpoint>
   </Building>
  </Consumer>
  <Producer/>
 </OptimalControl>
 <DesignOptimization ID="L5000">
  <Consumer/>
  <Producer>
   <PV>
    <ModuleArea Key="PEEMI1002">
     <DataType>ContinuousReal</DataType>
     <TechnologyLevel1>
      <Value>0</Value>
      <Cost>0</Cost>
     </TechnologyLevel1>
     <TechnologyLevel2>
      <Value>1000</Value>
      <Cost>380000</Cost>
     </TechnologyLevel2>
    </ModuleArea>
   </PV>
  </Producer>
  <DistrictSystem/>
  <Storage/>
 </DesignOptimization>
```

Figure 33. Principle of Parameterized Optimal Control and Design Optimization Operation

In addition, Figure 33 illustrates how the joint parameterized optimal control and design optimization are conducted using the UP schema. The instantiated instance includes the `DesignOptimization` and `OptimalControl` elements. These elements in the UP instance have the necessary information for the optimization, such as min/max interval, min/max cost interval, etc. In the case of joint design optimization and parameterized optimization, only the genetic algorithm is utilized to find the optimal design and operations. More specifically, when the genetic algorithm is executed, the chromosomes involve both variables and control variables. This is possible because the optimal values of the parameterized optimal control variables are pre-determined, as introduced in CHAPTER 4. For instance, if the PV module tilt angle is included in parameterized optimal control, the gene value of the PV module tilt angle can only have the set {0°, 30°, 45°, 60°, 90°}, which are pre-determined values. The parameterized optimal control variable is reflected to the physical components at every time step.

Furthermore, Figure 34 illustrates the principle of joint optimal control and design optimization. The only difference between the parameterized optimal control + design optimization in Figure 33 and optimal control + design optimization in Figure 34 cases is the optimal control utilizes the MINLP. When the bi-level optimization (genetic algorithm and MINLP) is conducted, the genetic algorithm provides necessary data for MINLP such as initial states, size variables, etc. The optimal control is conducted at every 24-hour time horizon to prevent from losing urban energy dynamics.

If the genetic algorithm for the design optimization is converged, the optimization results are applied to the physical layer components using the `ConductsDesignOptimization` relationship. If not, another set of chromosomes is generated and this process is repeated until the algorithm is converged.



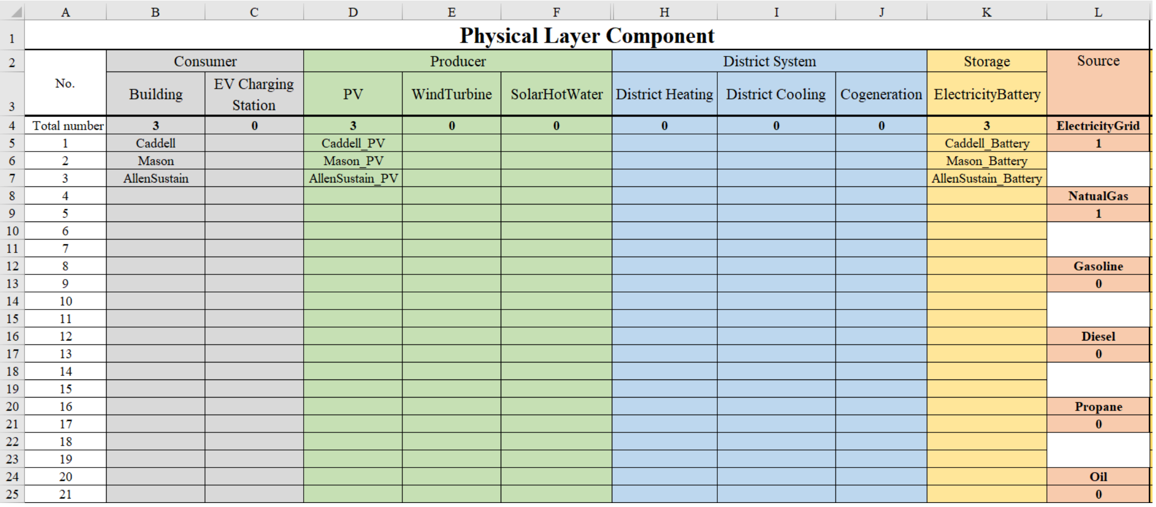
```
<LogicalLayer>
 <OptimalControl ID="L4000">
  <Consumer>
   <Building>
    <TemperatureSetpoint Key="BEFMI1002">
     <Min>23</Min>
     <MinCost>0</MinCost>
     <Max>29</Max>
     <MaxCost>0</MaxCost>
     <Integer>No</Integer>
    </TemperatureSetpoint>
   </Building>
  </Consumer>
  <Producer/>
 </OptimalControl>
 <DesignOptimization ID="L5000">
  <Consumer/>
  <Producer>
   <PV>
    <ModuleArea Key="PEEMI1002">
     <DataType>ContinuousReal</DataType>
     <TechnologyLevel1>
      <Value>0</Value>
      <Cost>0</Cost>
     </TechnologyLevel1>
     <TechnologyLevel2>
      <Value>1000</Value>
      <Cost>380000</Cost>
     </TechnologyLevel2>
    </ModuleArea>
   </PV>
  </Producer>
  <DistrictSystem/>
  <Storage/>
 </DesignOptimization>
```

Figure 34. Principle of Optimal Control and Design Optimization Operation (Bi-level Optimization)

There is no provision to generate the urban network model through a GUI from pre-existing data (GIS, CAD, GRID, or others) in the current stage of the development. This thesis will, in fact, not deal with the front-end part of the development. Instead, a number of modules are provided that let the modeler define the urban network in terms of components assuming pre-made simulation modules are available, as discussed earlier. Relationships are to be added by the modeler as well as control logic structure and statements, either in the form of RBC or optimal control logic, albeit that the latter is not yet generically supported in the UP schema (this will be done in the next stage of the development).

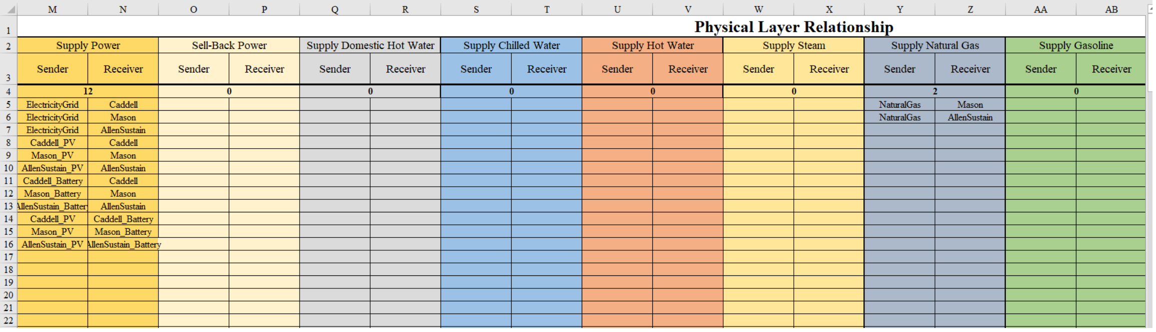
5.1.1 Input Module

In the developed software platform, the urban network model is populated by the modeler (thus instantiating a UP instance) through a set of tables supported by Excel with embedded rules to guarantee syntactical correctness. The series of sequential input worksheets are shown below.



	A	B	C	D	E	F	H	I	J	K	L
1	**Physical Layer Component**										
2	No.	Consumer		Producer			District System			Storage	Source
3		Building	EV Charging Station	PV	WindTurbine	SolarHotWater	District Heating	District Cooling	Cogeneration	ElectricityBattery	
4	Total number	**3**	**0**	**3**	**0**	**0**	**0**	**0**	**0**	**3**	**ElectricityGrid**
5	1	Caddell		Caddell_PV						Caddell_Battery	**1**
6	2	Mason		Mason_PV						Mason_Battery	
7	3	AllenSustain		AllenSustain_PV						AllenSustain_Battery	
8	4										**NatualGas**
9	5										**1**
10	6										
11	7										
12	8										**Gasoline**
13	9										**0**
14	10										
15	11										
16	12										**Diesel**
17	13										**0**
18	14										
19	15										
20	16										**Propane**
21	17										**0**
22	18										
23	19										
24	20										**Oil**
25	21										**0**

Figure 35. Physical Layer Input Sheet – Component Definition

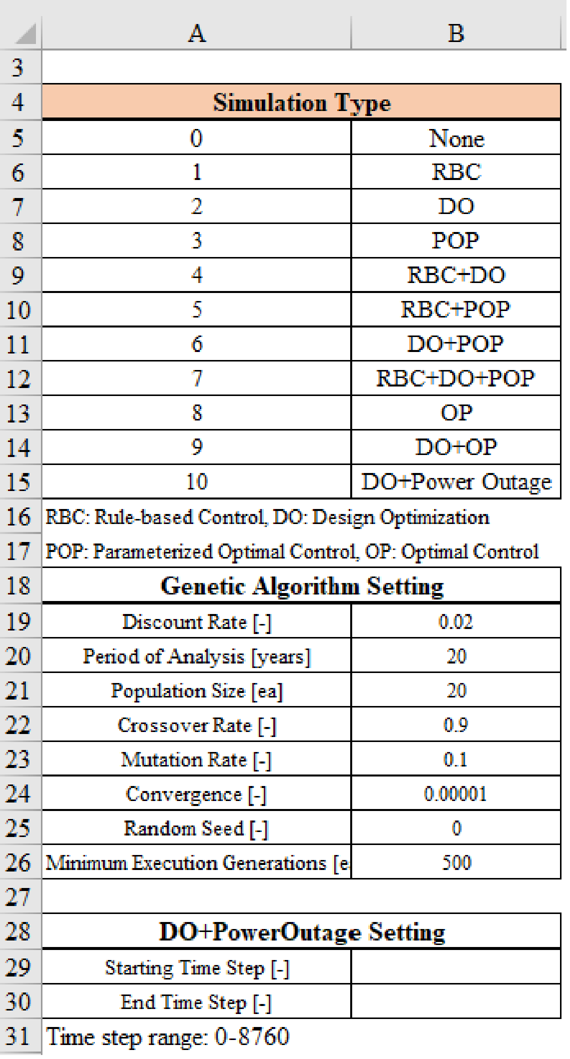


	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB
1	Physical Layer Relationship															
2	Supply Power		Sell-Back Power		Supply Domestic Hot Water		Supply Chilled Water		Supply Hot Water		Supply Steam		Supply Natural Gas		Supply Gasoline	
3	Sender	Receiver	Sender	Receiver	Sender	Receiver	Sender	Receiver	Sender	Receiver	Sender	Receiver	Sender	Receiver	Sender	Receiver
4	**12**		**0**		**0**		**0**		**0**		**0**		**2**		**0**	
5	ElectricityGrid	Caddell											NaturalGas	Mason		
6	ElectricityGrid	Mason											NaturalGas	AllenSustain		
7	ElectricityGrid	AllenSustain														
8	Caddell_PV	Caddell														
9	Mason_PV	Mason														
10	AllenSustain_PV	AllenSustain														
11	Caddell_Battery	Caddell														
12	Mason_Battery	Mason														
13	llenSustain_Batter	AllenSustain														
14	Caddell_PV	Caddell_Battery														
15	Mason_PV	Mason_Battery														
16	AllenSustain_PV	llenSustain_Battery														
17																
18																
19																
20																
21																
22																

Figure 36. Physical Layer Input Sheet – Component Relationship Definition

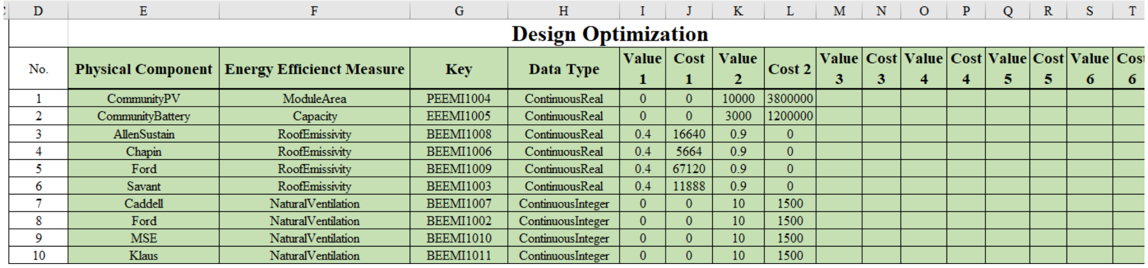
Modelers start by defining each physical layer component, as shown in Figure 35. Once the names of physical components of type `Consumer`, `Producer`, `DistrictSystem`, `Storage` components are provided, the corresponding simulation module for that specific component (as described in Chapter 3) is automatically found in the simulation module library and is imported into the executable code. This happens in two steps. Step 1 loads the generic simulation module (e.g., EPC) into the platform. Step 2 finds the component specification containing building and usage data for the buildings in the network (e.g., Input_Caddell). This building-specific data is pre-generated; it fully specifies the input for the run time module of a specific component (e.g., Caddell building). Note that it is assumed that this specification (file) is available for all nodes in the network. After the completion of the two steps, all component run-time modules are fully specified, whereas the run-time modules have "stubs" that can connect to instances of three object types.

Physical energy flows between two modules, choosing from the list of enumerated types of relationships. Modelers provide the energy relationships between the physical components, as shown in Figure 36. Each physical component should be placed in the available spot, as shown in Table 1. For example, `PV` cannot be placed in the "Receiver" cell in the "SupplyPower" column and should be placed in the "Sender" cell in the “SupplyPower” relationship if it exists. If modelers make mistakes, the generated UP instance is flagged as an invalid instance during the validation step, and the software is terminated with a warning message.



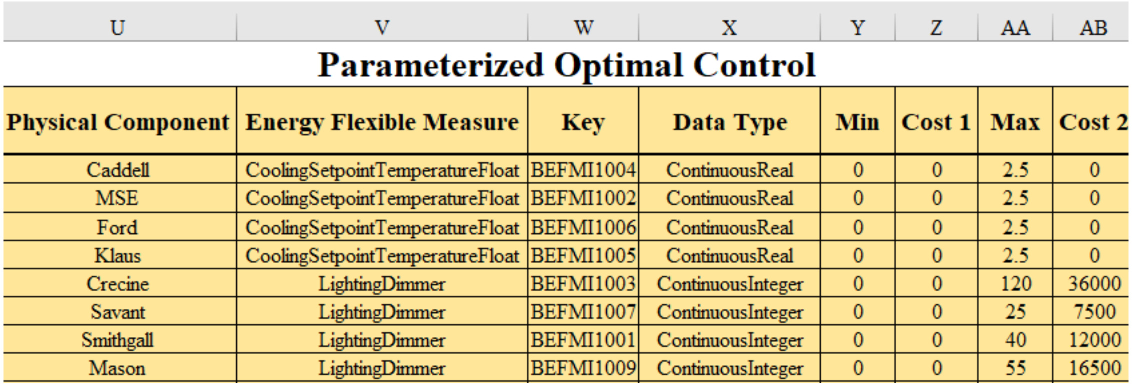
	A	B
3		
4	**Simulation Type**	
5	0	None
6	1	RBC
7	2	DO
8	3	POP
9	4	RBC+DO
10	5	RBC+POP
11	6	DO+POP
12	7	RBC+DO+POP
13	8	OP
14	9	DO+OP
15	10	DO+Power Outage
16	RBC: Rule-based Control, DO: Design Optimization	
17	POP: Parameterized Optimal Control, OP: Optimal Control	
18	**Genetic Algorithm Setting**	
19	Discount Rate [-]	0.02
20	Period of Analysis [years]	20
21	Population Size [ea]	20
22	Crossover Rate [-]	0.9
23	Mutation Rate [-]	0.1
24	Convergence [-]	0.00001
25	Random Seed [-]	0
26	Minimum Execution Generations [e	500
27		
28	**DO+PowerOutage Setting**	
29	Starting Time Step [-]	
30	End Time Step [-]	
31	Time step range: 0-8760	

Figure 37. Logical Layer Input Sheet – Simulation Type Selection



D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
	Design Optimization															
No.	**Physical Component**	**Energy Efficient Measure**	**Key**	**Data Type**	**Value 1**	**Cost 1**	**Value 2**	**Cost 2**	**Value 3**	**Cost 3**	**Value 4**	**Cost 4**	**Value 5**	**Cost 5**	**Value 6**	**Cos 6**
1	CommunityPV	ModuleArea	PEEMI1004	ContinuousReal	0	0	10000	3800000								
2	CommunityBattery	Capacity	EEEMI1005	ContinuousReal	0	0	3000	1200000								
3	AllenSustain	RoofEmissivity	BEEMI1008	ContinuousReal	0.4	16640	0.9	0								
4	Chapin	RoofEmissivity	BEEMI1006	ContinuousReal	0.4	5664	0.9	0								
5	Ford	RoofEmissivity	BEEMI1009	ContinuousReal	0.4	67120	0.9	0								
6	Savant	RoofEmissivity	BEEMI1003	ContinuousReal	0.4	11888	0.9	0								
7	Caddell	NaturalVentilation	BEEMI1007	ContinuousInteger	0	0	10	1500								
8	Ford	NaturalVentilation	BEEMI1002	ContinuousInteger	0	0	10	1500								
9	MSE	NaturalVentilation	BEEMI1010	ContinuousInteger	0	0	10	1500								
10	Klaus	NaturalVentilation	BEEMI1011	ContinuousInteger	0	0	10	1500								

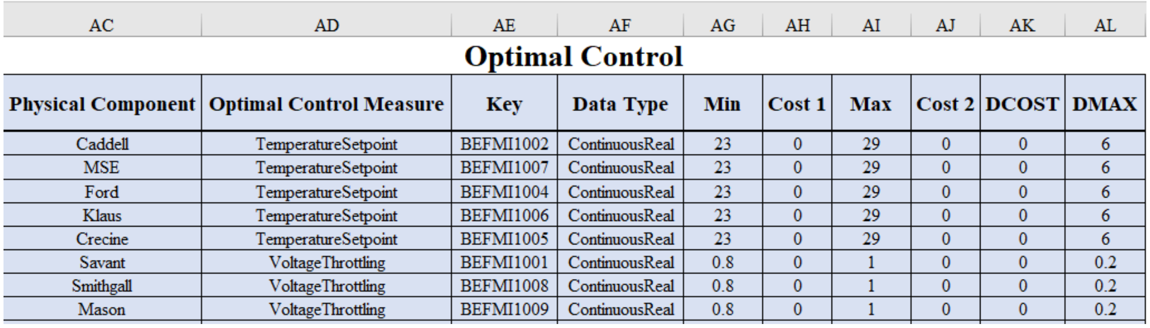
Figure 38. Logical Layer Input Sheet – Design Variable Input



Parameterized Optimal Control

U	V	W	X	Y	Z	AA	AB
Physical Component	**Energy Flexible Measure**	**Key**	**Data Type**	**Min**	**Cost 1**	**Max**	**Cost 2**
Caddell	CoolingSetpointTemperatureFloat	BEFMI1004	ContinuousReal	0	0	2.5	0
MSE	CoolingSetpointTemperatureFloat	BEFMI1002	ContinuousReal	0	0	2.5	0
Ford	CoolingSetpointTemperatureFloat	BEFMI1006	ContinuousReal	0	0	2.5	0
Klaus	CoolingSetpointTemperatureFloat	BEFMI1005	ContinuousReal	0	0	2.5	0
Crecine	LightingDimmer	BEFMI1003	ContinuousInteger	0	0	120	36000
Savant	LightingDimmer	BEFMI1007	ContinuousInteger	0	0	25	7500
Smithgall	LightingDimmer	BEFMI1001	ContinuousInteger	0	0	40	12000
Mason	LightingDimmer	BEFMI1009	ContinuousInteger	0	0	55	16500

Figure 39. Logical Layer Input Sheet – Parameterized Optimal Control Variable Input



Optimal Control

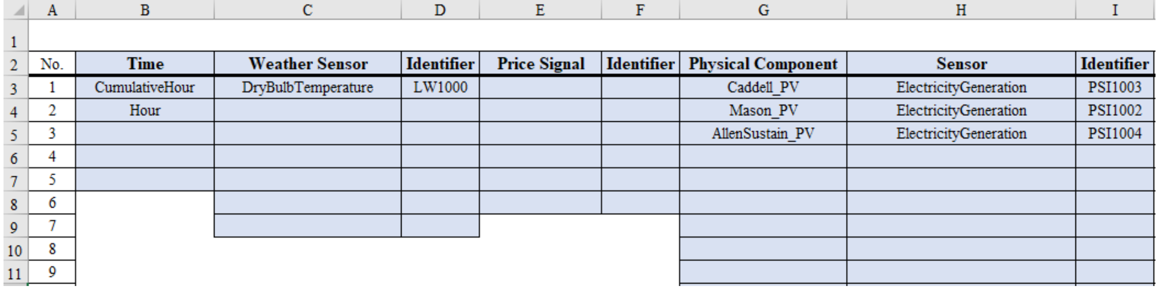
AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL
Physical Component	**Optimal Control Measure**	**Key**	**Data Type**	**Min**	**Cost 1**	**Max**	**Cost 2**	**DCOST**	**DMAX**
Caddell	TemperatureSetpoint	BEFMI1002	ContinuousReal	23	0	29	0	0	6
MSE	TemperatureSetpoint	BEFMI1007	ContinuousReal	23	0	29	0	0	6
Ford	TemperatureSetpoint	BEFMI1004	ContinuousReal	23	0	29	0	0	6
Klaus	TemperatureSetpoint	BEFMI1006	ContinuousReal	23	0	29	0	0	6
Crecine	TemperatureSetpoint	BEFMI1005	ContinuousReal	23	0	29	0	0	6
Savant	VoltageThrottling	BEFMI1001	ContinuousReal	0.8	0	1	0	0	0.2
Smithgall	VoltageThrottling	BEFMI1008	ContinuousReal	0.8	0	1	0	0	0.2
Mason	VoltageThrottling	BEFMI1009	ContinuousReal	0.8	0	1	0	0	0.2

Figure 40. Logical Layer Input Sheet – Optimal Control Variable Input

If rule-based control is present in the network, a logical layer sheet allows to instantiate sensors, actuators, and rule-based control statements. Figure 41 depicts the sensor part in the logical layer. The sensor part in the excel input file is comprised of four different columns categorized by sensor types (e.g., `Time`, `Weather`, `PriceSignal`, physical component sensor). A modeler can choose pre-defined available sensors by clicking the drop-down menu in each cell. Except for the `Time` sensor, as soon as one sensor is instantiated, its ID or Key is assigned to the selected sensor.

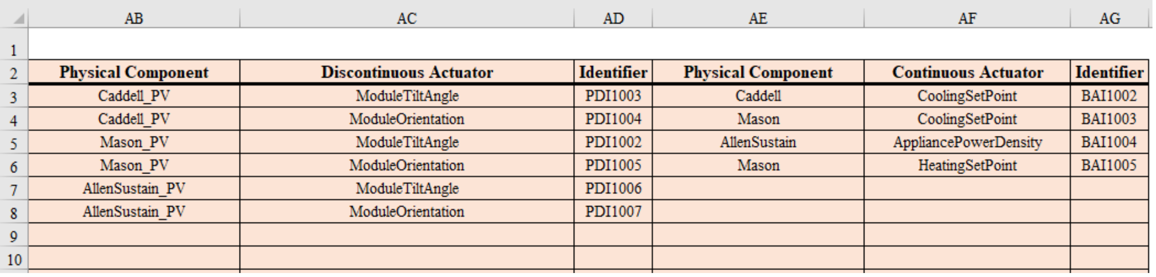
Actuators are instantiated similarly, as shown in Figure 42. It is divided into discontinuous and continuous actuators. Some specific pre-defined actuator values are only available for the discontinuous actuator, and continuous values can be used for continuous actuators. For the discontinuous actuator, the pre-defined value can be found in Table 9 and Table 10.

After finishing the definition of the physical layer, modelers file the logical layer input sheet if they include RBC and optimal control controls in conjunction with DO in the simulation. In the setup of the run-time simulation, the different control settings are distinguished by introducing the concept of “Simulation Type.” Figure 37 shows that this is set in the logical layer’s specification. Initially, modelers should select a simulation type and simulation scope. The available type is introduced in the “Simulation Type” table, and modelers type one of the integers from 0 to 10 based on the urban energy network simulation type. The “Genetic Algorithm Setting” table below is a table for setting the GA options. The “DO+Power Outage Setting” table is for setting the power outage start/end time steps. Figure 38, Figure 39, and Figure 40 show the DO and (parameterized) optimal control inputs cells, respectively. The DO input cells in Figure 38 have the “Physical Component” column that allows modelers to select one of the physical components from the physical layer sheet, and the “Energy Efficient Measure” column automatically shows the available design variables based on the physical component type. For instance, if `PV` type element is input in the “Physical Component” column as shown in Figure 38, only “ModuleArea” is shown in the “Energy Efficient Measure” column’s drop-down option because the `PV` element has only PV module area as a design variable. The “Data Type” column allows modelers to choose one of the “Continuous Real,” “Continuous Integer,” and “Discontinuous” options depending on the characteristic of the selected design variable. On the right side of the columns, modelers are required to input each design variable’s value and cost. There should be at least value 1,2 and cost 1,2 columns filled up. For the “Discontinuous” data type, the value and cost columns can be extended to up to six value and cost values. As shown in Figure 39, the optimal control input cells are identical to the DO’s input cells except for the fact that the “Energy Flexible Measure” column. The “Energy Flexible Measure” column shows the available optimal control variables depending on the physical component type. Figure 40 is similar to Figure 39, and the only difference is optimal control includes “DCOST” and “DMAX.”



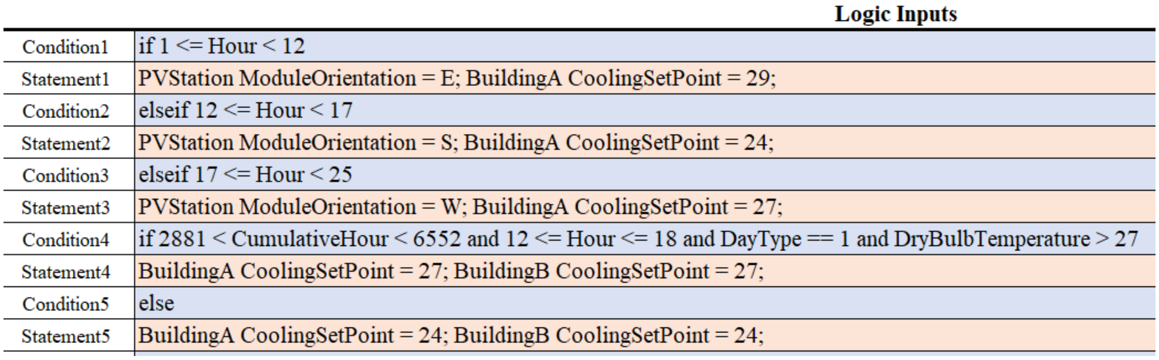
	A	B	C	D	E	F	G	H	I
1									
2	No.	**Time**	**Weather Sensor**	**Identifier**	**Price Signal**	**Identifier**	**Physical Component**	**Sensor**	**Identifier**
3	1	CumulativeHour	DryBulbTemperature	LW1000			Caddell_PV	ElectricityGeneration	PSI1003
4	2	Hour					Mason_PV	ElectricityGeneration	PSI1002
5	3						AllenSustain_PV	ElectricityGeneration	PSI1004
6	4								
7	5								
8	6								
9	7								
10	8								
11	9								

Figure 41. Logical Layer Input Sheet – Rule-based Control Sensor Input



	AB	AC	AD	AE	AF	AG
1						
2	**Physical Component**	**Discontinuous Actuator**	**Identifier**	**Physical Component**	**Continuous Actuator**	**Identifier**
3	Caddell_PV	ModuleTiltAngle	PDI1003	Caddell	CoolingSetPoint	BAI1002
4	Caddell_PV	ModuleOrientation	PDI1004	Mason	CoolingSetPoint	BAI1003
5	Mason_PV	ModuleTiltAngle	PDI1002	AllenSustain	AppliancePowerDensity	BAI1004
6	Mason_PV	ModuleOrientation	PDI1005	Mason	HeatingSetPoint	BAI1005
7	AllenSustain_PV	ModuleTiltAngle	PDI1006			
8	AllenSustain_PV	ModuleOrientation	PDI1007			
9						
10						

Figure 42. Logical Layer Input Sheet - Rule-based Control Sensor Actuator Input



	Logic Inputs
Condition1	if 1 <= Hour < 12
Statement1	PVStation ModuleOrientation = E; BuildingA CoolingSetPoint = 29;
Condition2	elseif 12 <= Hour < 17
Statement2	PVStation ModuleOrientation = S; BuildingA CoolingSetPoint = 24;
Condition3	elseif 17 <= Hour < 25
Statement3	PVStation ModuleOrientation = W; BuildingA CoolingSetPoint = 27;
Condition4	if 2881 < CumulativeHour < 6552 and 12 <= Hour <= 18 and DayType == 1 and DryBulbTemperature > 27
Statement4	BuildingA CoolingSetPoint = 27; BuildingB CoolingSetPoint = 27;
Condition5	else
Statement5	BuildingA CoolingSetPoint = 24; BuildingB CoolingSetPoint = 24;

Figure 43. Logical Layer Input Sheet – Rule-based Control Controller Input

It should be understood that at this stage, the modeler makes the association between actual installed sensors and actuators using the handles that the simulation models provide to their actual sensors (associated with a simulated outcome or observed state) and actual actuators (associated with a control variable or design parameter in the simulation module). As mentioned earlier, the association between an actual sensor and the sensor in the simulation module is not always direct. The reason for this is the abstraction and choice of granularity used in the simulation module. The examples will make this clearer and show that in practical cases, this is seldom a problem. It should be noted that in design optimization studies, there is not an actual sensor or actuator. In that case, the simulated sensor and actuator are "idealized" and hence identical (in abstraction and generality) to what the simulation modules expose.

With instantiated sensors and actuators, logic expressions can be constructed in the controller column. The controller allows to input some arithmetic operators (+, -, *, /), assignment operators (+=,-=), comparison operator (==, <=, >=, <, >), and logical operator (and). Further, the logic should be comprised of an "if-else" format with any number of "else if" statements. One example of logic is shown in Figure 43.

5.1.2 Rule-based Logic Conflict Detection

A set of logical expressions may have internal conflicts, so it is necessary that the rule-based specification logic is inspected after a UP instance is generated. One example of rule-based logic conflict is if two or more if-else statements control one actuator, which means that the actuator has potential circular relationships. The logic is examined by the rule-based logic conflict inspection algorithm shown in Figure 44.

The algorithm shown in Figure 44 initially defines a m x n matrix filled with all zeros (m: the number of sets, n: the number of instantiated actuators). A set is several grouped if-else statements. Then, if set m (m=1,2,…,M) controls an actuator n (n=1,2,…N), matrix(m,n)=1, otherwise, leave the matrix as it is. After this process, if any column's sum is greater than 1, the given logic is potentially conflicted, and the software is automatically terminated. If there is no column whose sum is greater than 1, the algorithm determines there is no potential logic conflict in the given logic.

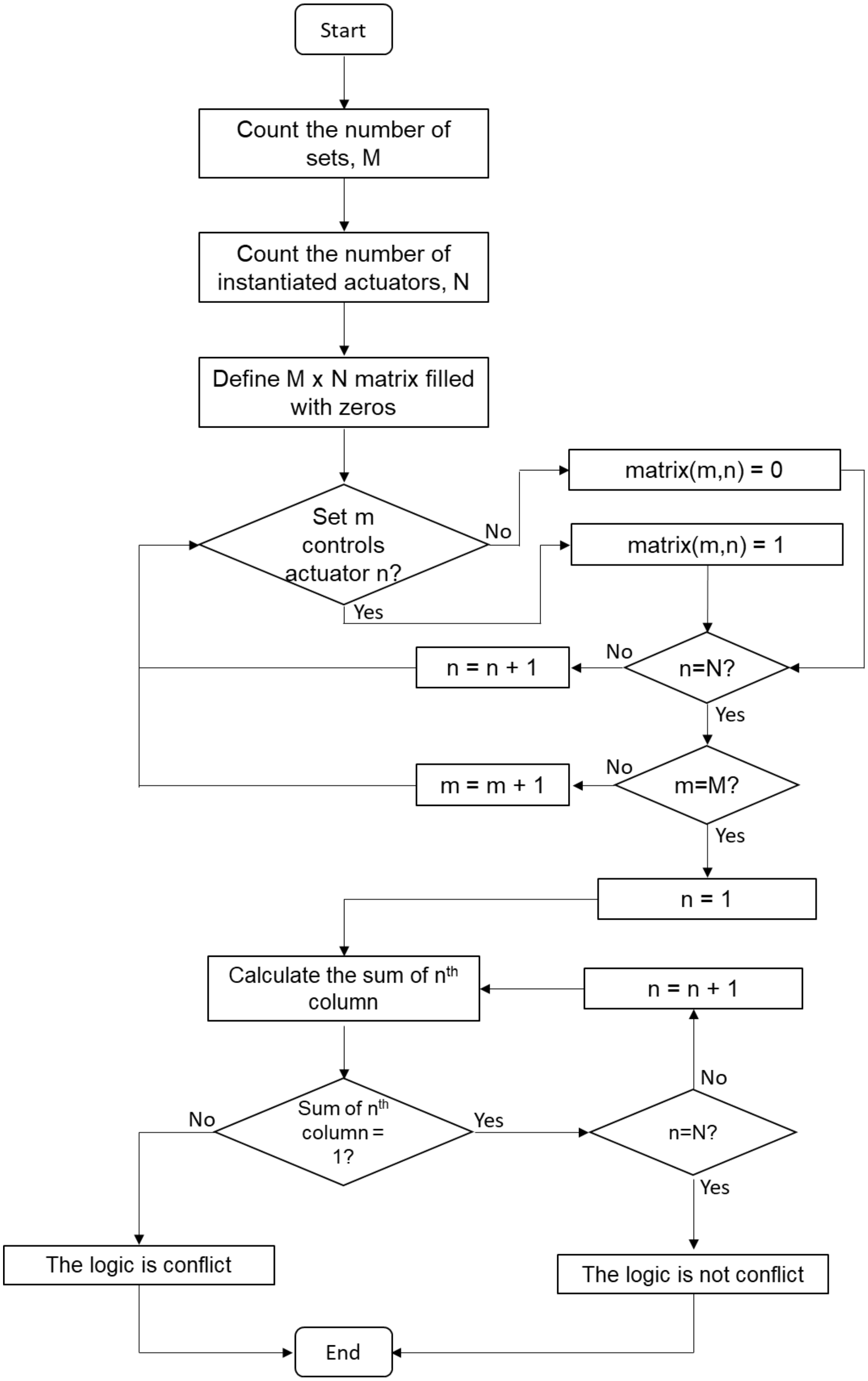
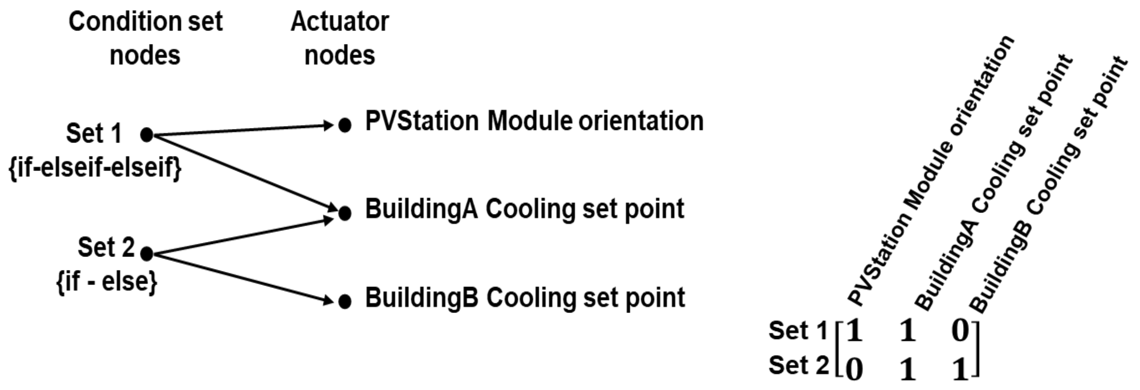


Figure 44. Rule-based Control Logic Conflict Detection Algorithm



(a) (b)

Figure 45. (a) Instantiated Sensor/Actuator from the Hypothetical Setting in Bipartite Graph (The arrow represents control) (b) Example Logic Matrix

As an example, the hypothetical scenario in Figure 3a is used. It is assumed that the logic in Figure 43 is used for the hypothetical scenario in Figure 3(a). Furthermore, the set and logic relationship in a bipartite graph and its corresponding matrix is shown in Figure 45. The logic in Figure 43 has two sets because there are two if-else statements. The first set is from conditions 1 to 3, and the second set is from conditions 4 to 5. The number of instantiated actuators in Figure 43 is 3 (PVStation ModuleOrientation, BuildingA CoolingSetPoint, and BuildingB CoolingSetPoint). The instantiated logic is converted to a bipartite graph format in Figure 45(a). Figure 45(a) shows that the set 1 controls PVStation ModuleOrientation and BuildingA CoolingSetPoint, and the set 2 controls BuildingA and BuildingB's CoolingSetPoint. Based on this information, a 2x3 matrix can be defined because there are two sets and three actuators. Figure 45(b) shows the completed matrix following the proposed logic conflict algorithm. It can be inferred that the proposed logic in Figure 43 is conflicting since the sum of the 2nd column equals 2, which means the BuildingA CoolingSetPoint actuator is controlled by two different sets of logic.

CHAPTER 6. APPLICATION OF THE UP SCHEMA

The purpose of the case studies is to demonstrate the UP schema can facilitate any combination of various controls and design optimization in the building-integrated energy network.

6.1 Ten Selected Buildings on the Georgia Tech Campus

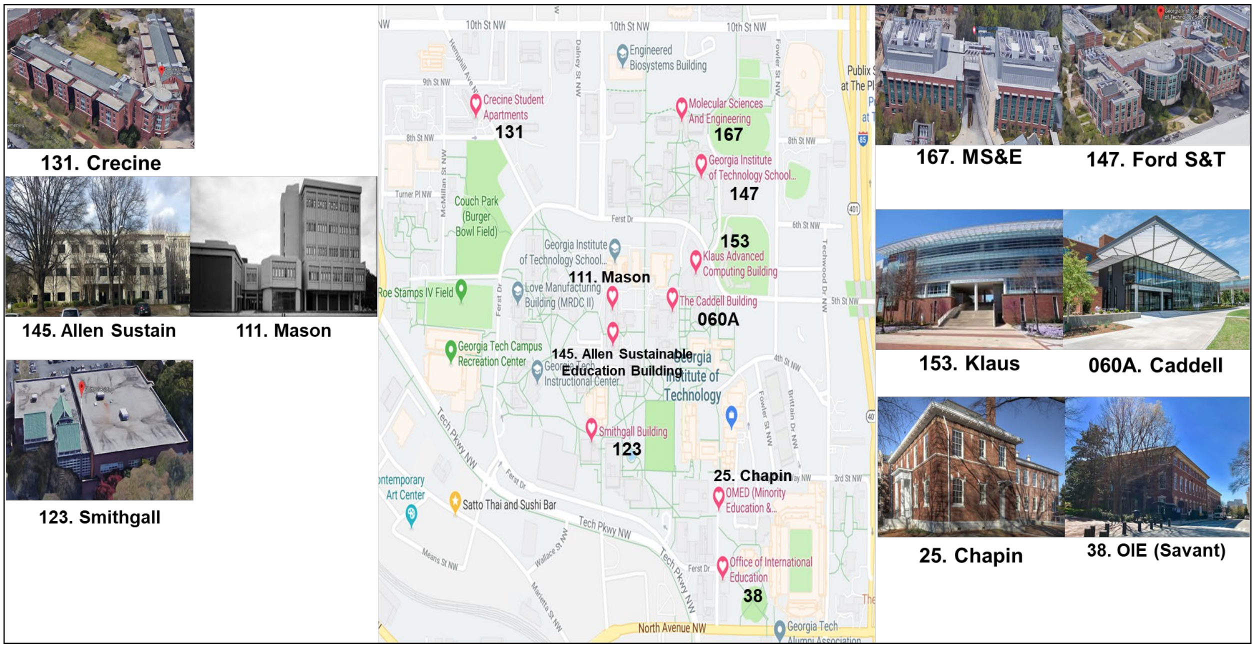


Figure 46. Selected Georgia Tech Buildings for the Test Cases

Figure 46 shows the selected Georgia Tech campus buildings used for the test cases. Table 20 tabulates the selected buildings' gross floor area, primary energy sources for heating, cooling, and DHW.

Table 21 shows the selected EVCS's total number of charging parking lots and their charging powers. Figure 47 shows the EVCS's schedule. The two selected EVCSs have the same schedules.

Furthermore, district heating and cooling plants provide steam and chilled water to some Georgia Tech campus buildings. The district heating and cooling plants provide all the selected buildings except the Caddell building for the selected buildings in this chapter. Caddell building is equipped with an electric VRF system that provides both heating and cooling. The district heating plant is equipped with boilers (COP: 0.75), generates steam, and distributes it throughout the campus. The district cooling has multiple electric chillers (COP: 4.45). It generates chilled water to satisfy the cooling requirements for Georgia Tech campus buildings (Lee 2012). A 10% distribution loss typical of thermal energy loss efficiency in district networks is employed for the test cases (CEN 2007b).

Table 20. Ten Selected Building's Information

Name	Conditioned Gross Floor Area [m^2]	Primary Energy Source for Heating	Primary Energy Source for Cooling	Primary Energy Source for DHW
Caddell	877	Electricity	Electricity	Electricity
Mason	5,444	Natural Gas	Electricity	Electricity
Allen Sustain	3,085	Natural Gas	Electricity	Natural Gas
MS&E	20,264	Natural Gas	Electricity	Natural Gas
Ford	26,840	Natural Gas	Electricity	Electricity
Klaus	19,372	Natural Gas	Electricity	Electricity
Chapin	524	Natural Gas	Electricity	Natural Gas
Savant	2,404	Natural Gas	Electricity	Electricity
Crecine	12,771	Natural Gas	Electricity	Electricity
Smithgall	3,957	Natural Gas	Electricity	Electricity

Table 21. Two Selected EVCS's Information

Name	Total Number of EV Charging Parking Lots [ea.]	Charging Power [kW]
Area4	6	7.2
W23Deck	4	7.2

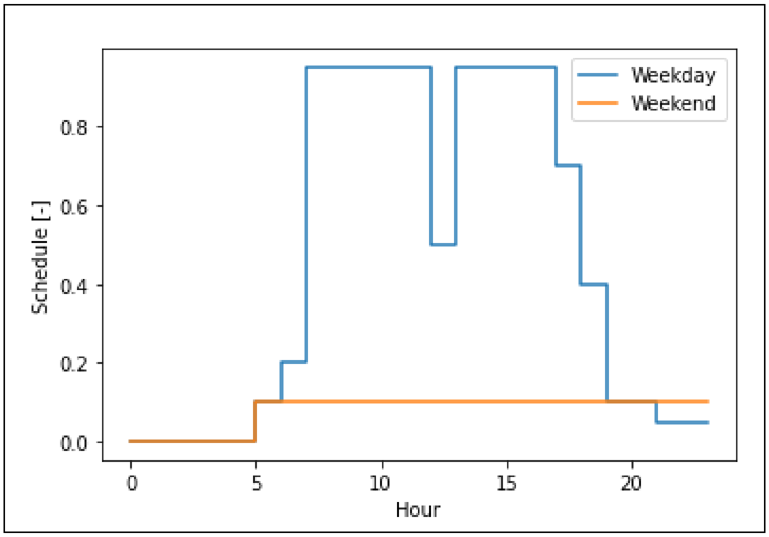


Figure 47. Two selected EVCS's Schedule

The electricity cost used in the Georgia Tech test cases is shown in Table 22 (Georgia Power 2021). Also, the year of 2018's natural gas prices are shown in Figure 50 (U.S. EIA 2021a).

Table 22. Electricity Cost for the Georgia Tech Test Cases

Electricity Cost ($/kWh)	Jun-Sep	Oct - May
On-Peak (2:00 p.m. to 7:00 p.m., Monday - Friday)	0.16923	0.074646
Off-Peak	0.074646	0.074646

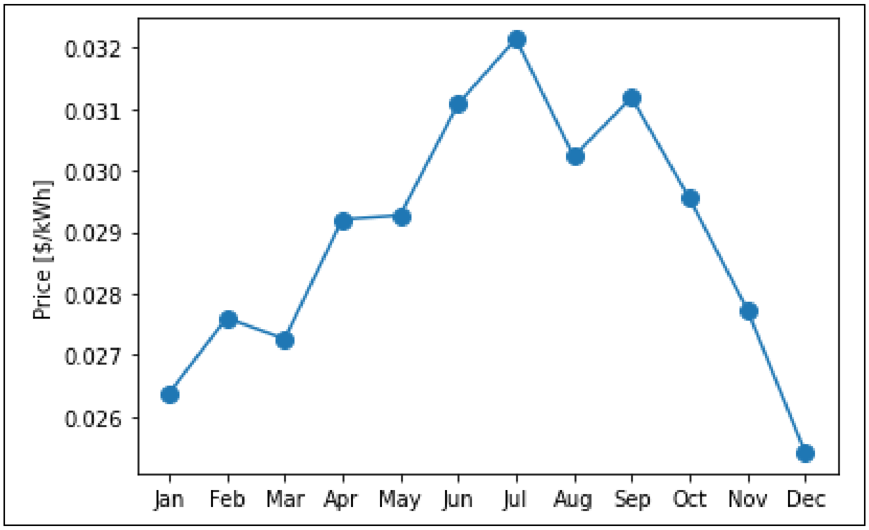


Figure 48. Natural Gas Cost for the Georgia Tech Test Cases

Figure 49 illustrates the Georgia Tech buildings' energy setting used in explicit settings 2 and 3. The campus buildings except the Caddell building are connected to the campus district system loops. The buildings are supplied with steam and chilled water from the district heating and cooling power plants. The community PV and electricity battery (Power Pack) nodes do not currently exist. However, the explicit settings 2 and 3 use these nodes to determine how much utility cost and primary energy can be saved from rule-based control and parameterized optimal control. The explicit setting 4 is tested on residential buildings in Sacramento, CA. This will be introduced more in detail in subsection 6.5.

- **Explicit Setting 1**: Energy Performance Base (As-is)
- **Explicit Setting 2**: Rule-based Control
- **Explicit Setting 3**: Design Optimization + Parameterized optimal control
- **Explicit Setting 4**: Design Optimization + Optimal control

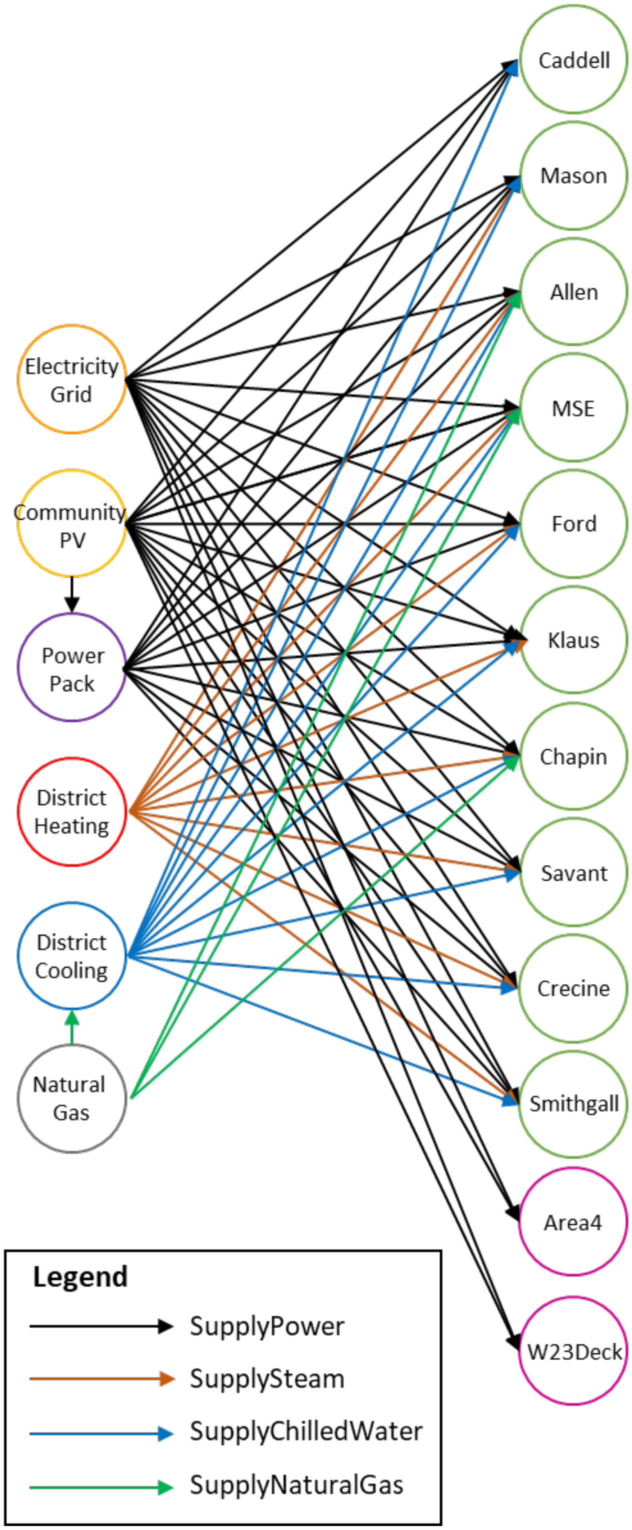


Figure 49. Directed Graph Representation Setting 2&3

6.2 Setting1: Energy Performance Base (As-is)

Setting 1 simulates the three buildings' baseline case (As-is).

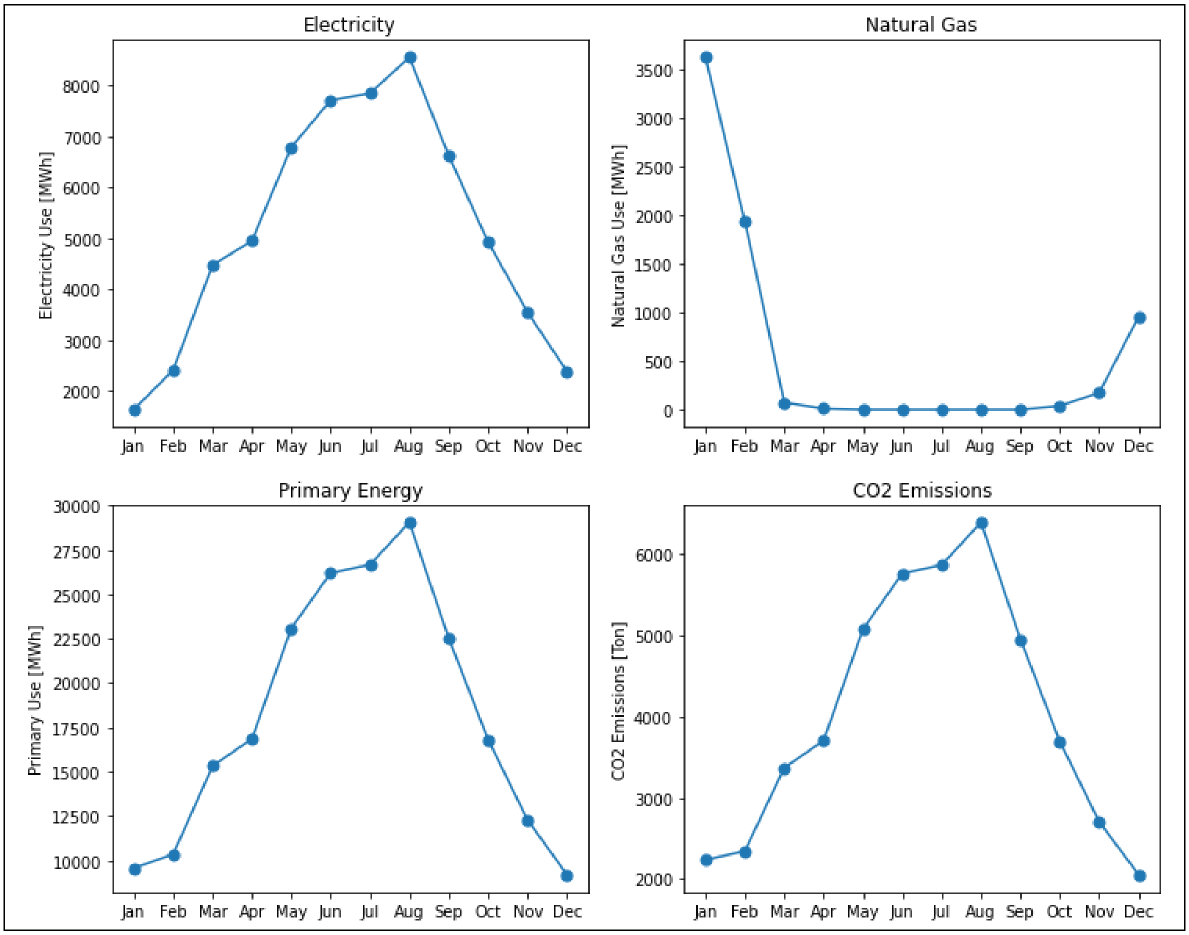


Figure 50. Performance Base Case Results for Setting 1

The base case is established to study potential energy efficiency improvements with different modifications.

6.3 Setting2: Rule-based Controls

Setting 2 includes a set of different RBC logics and a community PV with electricity battery capacities together. A total of six different variants (two RBC logics, only PV, and two electricity battery capacities, 200 and 400 kWh) are included in the evaluation. The community PV and electricity battery's input parameter data in Table 23 and Table 24. It is assumed that the community PV and electricity battery are installed at the Tech Green site near the selected buildings on the Georgia Tech campus.

Table 23. Community PV Information

Name	Module Area [m^2]	Orientation [-]	Tilt Angle [°]	Peak Power [W/m^2]	Performance Factor [-]
Community PV1	8,000	South	30	250	0.7
Community PV2	10,000	South	30	250	0.7

Table 24. Electricity Battery Information

Name	Capacity [kWh]	Charging/Discharging efficiency [%]	Charging/Discharging Power Limit [kW]	Minimum Stored Energy [kWh]	Self-discharging efficiency [-]
Power Pack 200	200	90/90	100/100	0	1
Power Pack 400	400	90/90	200/200	0	1

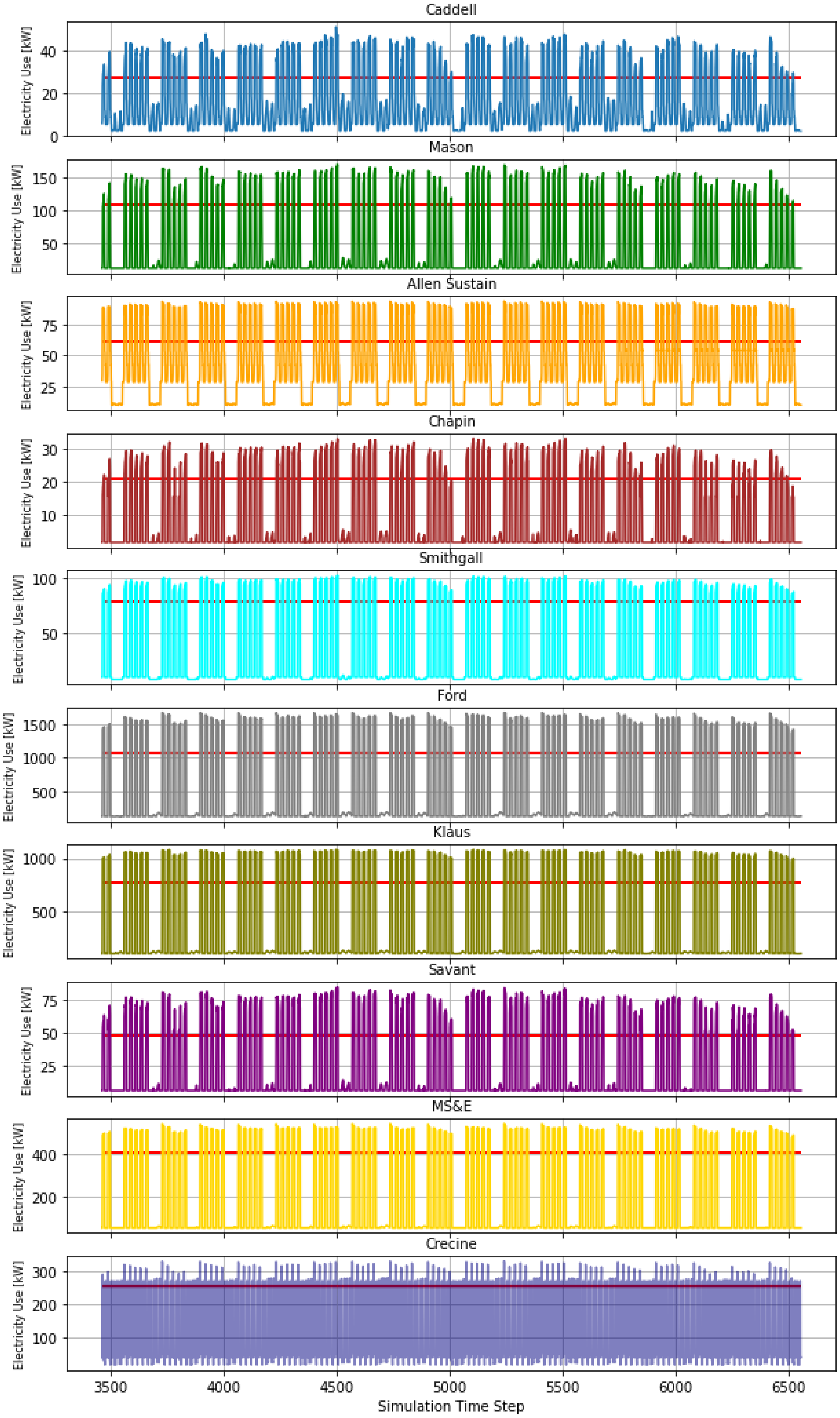


Figure 51. Electricity Use and the Trigger for DR for Logic (a)

Two different forms of RBCs, i.e., logic (a) and (b), are introduced. Their specifics are given in Appendix C. Logic (a) controls cooling setpoints in the ten Georgia

Tech buildings in a handcrafted demand response (DR) strategy. Logic (a) sends the DR signal independently to each building when on weekdays (“`DayType` == 1”) from 12 p.m. to 18 p.m. (“12 <= `Hour` <= 18”) from June to September (“3623 < `CumulativeHour` < 6552”) and building’s electricity consumption exceeds a certain amount, i.e., in Caddell 35 kW, in Mason 109 kW, and in AllenSustain 62 kW. Each building’s DR signal criterion is selected such that it activates when the electricity use intensity exceeds 0.02 kW/m^2 or 0.04 kW/m^2, depending on the building's electricity use. Figure 51 shows each building’s electricity use and DR trigger (red line) criterion. Logic (b) applies another form of DR control to each building. The three terms (“3623 < `CumulativeHour` < 6552 and 12 <= `Hour` <= 18 and `DayType` == 1”) are identical to logic (a). However, Logic (b) is executed based on the outdoor dry-bulb temperature and thus, the logic includes the temperature condition (“`DryBulbTemperature` > 27”).

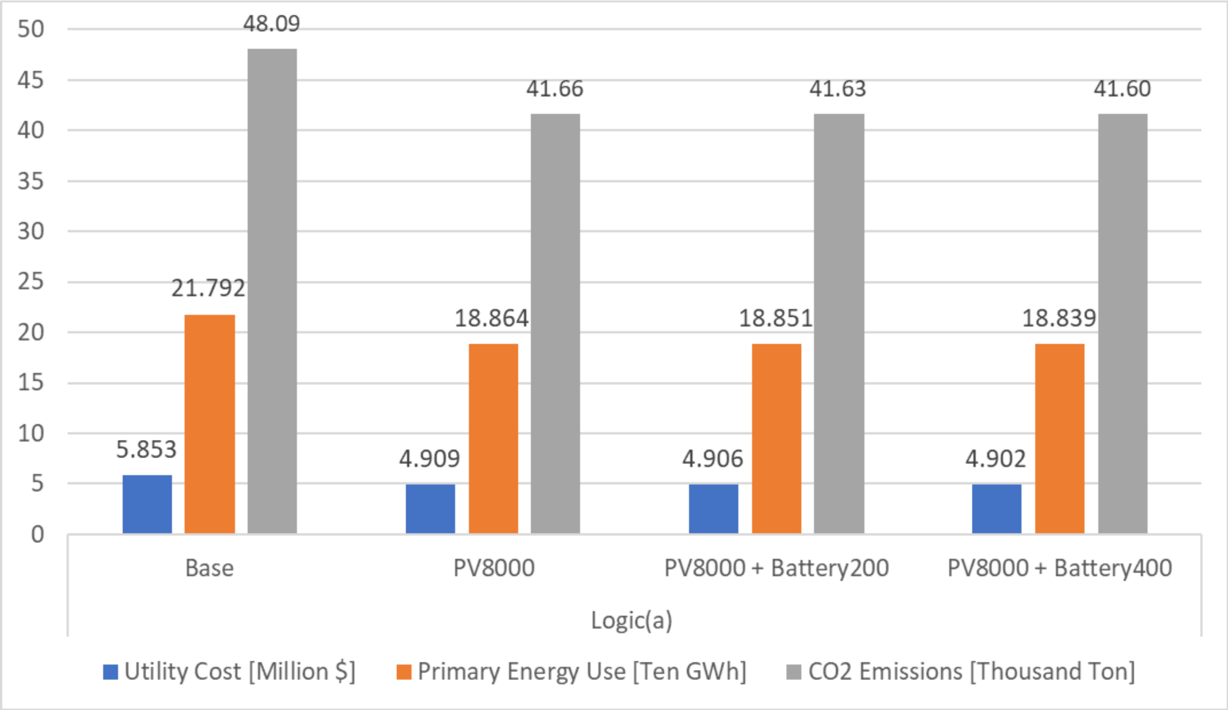


Figure 52. Results for Setting 2 if logic (a) and 8,000 m^2 PV are applied

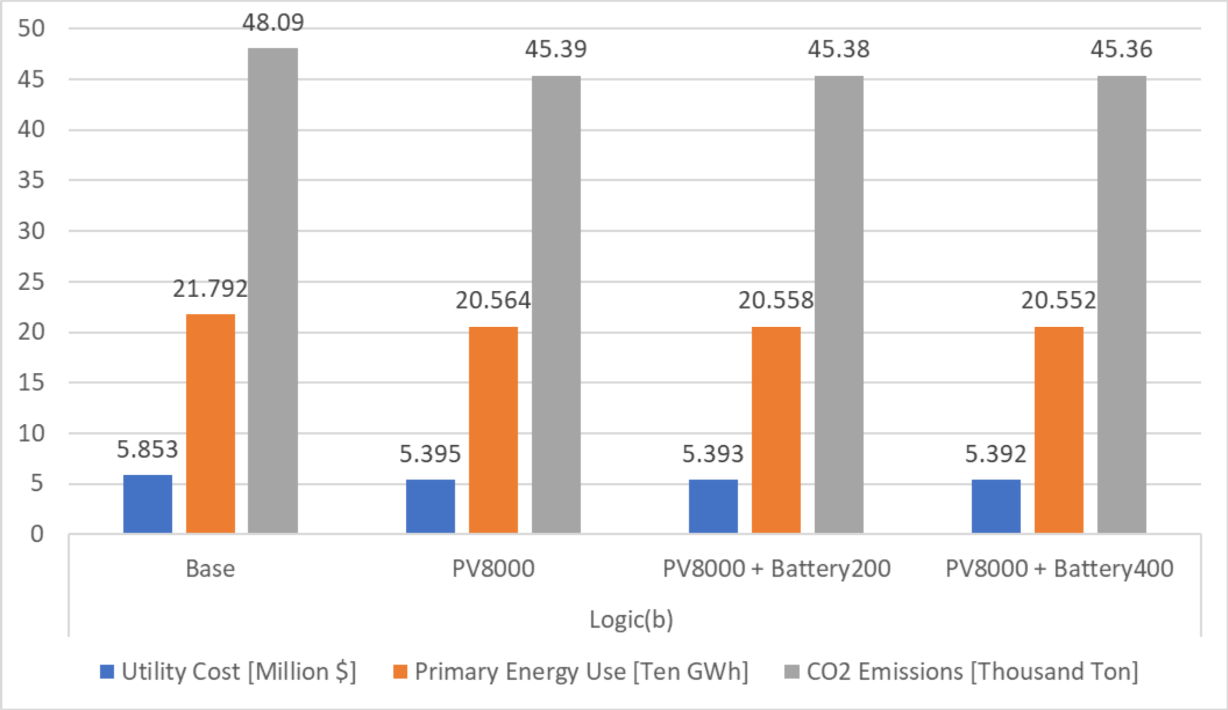


Figure 53. Results for Setting 2 if logic (b) and 8,000 m^2 PV are applied

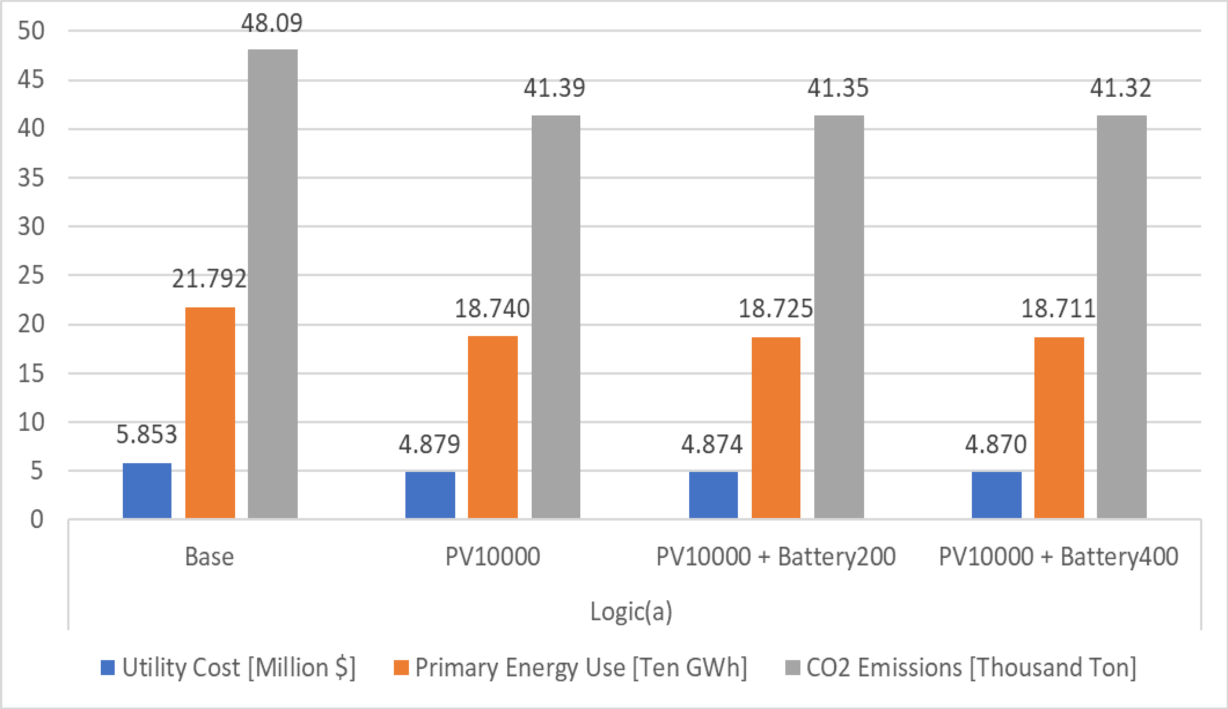


Figure 54. Results for Setting 2 if logic (a) and 10,000 m^2 PV are applied

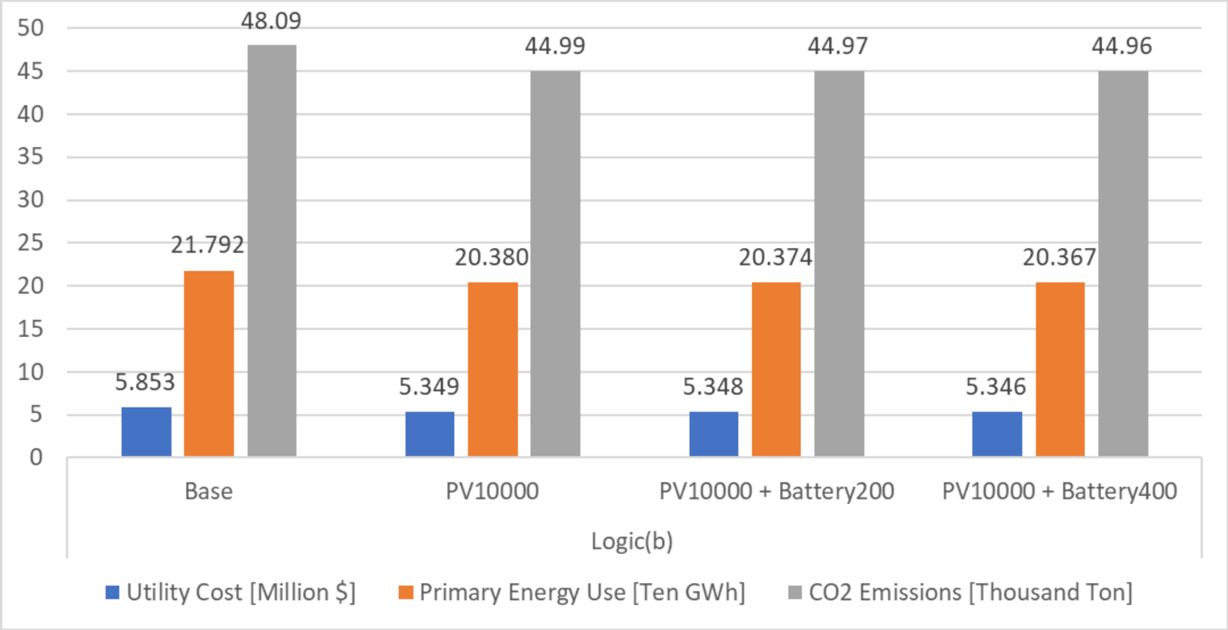


Figure 55. Results for Setting 2 if logic (b) and 10,000 m^2 PV are applied

Figure 52 and Figure 53 show setting 1's results when 8,000 m^2 community PV are applied. And Figure 54 and Figure 55 illustrate the setting's results with 10,000 m^2 community PV.

For logic (a)'s case (Figure 52 and Figure 54), the results show that when the 8,000 m^2 PV is installed, the utility cost and primary energy use, and CO_2 emissions decrease 16.1%, 13.4%, and 13.4%, respectively. This is mainly because the electricity is supplied initially from the power grid, but a lot of electricity is not supplied from the community PV, which is clean and free electricity. When the installations of the 200 kWh electricity batteries with 8,000 m^2 PV decrease the utility cost, primary energy use, and CO_2 emissions by 16.2%, 13.5%, and 13.4%. This reduction is lower than expected because the electricity battery's capacity is not optimally sized, as shown in Figure 56. Figure 56 shows the electricity battery state duration curve. It can be inferred that the electricity battery is barely used throughout the year. The energy-saving is not significant as already expected when the 400 kWh electricity battery is installed. This is also because the capacity of the electricity battery is too large for this test case. When the 10,000 m^2 community PV is installed, approximately 0.5% additional utility cost, primary energy use, and CO_2 emissions can be saved compared to the 8,000 m^2 community PV case.

For logic (b)'s case (Figure 53 and Figure 55), when only PV is installed, the utility costs, primary energy use, and CO_2 emissions are also dropped most. The utility cost, primary energy, and CO_2 emissions decrease by 7.8%, 5.6%, and 5.6%, respectively. Comparing logic (a) and logic (b)'s results, logic (a)'s utility costs for 8,000 m^2 PV are 8.3% lower than the 10,000 m^2 PV case. This is because, as Table 25 shows, the number of DR occurrences during the on-peak time for logic (a) is far higher for most buildings than for logic (b). More specifically, except for the Crecine building, the logic (a)'s DR is more executed to all other buildings than logic (b) even though the total number of logic (b)'s DR occurrences is higher. As Table 22 shows, the electricity cost during the on-peak hours is about 2.3 times higher than the off-peak hours. When the 200 kWh electricity battery is installed, 0.02% additional utility costs, 0.1% additional primary energy use, and CO_2 emissions are expected to decrease. Moreover, when the 400 kWh electricity is installed, 0.04% additional utility costs, 0.1% additional primary energy use, and CO_2 emissions are lowered. The minor savings from the electricity battery installations are the same as the logic (a)'s case. Figure 56 shows that the electricity battery usage hours are even shorter than logic (a)'s case. This is why when the electricity battery is installed with logic (b), the energy savings are less significant than the logic (a) case.

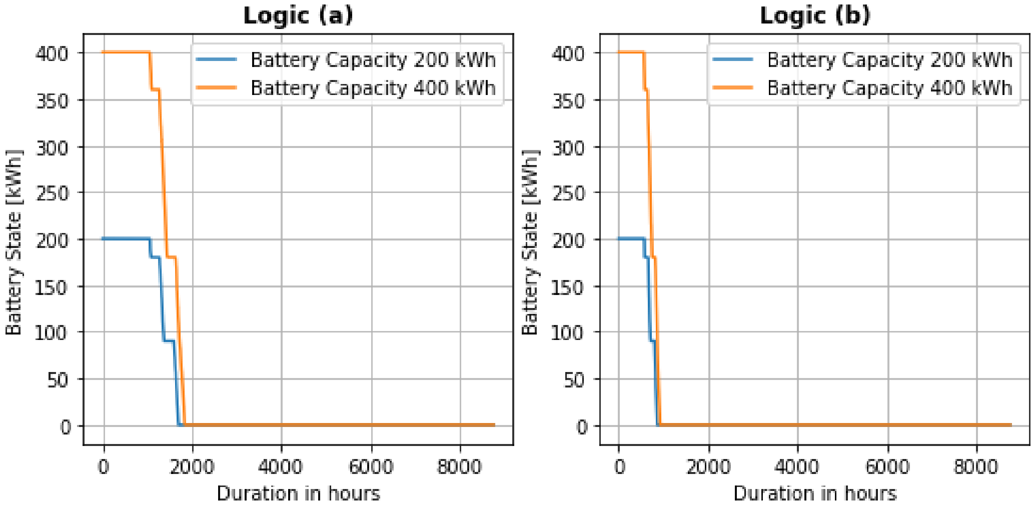


Figure 56. Electricity Battery State Duration Curve

Table 25. Number of Demand Response Logic Occurrences Comparison

Logic	Building	Time	Total no. of occurrence
Logic (A)	Caddell	On-Peak	313
		Off-Peak	103
	Mason	On-Peak	341
		Off-Peak	257
	Chapin	On-Peak	313
		Off-Peak	234
	Crecine	On-Peak	167
		Off-Peak	170
	Ford	On-Peak	348
		Off-Peak	261
	MSE	On-Peak	348
		Off-Peak	261
	Klaus	On-Peak	348
		Off-Peak	261
	Savant	On-Peak	348
		Off-Peak	264
	Smithgall	On-Peak	348
		Off-Peak	267
	Allen	On-Peak	348
		Off-Peak	261
Logic (B)	N/A	On-Peak	262
		Off-Peak	389

6.4 Setting3: Design Optimization + Parameterized Optimal Control

In this 'implicit' setting, there is no explicit predefined control setting. Instead, the desired solution is defined as the outcome of design optimization over the community PV module area, electricity battery capacity, roof emissivity, natural ventilation. We also add parameterized optimal control parameters, cooling setpoint temperature float, and lighting dimmer to see how this additional control parameter intervention affects the overall optimization. The optimization target is a minimal cost, so the first challenge is to attach installed cost figures to the sizing parameters, as shown in Table 26. To parameterize the DR intervention as an additional design variable, we introduce delta T as the amount by which the afternoon temperature can "float" above the standard cooling setpoint in the afternoon. For the productivity loss cost, we employ the productivity loss cost introduced in Equation (11).

To clarify, two simulations are run. The first simulation is conducted only with design variables as in Table 26. The second simulation is done with the design variables and the parameterized control variables in Table 27.

Table 26. List of Design Variables and Their Costs in Setting 3

Variable	Value		Cost	Applied building
	Min	Max		
PV module area	0	12,000	\$380/m^2	N/A
Battery capacity	0	3,000	\$400/kWh	N/A
Roof emissivity	0.4	0.9	\$16/m^2	Allen Sustain, Chapin, Ford, Savant
Natural ventilation opening area	0	5	\$150/window*	Caddell, Ford, MSE, Klaus,

\$150/window*- It is assumed that a 1 m^2 natural ventilation opening per window is installed.

Table 27. List of Control Variables and Their Costs in Setting 3

Variable	Value		Cost	Applied building
	Min	Max		
Lighting Dimmer	0	120	\$200/dimmer	Caddell
Lighting Dimmer	0	25	\$200/dimmer	MSE
Lighting Dimmer	0	40	\$200/dimmer	Ford
Lighting Dimmer	0	55	\$200/dimmer	Klaus
Temperature afternoon float	0	2.5	Productivity loss	Caddell, MSE, Ford, Klaus

Table 28. Additional Simulation Setup in Setting 3

	Value
Discount rate	0.02
Period of analysis	20 years
Thermal neutral temperature	25 °C
Average salary per occupant	$ 20
Productivity loss rate per every deg C	2 %

$$
\begin{gathered}
\min \sum_{t=1}^{8760} \{ (\text{utility cost}(t) + \text{ thermal comfort penalthcost}(t)) \cdot \\
\text{present value factor} + \text{retrofit upfront costs} \} \\
0\ \text{m}^2 \leq \text{PV module area} \leq 12{,}000\ \text{m}^2 \\
0\ \text{kWh} \leq \text{Electricity battery capacity} \leq 3{,}000\ \text{kWh} \\
0.4 \leq \text{Roof emissivity} \leq 0.9 \\
0\ \text{m}^2 \leq \text{Natural ventilation opening area} \leq 5\ \text{m}^2 \\
0 \leq \text{Lighting Dimmer} \leq \text{each building's maximum value} \\
0^\circ\text{C} \leq \text{Temperature afternoon float} \leq\ 2.5\ ^\circ\text{C} \\
\text{utility cost}(t) = \text{total electricity use}(t) \cdot \text{electricity cost}(t) + \text{natural gas use}(t) \cdot \text{natural} \\
\text{gas cost}(t),\ \forall t \in [1, 8760] \\
\text{retrofit upfront cost} = \sum technology\ value \cdot\ \ technology\ cost \\
\text{thermal penalty cost}(t) = (T_{thermal\ zone} - 25) \cdot L_{rate} \cdot S_{occ} \cdot N_{occ}, \\
\forall t \in [1, 8760] \\
\{\text{PV module area, Electricity battery capacity, Roof emissivity, Temperature} \\
\text{afternoon float}\} \in \mathbb{R} \\
\{\text{Natural ventilation opening area, Lighting Dimmer}\} \in \mathbb{Z} \\
\text{thermal penalty cost}(t) \geq 0,\ \forall t \in [1, 8760]
\end{gathered}
\tag{12}
$$

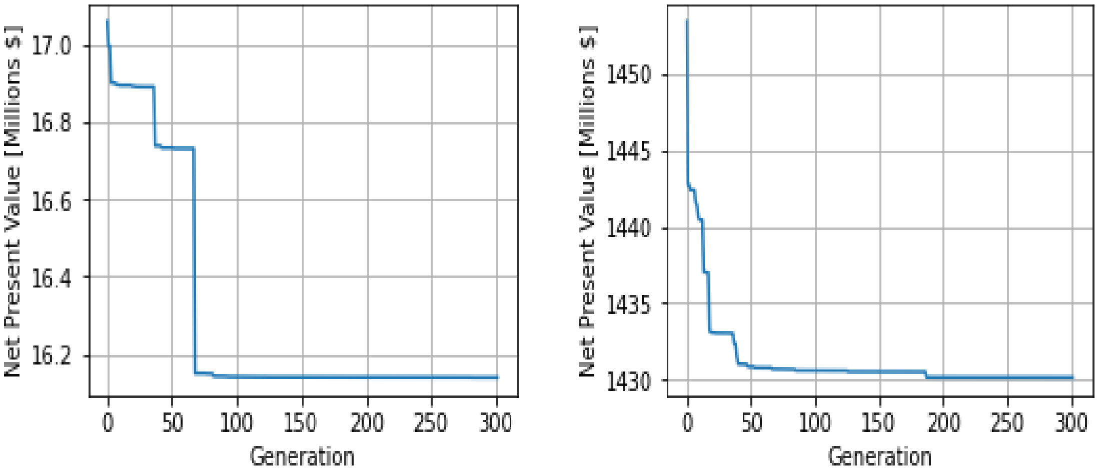
Equation(12) shows the optimization formulation for setting 3. The objective function consists of summating the utility costs, retrofit upfront cost, and thermal penalty comfort cost over the entire simulation time step. The constraint is comprised of the selected EEM and EFM minimum and maximum intervals. The PV module area, electricity battery capacity, roof emissivity, and temperature afternoon float parameters are real continuous variables. The natural ventilation opening area and lighting dimmer parameters are set as integer variables. Furthermore, the present value is calculated as shown in Equation (13) (Leland and Tarquin 2012).

$$\text{Present value} = \frac{(1+i)^{n}--1}{i\cdot(1+i)^{n}} \tag{13}$$

Where,

i: interest rate, [-]

n: analysis period, [year]



(a) **(b)**

Figure 57. Net Present Value (a) Without Control parameters (b) With Control Parameters

Figure 57 shows the objective function value over the genetic algorithm's generation. Figure 57(b)'s net present value is much higher than Figure 57(a) since the former only includes the occupants' productivity loss cost.

Table 29. Optimal Parameter Results in Setting 3

Name		Optimal value	
		Without Control Parameters	With Control Parameters
Roof Emissivity [-]	Allen Sustain	0.90	0.90
	Chapin	0.90	0.89
	Ford	0.90	0.90
	Savant	0.90	0.90
Community PV Area [m^2]		6,940	6,932
Electricity Battery Capacity [kWh]		11.7	12.4
Temperature Control [°C]	Caddell	N/A	0.01
	MS&E		0.00
	Ford		0.01
	Klaus		0.04
Natural Ventilation [m^2]	Caddell	N/A	0.0
	MS&E		0.0
	Ford		0.0
	Klaus		0.0
Lighting Dimmer [ea.]	Mason	N/A	2
	Crecine		0
	Savant		0
	Smithgall		38

Table 29 shows the results of the test case. We find that for both the design optimization parameter only case and the integrated control parameter case, the result indicates that investing in the PV installation is the most economical decision. This is because the cost of PV can directly lower the utility costs. However, the community PV area can't be increased much because of the PV module cost constraint. The results also show that the electricity battery's optimal capacity is low for both cases since there is not much excessive electrical energy from the community PV. This is because the electricity battery usage is too low in setting 2, and the optimal value in setting 3 is found to be a smaller community PV area and electricity battery capacity.

The investment on the design parameters such as roof emissivity and natural ventilation is not recommended. Firstly, the roof emissivity helps to keep the heat from the sun out. However, this is not cost-effective decision given the material cost. Secondly, the natural ventilation variable is not selected because the natural ventilation variable lowers approximately 6-9% of the cooling load, but as mentioned earlier, the Georgia Tech campus has high-efficient district cooling plant (COP: 4.45). Thus, this cooling load reduction is not needed considering the investment costs.

We find that increasing the temperature setpoint in the afternoon is rarely part of the optimal selection for the integrated control case. This is not surprising and in line with earlier findings. The underlying reason is that our aggressive estimate of the productivity loss cost overwhelms the gains from energy saving. For the lighting dimmer, the Smithgall and Mason buildings have 38 and 2 dimmers, respectively. The reason is that the two buildings consume far higher lighting powers than other buildings that do not have lighting dimmers. Since the lighting dimmer in the parameterized optimal control is set as in Equation (14). According to this equation, the optimal value (38 ea. lighting dimmer) leads to about 28% (=0.7+0.3*(40-38)/40) of lighting energy use from 12 p.m.

to 4 p.m. in the summer months in the Smithgall building. The overall cost saving from the dimmer installation in the Smithgall building is $ 7,747 over 20 years. This value is calculated by multiplying the yearly lighting power saving ($473.8) and discount factor (0.02) over 20 years (=16.35)). Hence, the cost-saving ($ 7,747) is more significant than the lighting dimmer initial investment cost ($7,600). Furthermore, the lighting energy saving helps to reduce the total cooling load. This result shows why the lighting dimmer option is selected in the Smithgall building.

$$\text{Lighting dimming factor} = 0.7 + \frac{(\text{max no. of dimmer - optimal no of dimmer})}{\text{max no. of dimmer}} \qquad \textbf{(14)}$$

6.5 Setting4: Design Optimization + Optimal Control

Setting 4, which incorporates design optimization and optimal control for assumed power outage situation, selected three residential buildings in Sacramento, CA. Unlike the previous settings, we switch from the Georgia Tech campus buildings to the residential buildings in CA. The university campus is usually equipped with a distributed power generator for backup power and stores powder to prepare the emergency situation such as power outages. However, this information cannot be obtained. Thus, we proceed with the design optimization and optimal control for the power outage situation in this setting.

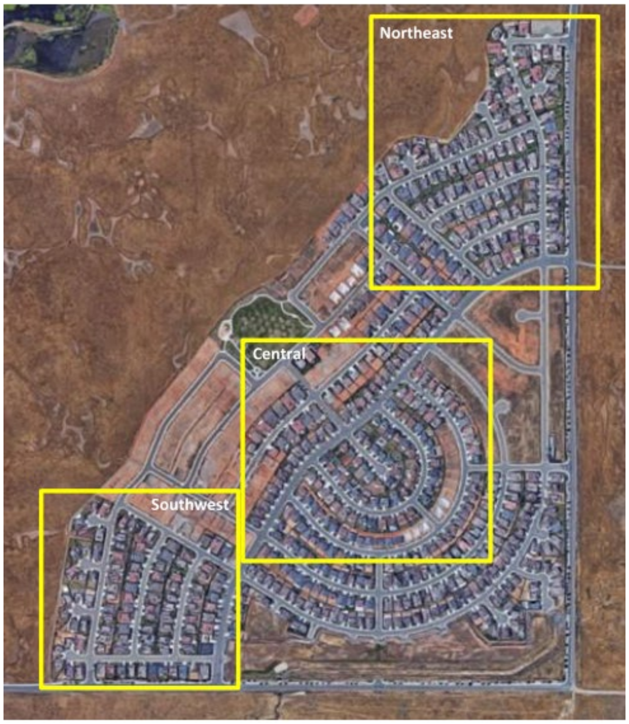


Figure 58. Anatolia Community – Rancho Cordova, Sacramento County, CA (Carneiro 2017)

There are fifteen homes in the Anatolia Community located in Sacramento County, CA. The nine homes are located in the southwest part of the community, five homes in the central part, and one home in the northeast portion (Carneiro 2017). For this test case, the three selected buildings in the southwest portion are located in the Southwest in Figure 58. The selected home's simulation models are all calibrated models against the actual energy data. It should be noted that setting 4 utilizes the 5R3C thermal model instead of EPC because of the 5R3C models' faster computational time. The 5R3C model is introduced in detail in Appendix A.

Table 30. Building Archetypes in Anatolia Community Energy Model Information, Adapted from (Carneiro 2017)

Name	Address*	Community Location	Building Orientation
T03	5xx0 Almond Falls Way	Southwest	E
T04	5xx2 Almond Falls Way	Southwest	W
T05	11xx7 Aspen Heights Ct	Southwest	S

Table 31. Three Selected Residential Building's Information

Name	Conditioned Gross Floor Area [m^2]	Primary Energy Source for Heating	Primary Energy Source for Cooling	Primary Energy Source for DHW
T03	260	Natural Gas	Electricity	Natural Gas
T04	260	Natural Gas	Electricity	Natural Gas
T05	260	Natural Gas	Electricity	Natural Gas

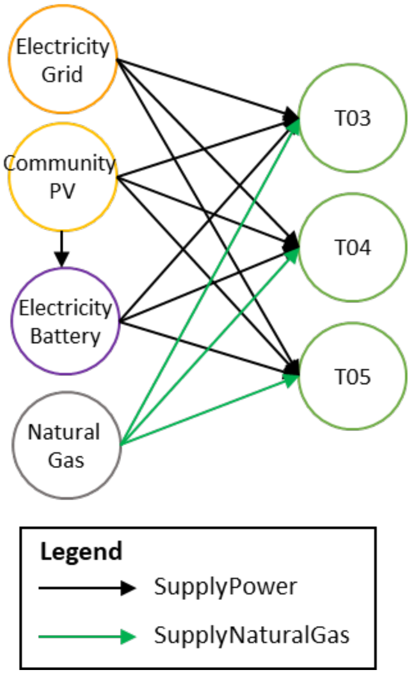


Figure 59. Directed Graph Representation Setting 4

Table 32 shows the electricity cost used in the test case (SMUD 2021), and Figure 60 shows the natural gas cost (U.S. EIA 2021b).

Table 32. Electricity Cost for the Anatolia Community Test Case

Electricity Cost ($/kWh)	Jun – Sep	Oct - May
12 am-12pm	0.1035	0.1277
12pm-5pm	0.1035	0.1765
5 pm-8 pm	0.1430	0.3105
8pm-12am	0.1035	0.1765

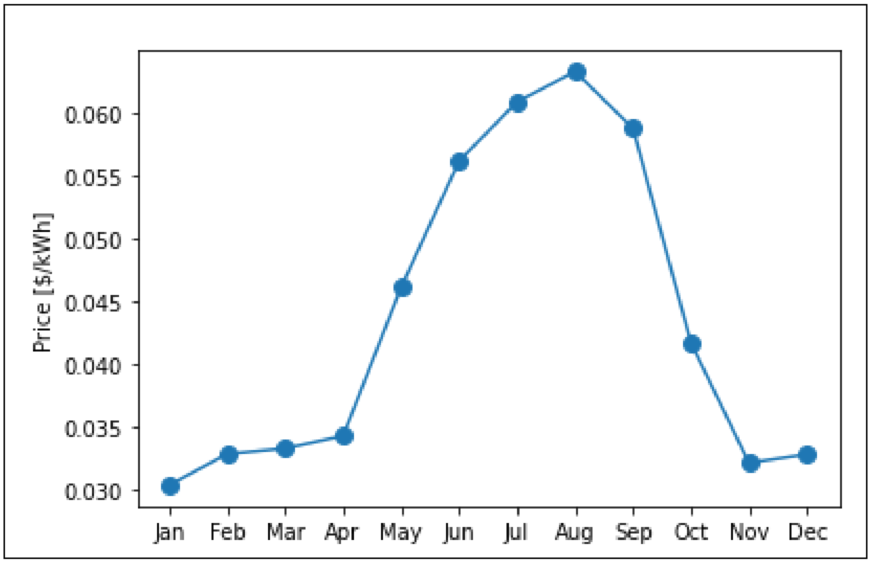


Figure 60. Natural Gas Cost for the Anatolia Community Test Case

We assume that the power outage occurs 1.5 days on the hottest week of the year for this test case. Figure 61 shows the weekly average outdoor temperature and the 29th week is the hottest week in Sacrament, CA, in the TMY3 file. Hence, we select the power outage on Tuesday noon of the 29th week for 1.5 days. In addition, we also include the subcase that assumes the occupants reduce the lighting and appliance power by 30%. Moreover, to run the power outage simulations, the developed code is set to run the day before the power outage period plus the hours before the power outage the day for the warm-up period. For instance, for this test case, the warm-up period is also 1.5 days because the power outage occurs at noon, and thus the warm-up period is 1.5 (=1 + 0.5 (12 hours)).

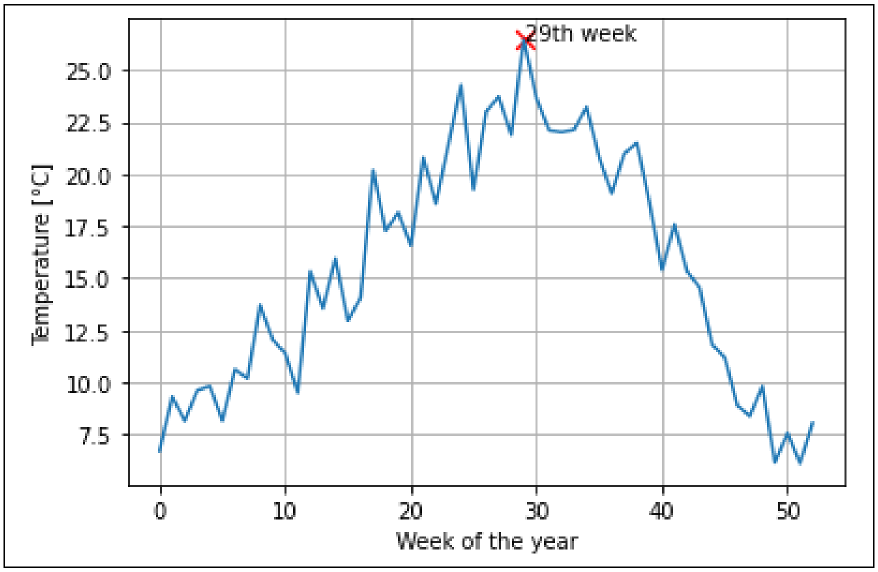


Figure 61. Weekly Average Dry-bulb Temperature of Sacramento, CA in TMY3

Table 33 tabulates the selected variables for the optimization and their costs. We choose the PV module, electricity battery, and temperature setpoint for the optimization. This test case aims to find how buildings should work together to avoid natural disasters and find the most cost-effective investment to withstand the power outage situation to increase resilience.

Table 34 shows the simulation setup in setting 4. The only difference from setting 3 is we increase the thermal neutral temperature by 2 °C compared to setting 3 (= 25 °C). We assume that when a power outage occurs, occupants can endure more heatwaves than normal days.

In addition, we develop a new 5R3C thermal model and make use of it for this test case because of the performance of the EPC model. The 5R3C model has fewer equations; still, the results are guaranteed compared to the EPC model.

Table 33. List of Design and Control Variables Used in Setting 4

Variable	Value		Cost	Applied Building
	Min	Max		
PV module area	0	1,000	$537/m^2	N/A
Battery capacity	0	150	$400/kWh	N/A
Temperature setpoint	24	29	Productivity loss cost	T03, T04, T05

Table 34. Additional Simulation Setup in Setting 4

	Value
Discount rate	0.02
Period of analysis	20 years
Thermal neutral temperature	27 °C
Average salary per occupant	$ 25
Productivity loss rate per every deg C	2 %

$$\min \sum_{t=4763}^{4800} (\text{utility cost}(t) + \text{ thermal comfort penalty cost}(t)) \cdot \text{present value factor} + \text{retrofit upfront cost}$$

s.t.

$$0\ \text{m}^2 \leq \text{PV module area} \leq 1{,}000\ \text{m}^2 \tag{15}$$

$$0\ \text{kWh} \leq \text{Electricity battery capacity} \leq 150\ \text{kWh}$$

$$24°\text{C} \leq \text{Temperature setpoint}(t) \leq 29°\text{C},\ \forall t \in [4763, 4800]$$

utility cost(t) = total electricity use(t)·electricity cost(t) + natural gas use(t) ·natural

$$\text{gas cost}(t)\ ,\ \forall t \in [4763, 4800]$$

$$\text{retrofit upfront cost} = \sum technology\ value \cdot\ technology\ cost$$

$$\text{thermal penalty cost}(t) = (T_{thermal\ zone} - 25) \cdot L_{rate} \cdot S_{occ} \cdot N_{occ},$$

$$\forall t \in [4763, 4800]$$

$$\{\text{PV module area, Electricity battery capacity, Temperature setpoint}\} \in \mathbb{R}$$

$$\text{thermal penalty cost}(t) \geq 0,\ \forall t \in [4763, 4800]$$

Equation (15) shows the optimization formulation for setting 4. The cost function consists of summating the utility costs, retrofit upfront cost, and thermal penalty comfort cost. The constraint is comprised of the selected EEM and EFM minimum and maximum intervals. The time variable t varies only from 4,763 to 4,800, which is a total of 3 days. This is because the optimization is conducted for the warming-up period of 1.5 days and the assumed power outage period of 1.5 days. The present value factor is calculated as in Equation (13).

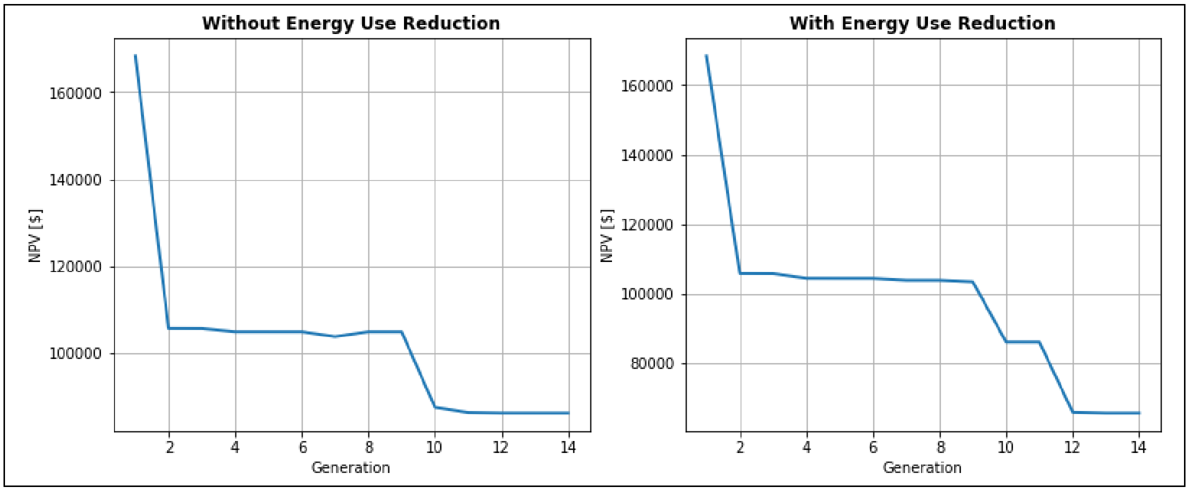


Figure 62. Cost Function Value Over the Genetic Algorithm Generation

Figure 62 illustrates the cost function values for both with/without 30% lighting/appliance use reduction cases.

Table 35. Optimal Design Parameter Results in Setting 4

Name	Optimal Value	
	Without 30% lighting/appliance energy reduction	With 30% lighting/appliance energy reduction
Community PV Area [m^2]	154.4	116.0
Electricity Battery Capacity [kWh]	84.1	83.5

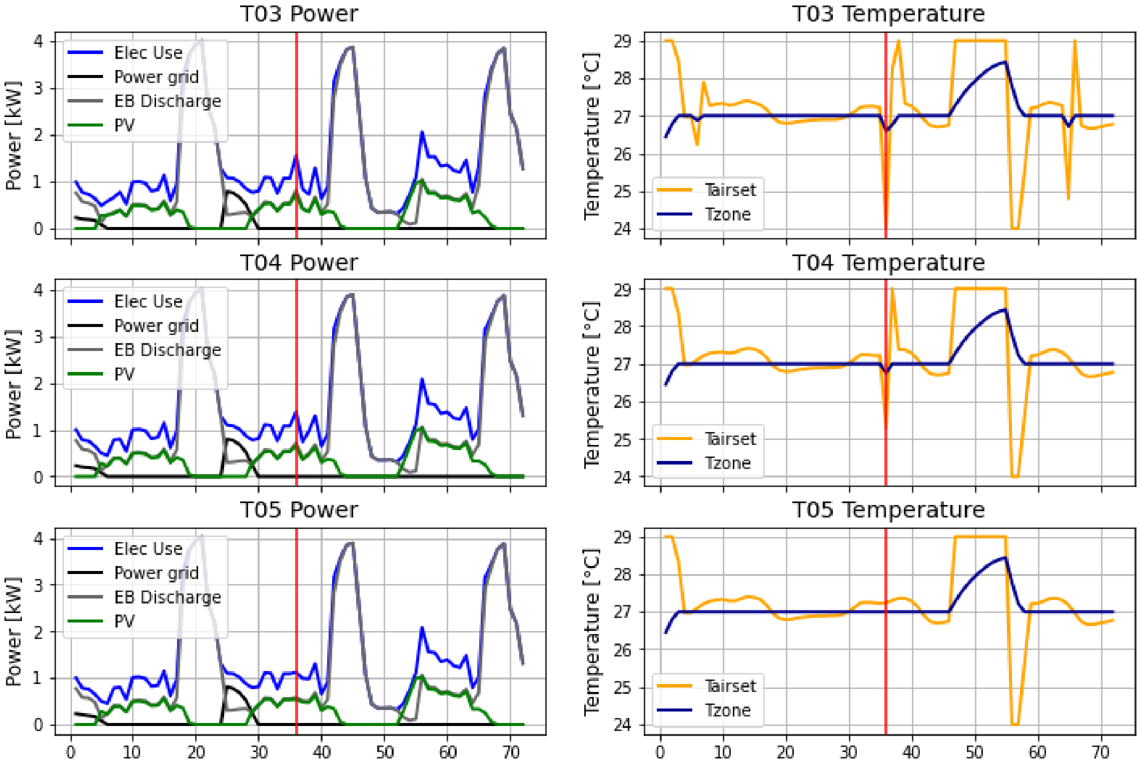


Figure 63. Power Flows for Physical Components and Building Indoor Temperature for No Lighting/Appliance Use Reduction (The red line indicates the start of the power outage)

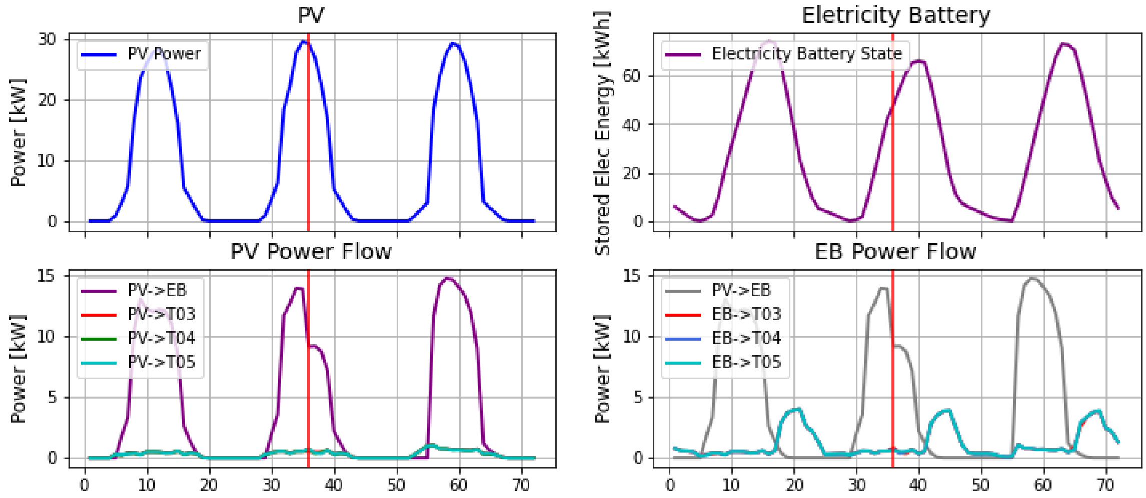


Figure 64. PV Electricity Generation and Stored Energy in Electricity Battery and Power Flows for No Lighting/Appliance Use Reduction (The red line indicates the start of the power outage)

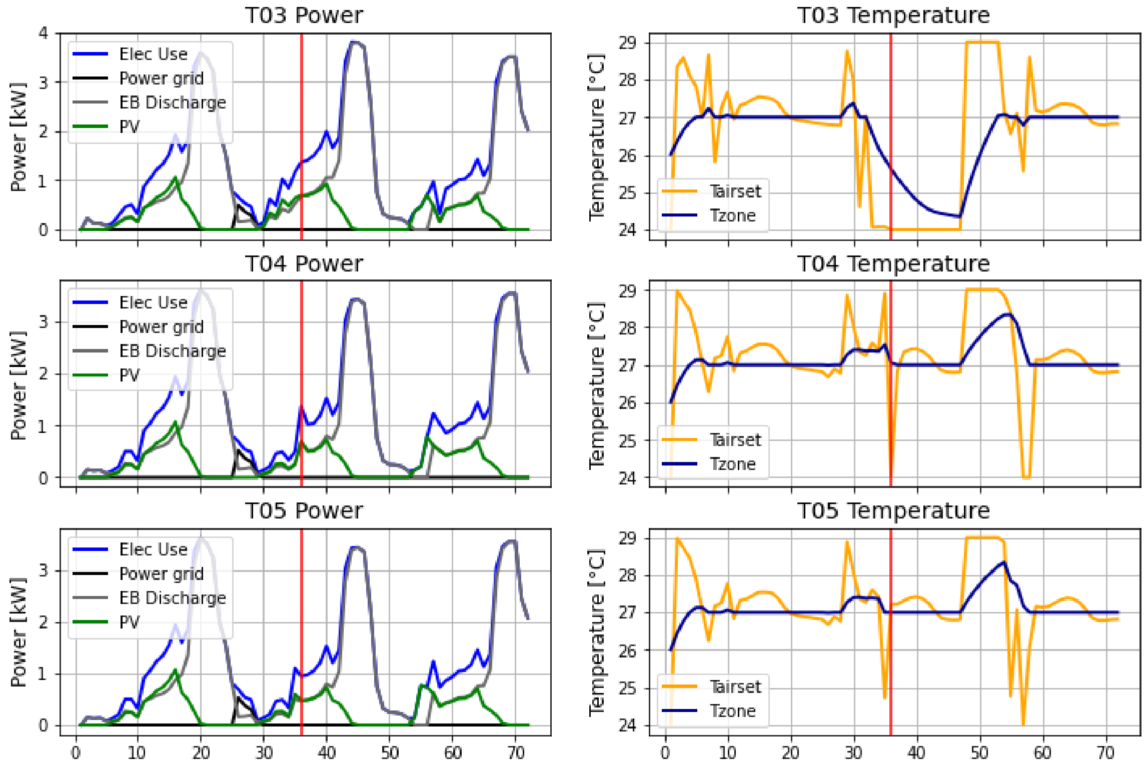


Figure 65. Power Flows for Physical Components and Building Indoor Temperature for 30% Lighting/Appliance Use Reduction (The red line indicates the start of the power outage)

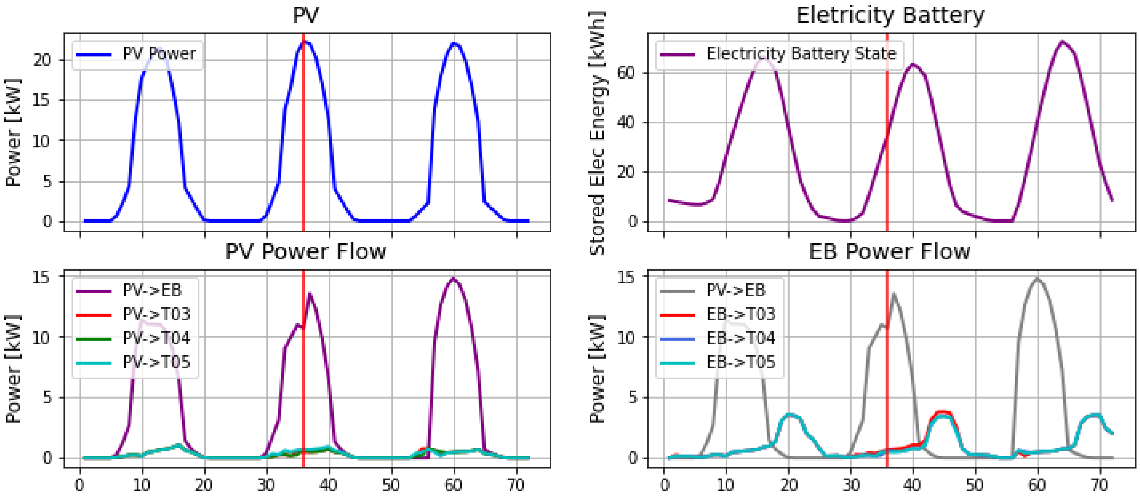


Figure 66. PV Electricity Generation and Stored Energy in Electricity Battery and Power Flows for 30% Lighting/Appliance Use Reduction (The red line indicates the start of the power outage)

Table 35 shows the optimization results. We find that the without lighting/appliance energy reduction case requires a larger PV module area than the with lighting/appliance energy reduction case, as expected. However, the capacity of the electricity battery is similar to each other.

Figure 63 and Figure 65 illustrate the buildings' electricity consumption, power flows, indoor temperature, and setpoints. We find that the electrical energy demands are met from PV during the daytime. The surplus electrical energy from PV is charged to the electricity battery. Since the three residential buildings have high electrical demands in the evening, it is essential to install the electricity battery to avoid the power outage disaster. Besides, the temperature setpoints always try to maintain the indoor temperature at 27 °C to avoid the occupants' thermal comfort penalty cost. The T03's temperature graph in Figure 63 illustrates that when the indoor temperature rises, the temperature setpoint is immediately lowered to 24 °C, which is the lower bound of the temperature setpoint.

Figure 64 and Figure 66 show the PV electricity generation, stored energy in the electricity battery, and power flows. The electricity battery’s stored energy peaks about 5 hours afternoon. This is because electricity is generated mostly around noon. The surplus electricity from PV is charged to the electricity battery. The stored energy is discharged around evening time when the buildings consume most electrical energy.

This case study shows how the physical components (e.g., buildings and PV) should act to survive during the power outage disaster. The test case also shows how design parameters (PV area, electricity battery capacity) and control parameters (temperature setpoint) can be optimized together using the UP schema.

CHAPTER 7. CONCLUSIONS

This thesis explores a novel application for systematic energy performance assessment at a large scale in grid-interactive efficient buildings. The thesis employs reduced-order physics-based simulation models as an underlying simulation engine. The advantage of using the reduced-order models is they can support energy-related macro design and control decisions rapidly.

The novel approach of the thesis is the fact that the assessment makes use of the UP schema. The UP schema enables to capture the urban prosumer settings and the system semantics of collections of buildings with integrated energy flow control logic. With the developed simulation platform, complex campus-scale design and control energy decisions can be made with the UP schema. Furthermore, since the simulation platform incorporates the UP schema, the simulation platform is scalable to more extensive portfolios and energy systems. Thus, the simulation platform is flexible enough to explore diverse physical elements with design and control topologies straightforwardly. The physical layer of the UP schema includes the physical elements and energy supply relationships that use the directed graph theory. The logical layer of the UP schema conceptualizes the elements needed for optimal design and control. Having a robust underlying representation for urban prosumer scenarios have the ability to perform complex energy scenarios on computer simulations.

The thesis conducts case studies to verify the hypothesis. Ten Georgia Tech campus buildings are modeled and analyzed using the UP schema and developed simulation platform. The test cases demonstrate how much energy can be saved from rule-based control and find energy-related optimal campus planning. Moreover, the test case conducted in Sacramento, CA, proves the usability of the UP schema by finding optimal PV module and electricity battery capacity design with optimal temperature setpoint control with the UP schema.

7.1.1 Future Potential Updates on the UP Schema and Developed Software

First of all, the UP schema can further conceptualize more physical components such as fuel cell or thermal storage, etc. More diverse relationships in the physical layer, such as `SupplyHydrogen` can also be added. As introduced in CHAPTER 2, this is possible because when the UP schema is initially designed, extensibility is one of the design philosophies and thus does not require to input simulation model detail inputs.

Secondly, The logical layer in the UP schema can conceptualize more various control schemes such as proportional–integral–derivative (PID), multi-agent controls, model predictive control, etc. to rapidly implement the diverse controls for GEB. The logical layer can also conceptualize more diverse knowledge other than the controls. The logical layer can also conceptualize the objective function and simulation mode (e.g., rule-based, optimal control, power outage, etc.). The current version of the developed Python software is hard-coded to conduct optimizations only with the presumed objective function (e.g., minimal cost). However, when the UP schema conceptualizes more information about the objective function, more diverse optimization such as environmental regulation objective (CO_2 emissions) will be available.

Furthermore, the current version of the UP schema and the developed software do not have a feature that considers important energy aspects such as demand charges. This can be either conceptualized in the UP schema or can also be hard-coded in the software. However, this should be added to the energy scenario analysis to make robust decisions.

7.1.2 Future Potential Research on Grid-interactive Efficient Buildings

This thesis proves that the UP schema in the software can facilitate the control and design optimization in GEB. The following is recommended in future research to advance the current state of the art in this thesis.

Firstly, it is recommended to convert the simulation models into linear models. A few reasons are i) linear models guarantee global optimum when optimization is conducted ii) linear models are solved faster than non-linear models. The results between the original (primarily non-linear) and linear models should also be verified. Even though PV model comparisons are briefly introduced in Appendix B, this experiment should be expanded to buildings and any other non-linear simulation models.

Secondly, utilizing data-driven or physics-based data-driven models with the UP schema should be researched. For these days, many buildings are equipped with smart or sub meters, and it is getting easier to obtain necessary data for energy estimation, control, etc.. As mentioned earlier, the UP schema is general enough to be applied to a variety of different types of models. This is not different for data-driven models. One interesting study would be to analyze a neighborhood consisting of some physics-based simulation models, and others are data-driven models. And see how the analysis preparation time, overall optimization/control execution time, and their results.

Thirdly, uncertainty in GEB should be taken into consideration. This is because all the information required to apply bottom-up physics-based models is not always directly available, and assumptions from complementary sources are required to provide the required model inputs. A certain level of assumptions uncertainty exist and should be taken into account (J. N. B. Fonseca and Oliveira Panão 2017). This idea can be further developed to apply stochastic control schemes.

Fourthly, whether complex relationships in GEB can be represented and control/optimization can also be conducted using deep learning algorithms such as graph neural networks should be researched. This may correspond mainly to when the data-driven models are used, but there is the possibility to be used for the physics-based models. The UP schema can also be used to automate the process since the schema utilizes the direct graph theory.

APPENDIX A. DESCRIPTION OF DEFAULT SUBHEADING SCHEME

Figure 67 shows the schematic of the developed 5R3C thermal model. The sky temperature (T_{sky}) is referred from (Clark, G, Allen 1978). Furthermore, the system energy consumptions such as HVAC, lighting, appliance, DHW remain the same as the original EPC and thus are omitted in this section.

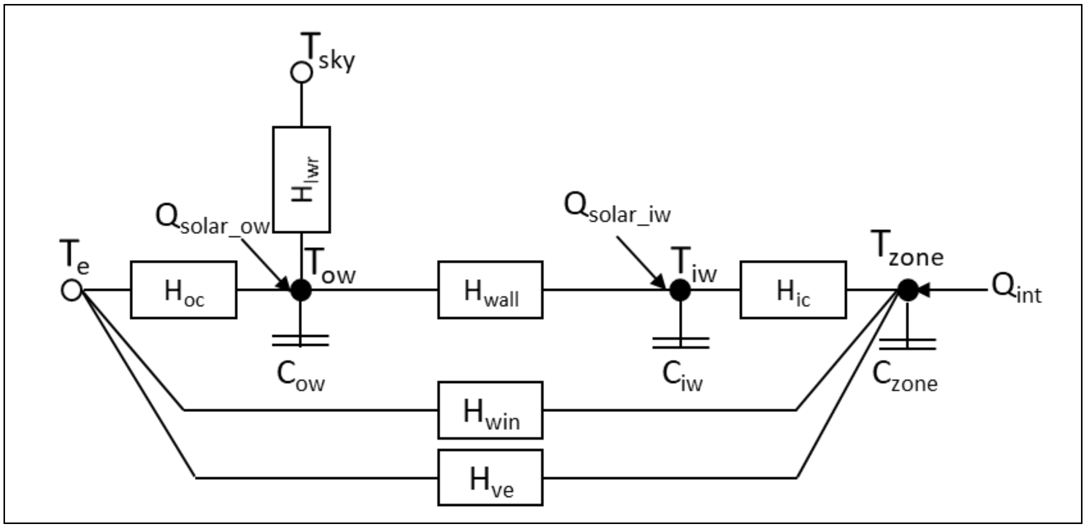


Figure 67. 5R3C Thermal Model

$$\begin{gathered}
T_{sky} = \{(0.787 + 0.7641 \cdot \ln(\tfrac{T_{dp}}{273})) \cdot (1+0.0224n-0.0035n^2+0.00028n^3)\}^{0.25} T_a \\
H_{lwr} = h_{rad} \cdot \varepsilon_{rad} \cdot F_{view} \\
Q_{solar\_ow} = Q_{solar} \cdot A_{opaque} \cdot \alpha \\
Q_{solar\_iw} = \psi_i \cdot Q_{solar} \cdot A_{window} \cdot \tau \cdot F_{overhang} \cdot F_{Fin} \cdot F_{horizon} \cdot SRF \cdot (1-F_{win\_frame}) \\
H_{env} = \frac{A_{opaque}}{A_{envelop}} U_{opaque} + \frac{A_{roof}}{A_{envelop}} U_{roof} \\
\varepsilon_{rad} = \frac{A_{opaque}}{A_{envelop}} \varepsilon_{opaque} + \frac{A_{roof}}{A_{envelop}} \varepsilon_{roof}
\end{gathered} \tag{16}$$

$$mC_p \frac{dT_{ow}}{dt} = H_{oc}A_{env}(T_e - T_{ow}) + H_{env}A_{env}(T_{iw} - T_{ow}) + H_{lwr}A_{env}(T_{sky} - T_{ow}) + Q_{solar\_ow} \tag{17}$$

$$\frac{dT_{ow}}{dt} = \frac{3600}{0.5C_{ow}}\{H_{oc}A_{env}(T_e - T_{ow}) + H_{env}A_{env}(T_{iw} - T_{ow}) + H_{lwr}A_{env}(T_{sky} - T_{ow}) + Q_{solar\_ow}\}$$

$$mC_p\frac{dT_{iw}}{dt} = H_{ic}A_{env}(T_{zone} - T_{iw}) + H_{env}A_{env}(T_{ow} - T_{iw}) + Q_{solar\_iw}$$

$$\frac{dT_{iw}}{dt} = \frac{3600}{0.5C_{iw}}\{-(H_{ic}+H_{env})A_{env}T_{iw} + H_{env}A_{env}T_{ow} + H_{ic}A_{env}T_{zone} + Q_{solar\_iw}\}$$

$$mC_p\frac{dT_{zone}}{dt} = H_{ic}A_{env}(T_{iw} - T_{zone}) + H_{win}A_{win}(T_e - T_{zone}) + H_{ve}A_{zone}(T_e - T_{zone}) + \omega Q_{int}$$

$$\frac{dT_{zone}}{dt} = \frac{3600}{\xi\cdot C_{zone}}\{-(H_{ic}A_{env} + H_{ve}A_{zone} + H_{win}A_{win})T_{zone} + H_{ic}A_{env}T_{iw} + H_{win}A_{win}T_{zone} + H_{ve}A_{zone}T_e + \omega Q_{int}\}$$

$$Q_{HC,nd} = H_{ic}A_{env}(T_{air,set} - T_{iw}) + H_{win}A_{win}(T_{air,set} - T_e) + H_{ve}A_{env}(T_{air,set} - T_e) - \omega Q_{int}$$

$$\frac{dT_{zone,ac}}{dt} = \frac{3600}{f_{int\_mass}\rho_{air}V_{air}C_p}\{-(H_{ic}A_{env} + H_{ve}A_{zone} + H_{win}A_{win})T_{zone} + H_{ic}A_{env}T_{iw} + H_{win}A_{win}T_{zone} + H_{ve}A_{zone}T_e + Q_{HC,nd}\}$$

Where,

T_e: Ambient temperature, [K]
T_{sky}: Sky temperature, [K]
T_{ow}: Outdoor wall temperature, [K]
T_{iw}: Internal wall temperature, [K]
T_{zone}: Thermal zone air temperature, [K]
T_{dew}: Dew-point temperature, [K]
U_{opque}: Opaque envelope U-value, [W/(m^2·K)]
U_{roof}: Roof U-value, [W/(m^2·K)]
H_{oc}: Outdoor convection resistance, [W/(m^2·K)]
H_{lwr}: Long-wave radiation resistance, [W/(m^2·K)]
H_{wall}: Envelope resistance, [W/(m^2·K)]
H_{win}: Window resistance, [W/(m^2·K)]
H_{ve}: Ventilation resistance, [W/(m^2·K)]
H_{ic}: Internal convection resistance, [W/(m^2·K)]
h_{rad}: Radiation heat transfer coefficient, [W/(m^2·K)]
ε_{rad} Emissivity, [-]
F_{view}: View factor between building and sky, [-]
α: Absorptivity, [-]
ξ: Thermal zone internal mass factor, [-]

ψ_i: Monthly internal solar radiation contribution factor (i=1,2, …, 12), [-]

ω: Internal heat gain contribution factor, [-]

τ: Window solar transmittance, [-]

F_{fin}: Fin solar reduction factor, [-]

$F_{overhang}$: Overhang solar reduction factor, [-]

$F_{horizon}$: Mutual building shading solar reduction factor, [-]

SRF: Blind solar reduction factor. [-]

F_{win_frame}: Window frame factor (assumed 0.3), [-]

Qsolar_ow: Solar radiation cast on outdoor wall, [W]

Qsolar_iw: Solar radiation cast on internal wall, [W]

C_{ow}: Outdoor wall heat capacitance, [J/(kg·K)]

C_{iw}: Internal wall heat capacitance, [J/(kg·K)]

C_{zone}: Thermal zone heat capacitance, [J/(kg·K)]

C_p: Total material heat capacitance, [J/(kg·K)]

A_{opaque}: Opaque envelope area, [m^2]

A_{roof}: Roof envelope area, [m^2]

A_{win}: window area, [m^2]

A_{zone}: Thermal zone area, [m$_2$]

m: Envelope mass, [kg]

Q_{HC_nd}: Thermal need, [W]

Q_{int}: Internal heat gain (occupancy + lighting + appliance), [W]

n: opaque sky cover in tenths, $(0 \leq n \leq 10)$

t: Time, [hr]

A.1 Result Validation

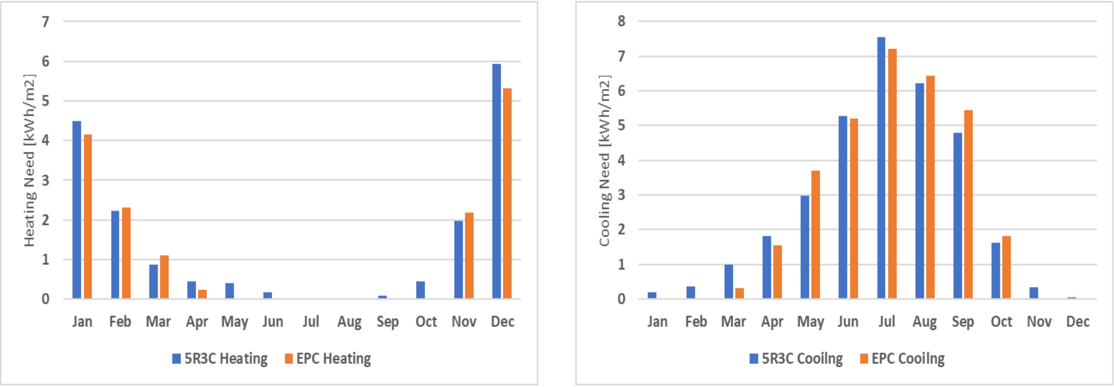
The 5R3C thermal model's results are verified against the EPC results. We choose the heating and cooling needs as the performance indicator because the thermal needs are calculated based on the thermal zone temperature and impact the building system energy consumption. It should be noted that the parameters (H_{oc}, H_{ic}, h_{rad}, ξ, ψ_i, ω) are calibrated against the corresponding EPC model. This is because H_{oc}, h_{rad}, ξ, and ψ_i parameters do not exist in the 5R3C model and H_{ic} and ω parameters are not certain because of the simulation engines' difference. The calibration's objective function and constraints are shown in Equation (18).

$$\min\{(\sum Q_{heating\_need\_5R3C} - \sum Q_{heating\_need\_EPC})^2 + (\sum Q_{cooling\_need\_5R3C} - \sum Q_{cooling\_need\_EPC})^2\}$$

s.t.

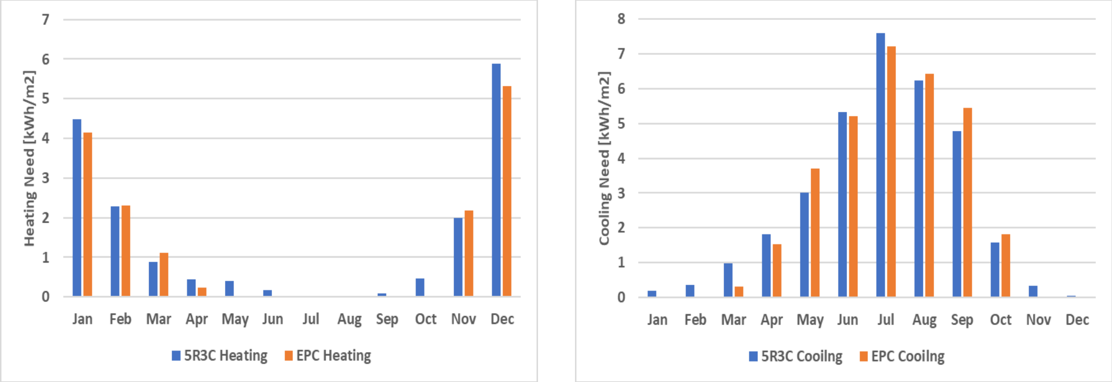
$$\begin{gathered} 1\ \mathrm{W/(m^2 \cdot K)} \le H_{oc} \le 30\ \mathrm{W/(m^2 \cdot K)} \\ 1\ \mathrm{W/(m^2 \cdot K)} \le H_{ic} \le 7\ \mathrm{W/(m^2 \cdot K)} \\ 1\ \mathrm{W/(m^2 \cdot K)} \le h_{rad} \le 10\ \mathrm{W/(m^2 \cdot K)} \\ 1 \le \xi \le 10 \\ 0 \le \psi_i \le 1, \forall i \in [1,12] \\ 0 \le \omega \le 1 \end{gathered} \qquad \textbf{(18)}$$

The following figures illustrate the result comparisons between the 5R3C and EPC models, as shown from Figure 68 to Figure 70. We also verify the accuracy of the 5R3C model to the four Georgia Tech campus buildings, and the results are shown from Figure 71 to Figure 74.



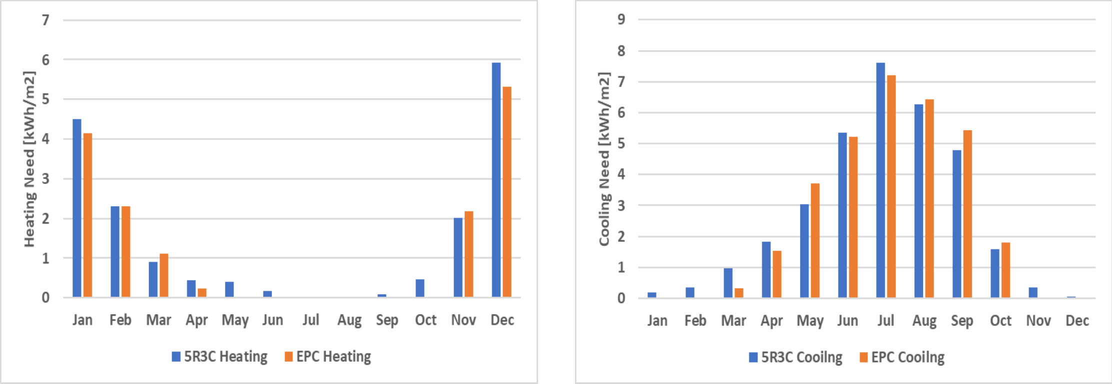
(a) (b)

Figure 68. T03 (a) Heating Need (yearly difference : 11.6%) (b) Cooling Need (Yearly difference : 1.6%)



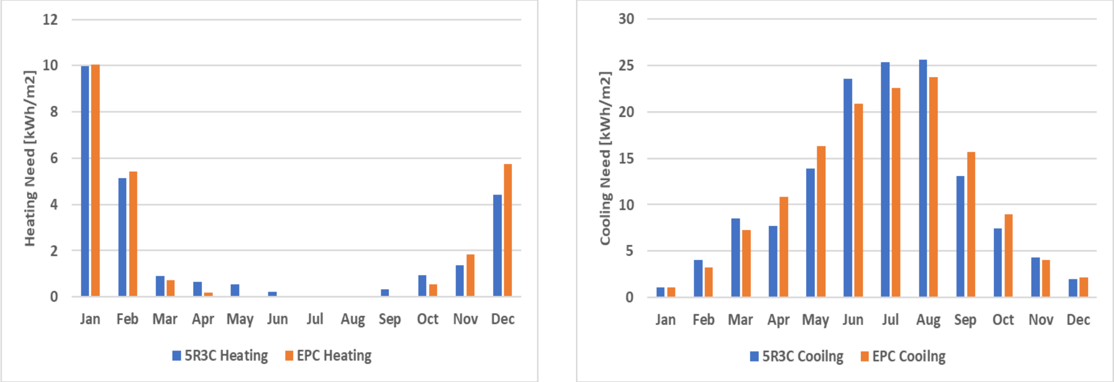
(a) (b)

Figure 69. T04 (a) Heating Need (yearly difference : 12.0%) (b) Cooling Need (Yearly difference : 1.9%)



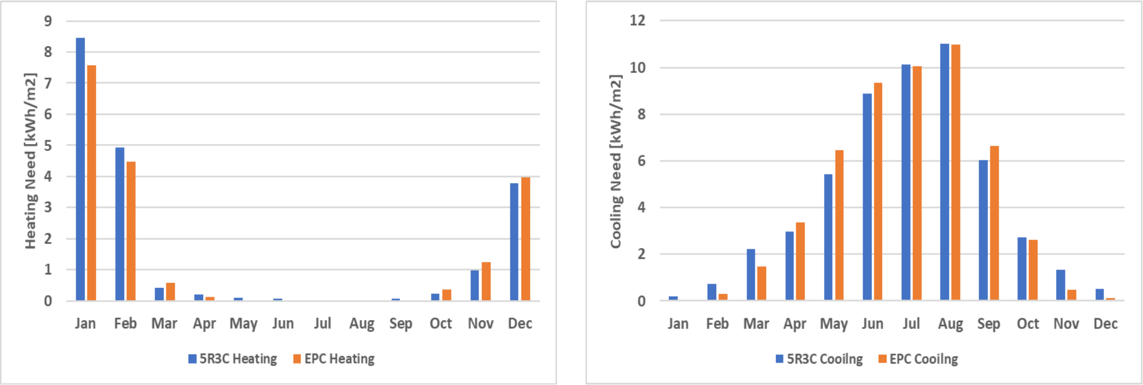
(a) (b)

Figure 70. T05 (a) Heating Need (yearly difference : 0.0%) (b) Cooling Need (Yearly difference : 0.0%)



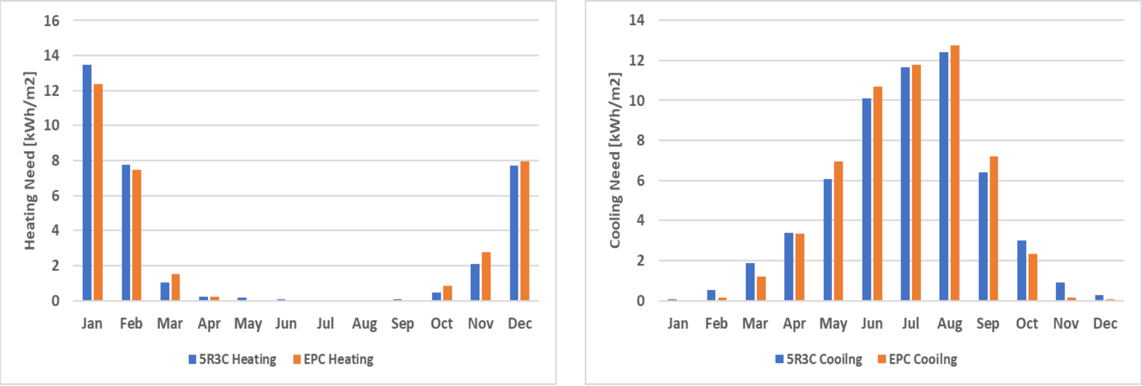
(a) (b)

Figure 71. Caddell (a) Heating Need (yearly difference : 0.0%) (b) Cooling Need (Yearly difference : 0.0%)



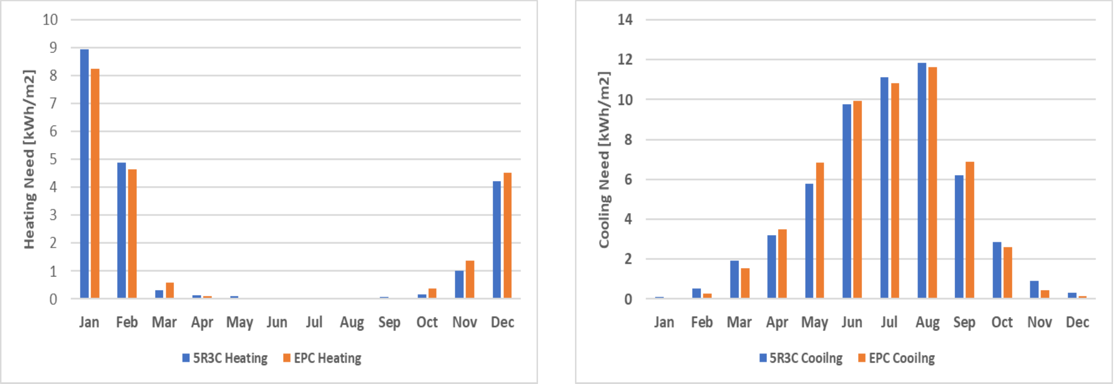
(a) (b)

Figure 72. Mason (a) Heating Need (yearly difference : 5.3%) (b) Cooling Need (Yearly difference : 0.7%)



(a) (b)

Figure 73. Smithgall (a) Heating Need (yearly difference : 0.0%) (b) Cooling Need (Yearly difference : 0.0%)



(a) (b)

Figure 74. Savant (a) Heating Need (yearly difference : 0.0%) (b) Cooling Need (Yearly difference : 0.0%)

APPENDIX B. COMPARISON BETWEEN THE ORIGINAL AND LINEARIZED SOLAR MODELS

APPENDIX B aims to show the linearization process of the clear-sky model (ASHRAE 2017) and compare the results between the original and linearized models.

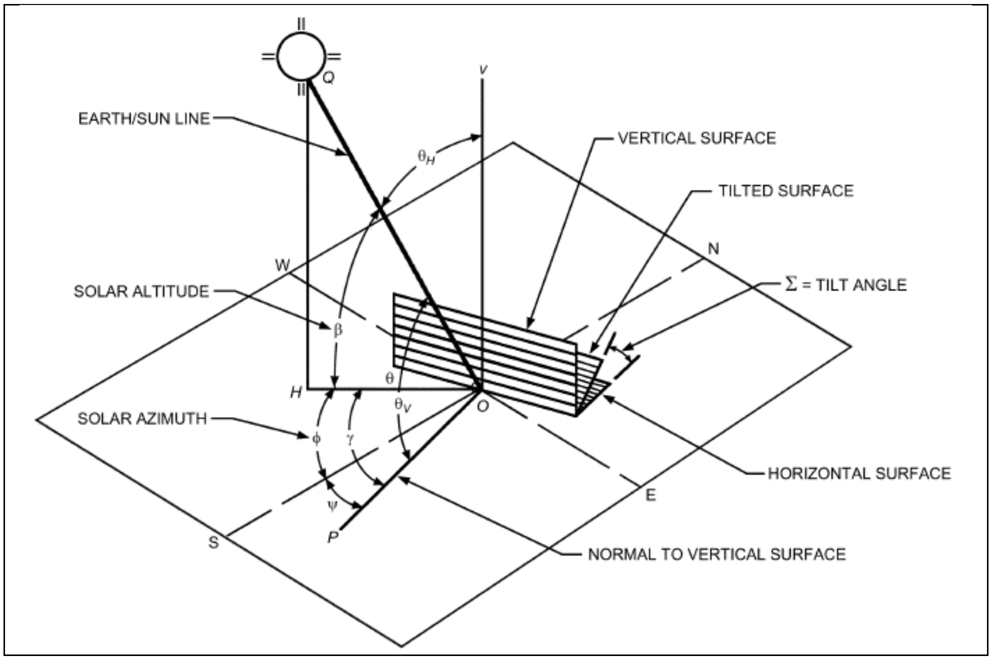


Figure 75. Solar Angles for Vertical and Horizontal Surfaces (ASHRAE 2017)

$E_t = E_{t,b} + E_{t,d} + E_{t,r}$

$E_t = E_b \cdot \cos\theta + E_d(\Upsilon \sin\Sigma + \cos\Sigma) + (E_b \cdot \sin\beta + E_d) \cdot \rho_g \cdot \frac{1-cos\Sigma}{2}$

$\cos\theta = \cos\beta\cos(\varphi - \psi)\sin\Sigma + \sin\beta\cos\Sigma$

Using the $\cos(\alpha - \beta) = \cos(\alpha)\cos(\beta) + \sin(\alpha)\sin(\beta)$ relationship, the $\cos(\theta)$ equation can be converted as shown below. **(19)**

$\cos\theta = \cos\beta\{\cos(\varphi)\cos(\psi) + sin(\varphi)sin(\psi)\}\sin\Sigma + \sin\beta\cos\Sigma$

$= \cos\beta\cos(\varphi)\cos(\psi)\sin\Sigma + \cos\beta sin(\varphi)sin(\psi)\sin\Sigma + \sin\beta\cos\Sigma$

$E_t = E_b \cdot (\cos\beta\cos(\varphi)\cos(\psi)\sin\Sigma + \cos\beta sin(\varphi)sin(\psi)\sin\Sigma + \sin\beta\cos\Sigma) + E_d(\Upsilon\sin\Sigma + \cos\Sigma) + (E_b \cdot \sin\beta + E_d)\rho_g \cdot \frac{1-cos\Sigma}{2}$

$$= E_b\cos\beta\cos(\varphi)\cos(\psi)\sin\Sigma + E_b\cos\beta\, sin(\varphi) sin(\psi)\sin\Sigma + E_b\cdot\sin\beta\cos\Sigma + E_d(\Upsilon\sin\Sigma + \cos\Sigma) + (E_b\cdot\sin\beta + E_d)\rho_g\cdot\frac{1-cos\Sigma}{2}$$

$$= E_b\cos\beta\cos(\varphi)\cos(\psi)\sin(\Sigma) + E_b\cos\beta\sin(\varphi)\sin(\psi)\sin(\Sigma) + E_b\sin\beta\cos(\Sigma) + E_d\Upsilon\sin(\Sigma) + E_d\cos(\Sigma) + \frac{1}{2}\rho_g(E_b\sin\beta+E_d) - \frac{1}{2}\rho_g cos\Sigma\,(E_b\sin\beta+E_d)$$

$(\Upsilon = \max(0.45, 0.55 + 0.437\cos(\theta) + 0.313\cos^2(\theta))$

Where,

E_t: Clear-sky irradiance, [W/m^2]

$E_{t,b}$: Beam component originating from the solar disc, [W/m^2]

$E_{t,d}$: Diffuse component originating from the solar disc, [W/m^2]

$E_{t,r}$: Ground-reflected component originating from the ground in front of the receiving surface, [W/m^2]

E_b: Beam normal irradiance (measured perpendicularly to rays of the sum), [W/m^2]

E_d: Diffuse horizontal irradiance (measured on horizontal surface), [W/m^2]

θ: The angle of incidence, the angle between the line normal to the irradiated surface and the earth-sun line, [°]

Σ: Surface slope, [°]

β: Solar altitude angle, [°]

ρ_g: Ground reflectance of foreground surfaces, [-]

φ: Solar azimuth, [°]

ψ: Surface azimuth, , [°]

Υ: A factor for a vertical surface having the same azimuth as the receiving surface considered, [-]

The linearization process is shown in Equation (20).

$$f(\psi,\Sigma) = E_b\cos\beta\cos(\varphi)\cos(\psi)\sin(\Sigma) + E_b\cos\beta\, sin(\varphi) sin(\psi)\sin(\Sigma) + E_b\sin\beta\cos(\Sigma) + E_d\Upsilon\sin(\Sigma) + E_d\cos(\Sigma) + \frac{1}{2}\rho_g(E_b\sin\beta+E_d) - \frac{1}{2}\rho_g cos\Sigma\,(E_b\sin\beta+E_d) \tag{20}$$

$$\frac{df(\psi,\Sigma)}{d\psi} = -E_b\cos\beta\cos(\varphi)\sin(\psi)\sin(\Sigma) + E_b\cos\beta\, sin(\varphi)\cos(\psi)\sin(\Sigma)$$

$$\frac{df(\psi,\Sigma)}{d\Sigma} = E_b\cos\beta\cos(\varphi)\cos(\psi)\cos(\Sigma) + E_b\cos\beta\, sin(\varphi) sin(\psi)\cos(\Sigma) - E_b\sin\beta\sin(\Sigma)$$
$$+ E_d\Upsilon\cos(\Sigma) - E_d\sin(\Sigma) + \frac{1}{2}\rho_g sin\Sigma\, (E_b\sin\beta+E_d)$$

Linearized clear-sky model:

$$f_{\text{linearized}}(\psi,\Sigma) = f(\psi_o,\Sigma_o) + \frac{df(\psi,\Sigma)}{d\psi}(\psi - \psi_o) + \frac{df(\psi,\Sigma)}{d\Sigma}(\Sigma - \Sigma_o)$$

$$= E_b\cos\beta\cos(\varphi)\cos(\psi_o)\sin(\Sigma_o) + E_b\cos\beta\, sin(\varphi) sin(\psi_o)\sin(\Sigma_o) + E_b\sin\beta\cos(\Sigma_o)$$
$$+ E_d\Upsilon\sin(\Sigma_o) + E_d\cos(\Sigma_o) + \frac{1}{2}\rho_g(E_b\sin\beta+E_d) - \frac{1}{2}\rho_g cos\Sigma_o(E_b\sin\beta+E_d)$$

$$- \{E_b\cos\beta\cos(\varphi)\sin(\psi_o)\sin(\Sigma_o) - E_b\cos\beta\, sin(\varphi)\cos(\psi_o)\sin(\Sigma_o)\}\cdot(\psi - \psi_o)$$

$$+ \{E_b\cos\beta\cos(\varphi)\cos(\psi_o)\cos(\Sigma_o) + E_b\cos\beta\, sin(\varphi) sin(\psi_o)\cos(\Sigma_o) - E_b\sin\beta\sin(\Sigma_o)$$
$$+ E_d\Upsilon\cos(\Sigma_o) - E_d\sin(\Sigma_o) + \frac{1}{2}\rho_g sin\Sigma_o(E_b\sin\beta+E_d)\}\cdot(\Sigma - \Sigma_o)$$

B.1 Result Comparison between the Original and Linearized Models

The original and linearized PV models are compared to each other to verify the two models' results. We test the models in Atlanta's climate with a 1.6 m^2 PV model, which is the size of one typical PV model. We select the PV module tilt angle as a manipulated variable. Other input parameters are fixed, as shown in Table 36.

Table 36. PV Model Test Setup

Name	Input
Weather File	USA_GA_Atlanta-Hartsfield-Jackson.Intl.AP.722190_TMY3.epw
PV Module Area	1.6 m^2
PV Module Azimuth	South
Peak Power Coefficient	250 W/m^2
Performance Factor	0.7

Delta Cost*	1
Tilt Angle maximum change*	5°

(*)- Additional constraints are used to prevent the non-necessary tilt angle changes.

$$\text{Max } \sum_{t=1}^{8760} PV\ output$$

s.t.

$$\text{PV output} = \text{Equation (19)}\ ,\ \forall t \in [1,8760]$$
$$\text{PV tilt angle} \in [\text{tilt angle min, tilt angle max}],\ \forall t \in [1,8760] \tag{21}$$

$$\text{Max } \sum_{t=1}^{8760} PV\ output$$

s.t.

$$\text{PV output} = \text{Equation (20)}\ ,\ \forall t \in [1,8760]$$
$$\text{PV tilt angle} \in [\text{tilt angle min, tilt angle max}]\ ,\ \forall t \in [1,8760] \tag{22}$$

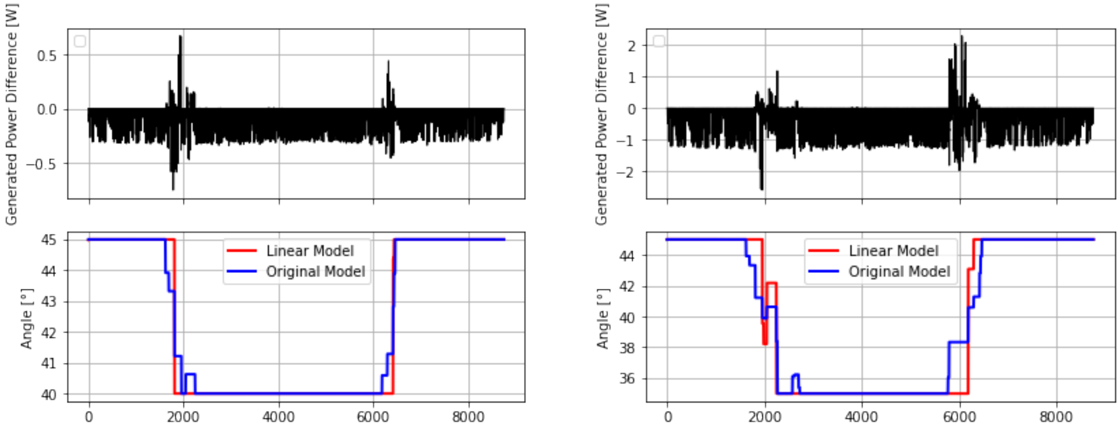
The Coefficient of Variance of Root Mean Squared Error (CVRMSE) is used as the performance indicator for the PV model comparison, and the CVRMSE equation is shown in Equation (23).

$$\text{CVRMSE} = \frac{1}{\bar{Y}}\sqrt{\frac{\sum_{i=1}^{N}(Y_i-\hat{Y}_i)^2}{N}} \tag{23}$$

Where,

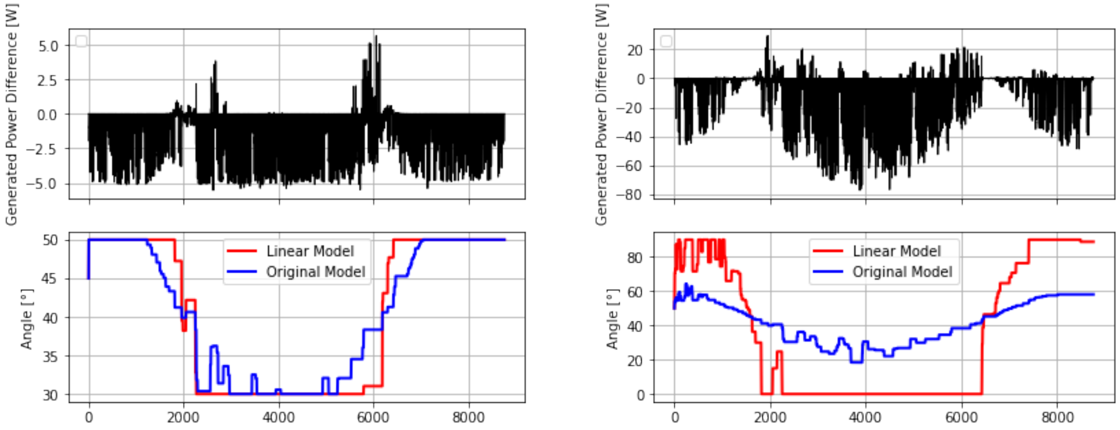
$\hat{Y}$: Predicted value
$\bar{Y}$: Mean of the observed data
Y: Observed data
N: Number of data points

From Figure 76, we find that as the PV slope angle interval increases, the tilt angle CVRMSE values also increase. This is because the linearized model is getting lost in the dynamics as the tilt angle interval increases. Furthermore, the generated power difference between the original and linearized models surges as the tilt angle interval setup increases. This can potentially lead to inaccurate decisions, especially when optimal PV module area with optimal PV module tilt angle problems is solved. Therefore, MINLP is employed to guarantee the confidence of the solutions in this thesis.



(a) Slope angle interval [40°, 45°]
Tilt angle CVRMSE: 0.002

(b) Slope angle interval [35°, 45°]
Tilt angle CVRMSE: 0.01



(c) Slope angle interval [30°, 50°]
Tilt angle CVRMSE: 0.02

(d) Slope angle interval [0°, 90°]
Tilt angle CVRMSE: 0.22

Figure 76. Results Comparison between the Original and Linearized Models (Generated power difference = generated electricity from the original model– generated electricity from the linearized model)

APPENDIX C. RBC LOGICS IN ENERGY SETTINGS 2

Table 37. RBC Logic Inputs in Setting 2

	RBC Logic Input
(a)	**if** 3623 < `CumulativeHour` < 6552 and 12 <= `Hour` <= 18 and `DayType` == 1 and Caddell `ElectricityUse` > 35 Caddell `CoolingSetPoint` = 27; **else** Caddell `CoolingSetPoint` = Original; **if** 3623 < `CumulativeHour` < 6552 and 12 <= `Hour` <= 18 and `DayType` == 1 and Mason `ElectricityUse` > 109 Mason `CoolingSetPoint` = 27; **else** Mason `CoolingSetPoint` = Original; **if** 3623 < `CumulativeHour` < 6552 and 12 <= `Hour` <= 18 and `DayType` == 1 and Chapin `ElectricityUse` > 21 Chapin `CoolingSetPoint` = 27; **else** Chapin `CoolingSetPoint` = Original; **if** 3623 < `CumulativeHour` < 6552 and 12 <= `Hour` <= 18 and `DayType` == 1 and Crecine ElectricityUse > 255 Crecine `CoolingSetPoint` = 27; **else** Crecine `CoolingSetPoint` = Original; **if** 3623 < `CumulativeHour` < 6552 and 12 <= `Hour` <= 18 and `DayType` == 1 and Ford `ElectricityUse` > 1074 Ford `CoolingSetPoint` = 27; **else** Ford `CoolingSetPoint` = Original; **if** 3623 < `CumulativeHour` < 6552 and 12 <= `Hour` <= 18 and `DayType` == 1 and MSE `ElectricityUse` > 405 MSE `CoolingSetPoint` = 27; **else** MSE `CoolingSetPoint` = Original; **if** 3623 < `CumulativeHour` < 6552 and 12 <= `Hour` <= 18 and `DayType` == 1 and Klaus `ElectricityUse` > 775 Klaus `CoolingSetPoint` = 27; **else**

```
    Klaus CoolingSetPoint = Original;
if 3623 < CumulativeHour < 6552 and 12 <= Hour <= 18 and DayType == 1 and Savant ElectricityUse > 48
    Savant CoolingSetPoint = 27;
else
    Savant CoolingSetPoint = Original;
if 3623 < CumulativeHour < 6552 and 12 <= Hour <= 18 and DayType == 1 and Smithgall ElectricityUse > 79
    Smithgall CoolingSetPoint = 27;
else
    Smithgall CoolingSetPoint = Original;
if 3623 < CumulativeHour < 6552 and 12 <= Hour <= 18 and DayType == 1 and AllenSustain ElectricityUse > 62
    AllenSustain CoolingSetPoint = 27;
else
    AllenSustain CoolingSetPoint = Original;
```

(b)

```
if 3623 < CumulativeHour < 6552 and 12 <= Hour <= 18 and DayType == 1 and DryBulbTemperature > 27
    Caddell CoolingSetPoint = 27; Mason CoolingSetPoint = 27;
    AllenSustain CoolingSetPoint = 27; Chapin CoolingSetPoint = 27;
    Crecine CoolingSetPoint = 27; MSE CoolingSetPoint = 27;
    Klaus CoolingSetPoint = 27; Savant CoolingSetPoint = 27;
    Smithgall CoolingSetPoint = 27; Ford CoolingSetPoint = 27;

else
    Caddell CoolingSetPoint = Original; Mason CoolingSetPoint = Original;
    AllenSustain CoolingSetPoint = Original; Chapin CoolingSetPoint = Original;
    Crecine CoolingSetPoint = Original; MSE CoolingSetPoint = Original;
    Klaus CoolingSetPoint = Original; Savant CoolingSetPoint = Original;
    Smithgall CoolingSetPoint = Original; Ford CoolingSetPoint = Original;
```

APPENDIX D. SOURCE CODE OF THE UP SCHEMA AND UP INSTANCES

D.1 Source Code of the UP Schema

```xml
<?xml version="1.0" encoding="utf-8" ?>
<!--Created with Liquid Studio 2019 (https://www.liquid-technologies.com)-->
<!--Created with Liquid Studio 2021 by Yun Joon Jung -->
<xs:schema elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
   <xs:simpleType name="Slope">
      <xs:restriction base="xs:integer">
         <xs:enumeration value="0" />
         <xs:enumeration value="30" />
         <xs:enumeration value="45" />
         <xs:enumeration value="60" />
         <xs:enumeration value="90" />
      </xs:restriction>
   </xs:simpleType>
   <xs:simpleType name="Orientation">
      <xs:restriction base="xs:string">
         <xs:enumeration value="S" />
         <xs:enumeration value="SE" />
         <xs:enumeration value="E" />
         <xs:enumeration value="W" />
         <xs:enumeration value="SW" />
      </xs:restriction>
   </xs:simpleType>
   <xs:simpleType name="OverhangFinAngles">
      <xs:restriction base="xs:integer">
         <xs:enumeration value="0" />
         <xs:enumeration value="30" />
         <xs:enumeration value="45" />
         <xs:enumeration value="60" />
      </xs:restriction>
   </xs:simpleType>
   <xs:complexType name="PhysicalLayerRelationship">
      <xs:sequence>
         <xs:element name="EnergySender">
            <xs:complexType>
               <xs:attribute name="KeyRef" type="xs:string" />
            </xs:complexType>
         </xs:element>
         <xs:element name="EnergyReceiver">
            <xs:complexType>
               <xs:attribute name="KeyRef" type="xs:string" />
```

```
            </xs:complexType>
         </xs:element>
      </xs:sequence>
   </xs:complexType>
   <xs:complexType name="LogicalLayerRelationship">
      <xs:sequence>
         <xs:element name="DataSender">
            <xs:complexType>
               <xs:attribute name="KeyRef" type="xs:string" />
            </xs:complexType>
         </xs:element>
         <xs:element name="DataReceiver">
            <xs:complexType>
               <xs:attribute name="KeyRef" type="xs:string" />
            </xs:complexType>
         </xs:element>
      </xs:sequence>
   </xs:complexType>
   <xs:complexType name="DO_Value_Cost">
      <xs:sequence>
         <xs:element name="Value" type="xs:float" />
         <xs:element name="Cost" type="xs:float" />
      </xs:sequence>
   </xs:complexType>
   <xs:complexType name="DOInput">
      <xs:sequence>
         <xs:element name="DataType">
            <xs:simpleType>
               <xs:restriction base="xs:string">
                  <xs:enumeration value="ContinuousReal" />
                  <xs:enumeration value="ContinuousInteger" />
                  <xs:enumeration value="Discontinuous" />
               </xs:restriction>
            </xs:simpleType>
         </xs:element>
         <xs:element name="TechnologyLevel1" type="DO_Value_Cost" />
         <xs:element name="TechnologyLevel2" type="DO_Value_Cost" />
         <xs:element name="TechnologyLevel3" type="DO_Value_Cost" minOccurs="0" />
         <xs:element name="TechnologyLevel4" type="DO_Value_Cost" minOccurs="0" />
         <xs:element name="TechnologyLevel5" type="DO_Value_Cost" minOccurs="0" />
         <xs:element name="TechnologyLevel6" type="DO_Value_Cost" minOccurs="0" />
      </xs:sequence>
   </xs:complexType>
   <xs:complexType name="OptimalControlConstraintInput">
      <xs:sequence>
         <xs:element name="Min" type="xs:float" />
         <xs:element name="MinCost" type="xs:float" />
         <xs:element name="Max" type="xs:float" />
         <xs:element name="MaxCost" type="xs:float" />
         <xs:element name="Integer">
            <xs:simpleType>
```

```
          <xs:restriction base="xs:string">
            <xs:enumeration value="Yes" />
            <xs:enumeration value="No" />
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="DCOST" type="xs:float" minOccurs="0" />
      <xs:element name="DMAX" type="xs:float" minOccurs="0" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="UPSchema">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="PhysicalLayer">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Consumer">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="Building" minOccurs="0"
maxOccurs="unbounded">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element name="Name" type="xs:string" />
                        </xs:sequence>
                        <xs:attribute name="Key">
                          <xs:simpleType>
                            <xs:restriction base="xs:string">
                              <xs:pattern value="[B][0-9][0-9][0-9][0-9]" />
                            </xs:restriction>
                          </xs:simpleType>
                        </xs:attribute>
                      </xs:complexType>
                    </xs:element>
                    <xs:element name="ElectricVehicleChargingStation" minOccurs="0"
maxOccurs="unbounded">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element name="Name" type="xs:string" />
                        </xs:sequence>
                        <xs:attribute name="Key">
                          <xs:simpleType>
                            <xs:restriction base="xs:string">
                              <xs:pattern value="[E][V][0-9][0-9][0-9][0-9]" />
                            </xs:restriction>
                          </xs:simpleType>
                        </xs:attribute>
                      </xs:complexType>
                    </xs:element>
                  </xs:sequence>
                </xs:complexType>
```

```
              </xs:element>
              <xs:element name="Producer">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="PV" minOccurs="0" maxOccurs="unbounded">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element name="Name" type="xs:string" />
                        </xs:sequence>
                        <xs:attribute name="Key">
                          <xs:simpleType>
                            <xs:restriction base="xs:string">
                              <xs:pattern value="[P][0-9][0-9][0-9][0-9]" />
                            </xs:restriction>
                          </xs:simpleType>
                        </xs:attribute>
                      </xs:complexType>
                    </xs:element>
                    <xs:element name="WindTurbine" minOccurs="0"
maxOccurs="unbounded">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element name="Name" type="xs:string" />
                        </xs:sequence>
                        <xs:attribute name="Key">
                          <xs:simpleType>
                            <xs:restriction base="xs:string">
                              <xs:pattern value="[W][0-9][0-9][0-9][0-9]" />
                            </xs:restriction>
                          </xs:simpleType>
                        </xs:attribute>
                      </xs:complexType>
                    </xs:element>
                    <xs:element name="SolarHotWater" minOccurs="0"
maxOccurs="unbounded">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element name="Name" type="xs:string" />
                        </xs:sequence>
                        <xs:attribute name="Key">
                          <xs:simpleType>
                            <xs:restriction base="xs:string">
                              <xs:pattern value="[S][0-9][0-9][0-9][0-9]" />
                            </xs:restriction>
                          </xs:simpleType>
                        </xs:attribute>
                      </xs:complexType>
                    </xs:element>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
```

```
                     <xs:element name="DistrictSystem">
                        <xs:complexType>
                           <xs:sequence>
                              <xs:element name="DistrictHeating" minOccurs="0"
maxOccurs="unbounded">
                                 <xs:complexType>
                                    <xs:sequence>
                                       <xs:element name="Name" type="xs:string" />
                                    </xs:sequence>
                                    <xs:attribute name="Key">
                                       <xs:simpleType>
                                          <xs:restriction base="xs:string">
                                             <xs:pattern value="[D][H][0-9][0-9][0-9][0-9]" />
                                          </xs:restriction>
                                       </xs:simpleType>
                                    </xs:attribute>
                                 </xs:complexType>
                              </xs:element>
                              <xs:element name="DistrictCooling" minOccurs="0"
maxOccurs="unbounded">
                                 <xs:complexType>
                                    <xs:sequence>
                                       <xs:element name="Name" type="xs:string" />
                                    </xs:sequence>
                                    <xs:attribute name="Key">
                                       <xs:simpleType>
                                          <xs:restriction base="xs:string">
                                             <xs:pattern value="[D][C][0-9][0-9][0-9][0-9]" />
                                          </xs:restriction>
                                       </xs:simpleType>
                                    </xs:attribute>
                                 </xs:complexType>
                              </xs:element>
                              <xs:element name="Cogeneration" minOccurs="0"
maxOccurs="unbounded">
                                 <xs:complexType>
                                    <xs:sequence>
                                       <xs:element name="Name" type="xs:string" />
                                    </xs:sequence>
                                    <xs:attribute name="Key">
                                       <xs:simpleType>
                                          <xs:restriction base="xs:string">
                                             <xs:pattern value="[C][O][0-9][0-9][0-9][0-9]" />
                                          </xs:restriction>
                                       </xs:simpleType>
                                    </xs:attribute>
                                 </xs:complexType>
                              </xs:element>
                           </xs:sequence>
                        </xs:complexType>
                     </xs:element>
```

```
              <xs:element name="Storage">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="ElectricityBattery" minOccurs="0"
maxOccurs="unbounded">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element name="Name" type="xs:string" />
                        </xs:sequence>
                        <xs:attribute name="Key">
                          <xs:simpleType>
                            <xs:restriction base="xs:string">
                              <xs:pattern value="[E][B][0-9][0-9][0-9][0-9]" />
                            </xs:restriction>
                          </xs:simpleType>
                        </xs:attribute>
                      </xs:complexType>
                    </xs:element>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
              <xs:element name="Source">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="ElectricityGrid" minOccurs="0">
                      <xs:complexType>
                        <xs:attribute name="Key">
                          <xs:simpleType>
                            <xs:restriction base="xs:string">
                              <xs:pattern value="SE1000" />
                            </xs:restriction>
                          </xs:simpleType>
                        </xs:attribute>
                      </xs:complexType>
                    </xs:element>
                    <xs:element name="NaturalGas" minOccurs="0">
                      <xs:complexType>
                        <xs:attribute name="Key" fixed="SN1000">
                          <xs:simpleType>
                            <xs:restriction base="xs:string" />
                          </xs:simpleType>
                        </xs:attribute>
                      </xs:complexType>
                    </xs:element>
                    <xs:element name="Gasoline" minOccurs="0">
                      <xs:complexType>
                        <xs:attribute name="Key" fixed="SG1000">
                          <xs:simpleType>
                            <xs:restriction base="xs:string" />
                          </xs:simpleType>
                        </xs:attribute>
```

```
                                 </xs:complexType>
                              </xs:element>
                              <xs:element name="Diesel" minOccurs="0">
                                 <xs:complexType>
                                    <xs:attribute name="Key" fixed="SD1000">
                                       <xs:simpleType>
                                          <xs:restriction base="xs:string" />
                                       </xs:simpleType>
                                    </xs:attribute>
                                 </xs:complexType>
                              </xs:element>
                              <xs:element name="Propane" minOccurs="0">
                                 <xs:complexType>
                                    <xs:attribute name="Key" fixed="SP1000">
                                       <xs:simpleType>
                                          <xs:restriction base="xs:string" />
                                       </xs:simpleType>
                                    </xs:attribute>
                                 </xs:complexType>
                              </xs:element>
                              <xs:element name="Oil" minOccurs="0">
                                 <xs:complexType>
                                    <xs:attribute name="Key" fixed="SO1000">
                                       <xs:simpleType>
                                          <xs:restriction base="xs:string" />
                                       </xs:simpleType>
                                    </xs:attribute>
                                 </xs:complexType>
                              </xs:element>
                           </xs:sequence>
                        </xs:complexType>
                     </xs:element>
                     <xs:element name="PhysicalLayerRelationship">
                        <xs:complexType>
                           <xs:sequence>
                              <xs:element name="SupplyPower" type="PhysicalLayerRelationship"
minOccurs="0" maxOccurs="unbounded" />
                              <xs:element name="SellBackPower" type="PhysicalLayerRelationship"
minOccurs="0" maxOccurs="unbounded" />
                              <xs:element name="SupplyDomesticHotWater"
type="PhysicalLayerRelationship" minOccurs="0" maxOccurs="unbounded" />
                              <xs:element name="SupplyChilledWater"
type="PhysicalLayerRelationship" minOccurs="0" maxOccurs="unbounded" />
                              <xs:element name="SupplyHotWater"
type="PhysicalLayerRelationship" minOccurs="0" maxOccurs="unbounded" />
                              <xs:element name="SupplySteam" type="PhysicalLayerRelationship"
minOccurs="0" maxOccurs="unbounded" />
                              <xs:element name="SupplyNaturalGas"
type="PhysicalLayerRelationship" minOccurs="0" maxOccurs="unbounded" />
                              <xs:element name="SupplyGasoline" type="PhysicalLayerRelationship"
minOccurs="0" maxOccurs="unbounded" />
```

```
                              <xs:element name="SupplyDiesel" type="PhysicalLayerRelationship"
minOccurs="0" maxOccurs="unbounded" />
                              <xs:element name="SupplyPropane" type="PhysicalLayerRelationship"
minOccurs="0" maxOccurs="unbounded" />
                              <xs:element name="SupplyOil" type="PhysicalLayerRelationship"
minOccurs="0" maxOccurs="unbounded" />
                           </xs:sequence>
                        </xs:complexType>
                     </xs:element>
                  </xs:sequence>
               </xs:complexType>
            </xs:element>
            <xs:element name="LogicalLayer">
               <xs:complexType>
                  <xs:sequence>
                     <xs:element name="Sensor">
                        <xs:complexType>
                           <xs:sequence>
                              <xs:element name="Time">
                                 <xs:complexType>
                                    <xs:sequence>
                                       <xs:element name="CumulativeHour" minOccurs="0">
                                          <xs:simpleType>
                                             <xs:restriction base="xs:integer">
                                                <xs:minInclusive value="1" />
                                                <xs:maxInclusive value="8760" />
                                             </xs:restriction>
                                          </xs:simpleType>
                                       </xs:element>
                                       <xs:element name="Hour" minOccurs="0">
                                          <xs:simpleType>
                                             <xs:restriction base="xs:integer">
                                                <xs:minInclusive value="1" />
                                                <xs:maxInclusive value="24" />
                                             </xs:restriction>
                                          </xs:simpleType>
                                       </xs:element>
                                       <xs:element name="Date" minOccurs="0">
                                          <xs:simpleType>
                                             <xs:restriction base="xs:integer">
                                                <xs:minInclusive value="1" />
                                                <xs:maxInclusive value="365" />
                                             </xs:restriction>
                                          </xs:simpleType>
                                       </xs:element>
                                       <xs:element name="Day" minOccurs="0">
                                          <xs:simpleType>
                                             <xs:restriction base="xs:integer">
                                                <xs:minInclusive value="1" />
                                                <xs:maxInclusive value="7" />
                                             </xs:restriction>
```

```
                            </xs:simpleType>
                          </xs:element>
                          <xs:element name="DayType" minOccurs="0">
                            <xs:simpleType>
                              <xs:restriction base="xs:integer">
                                <xs:minInclusive value="0" />
                                <xs:maxInclusive value="1" />
                              </xs:restriction>
                            </xs:simpleType>
                          </xs:element>
                        </xs:sequence>
                        <xs:attribute name="ID" fixed="L1100" type="xs:string" />
                      </xs:complexType>
                    </xs:element>
                    <xs:element name="PriceSignal">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element name="ElectricityPrice" minOccurs="0">
                            <xs:complexType>
                              <xs:simpleContent>
                                <xs:extension base="xs:float">
                                  <xs:attribute name="ID" fixed="LP1000"
type="xs:string" />
                                  <xs:attribute name="TimeStep">
                                    <xs:simpleType>
                                      <xs:restriction base="xs:integer">
                                        <xs:minInclusive value="1" />
                                        <xs:maxInclusive value="8760" />
                                      </xs:restriction>
                                    </xs:simpleType>
                                  </xs:attribute>
                                </xs:extension>
                              </xs:simpleContent>
                            </xs:complexType>
                          </xs:element>
                          <xs:element name="NaturalGasPrice" minOccurs="0">
                            <xs:complexType>
                              <xs:simpleContent>
                                <xs:extension base="xs:float">
                                  <xs:attribute name="ID" fixed="LP1100"
type="xs:string" />
                                  <xs:attribute name="TimeStep">
                                    <xs:simpleType>
                                      <xs:restriction base="xs:integer">
                                        <xs:minInclusive value="1" />
                                        <xs:maxInclusive value="8760" />
                                      </xs:restriction>
                                    </xs:simpleType>
                                  </xs:attribute>
                                </xs:extension>
                              </xs:simpleContent>
```

```
   </xs:complexType>
</xs:element>
<xs:element name="GasolinePrice" minOccurs="0">
   <xs:complexType>
      <xs:simpleContent>
         <xs:extension base="xs:float">
            <xs:attribute name="ID" fixed="LP1200"
type="xs:string" />

            <xs:attribute name="TimeStep">
               <xs:simpleType>
                  <xs:restriction base="xs:integer">
                     <xs:minInclusive value="1" />
                     <xs:maxInclusive value="8760" />
                  </xs:restriction>
               </xs:simpleType>
            </xs:attribute>
         </xs:extension>
      </xs:simpleContent>
   </xs:complexType>
</xs:element>
<xs:element name="DieselPrice" minOccurs="0">
   <xs:complexType>
      <xs:simpleContent>
         <xs:extension base="xs:float">
            <xs:attribute name="ID" fixed="LP1300"
type="xs:string" />

            <xs:attribute name="TimeStep">
               <xs:simpleType>
                  <xs:restriction base="xs:integer">
                     <xs:minInclusive value="1" />
                     <xs:maxInclusive value="8760" />
                  </xs:restriction>
               </xs:simpleType>
            </xs:attribute>
         </xs:extension>
      </xs:simpleContent>
   </xs:complexType>
</xs:element>
<xs:element name="PropanePrice" minOccurs="0">
   <xs:complexType>
      <xs:simpleContent>
         <xs:extension base="xs:float">
            <xs:attribute name="ID" fixed="LP1400"
type="xs:string" />

            <xs:attribute name="TimeStep">
               <xs:simpleType>
                  <xs:restriction base="xs:integer">
                     <xs:minInclusive value="1" />
                     <xs:maxInclusive value="8760" />
                  </xs:restriction>
               </xs:simpleType>
```

```
                                  </xs:attribute>
                                </xs:extension>
                              </xs:simpleContent>
                            </xs:complexType>
                          </xs:element>
                          <xs:element name="OilPrice" minOccurs="0">
                            <xs:complexType>
                              <xs:simpleContent>
                                <xs:extension base="xs:float">
                                  <xs:attribute name="ID" fixed="LP1500"
type="xs:string" />

                                  <xs:attribute name="TimeStep">
                                    <xs:simpleType>
                                      <xs:restriction base="xs:integer">
                                        <xs:minInclusive value="1" />
                                        <xs:maxInclusive value="8760" />
                                      </xs:restriction>
                                    </xs:simpleType>
                                  </xs:attribute>
                                </xs:extension>
                              </xs:simpleContent>
                            </xs:complexType>
                          </xs:element>
                        </xs:sequence>
                        <xs:attribute name="ID" fixed="L1300" type="xs:string" />
                      </xs:complexType>
                    </xs:element>
                    <xs:element name="Weather">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element name="DryBulbTemperature" minOccurs="0">
                            <xs:complexType>
                              <xs:simpleContent>
                                <xs:extension base="xs:float">
                                  <xs:attribute name="ID" fixed="LW1000"
type="xs:string" />

                                  <xs:attribute name="TimeStep">
                                    <xs:simpleType>
                                      <xs:restriction base="xs:integer">
                                        <xs:minInclusive value="1" />
                                        <xs:maxInclusive value="8760" />
                                      </xs:restriction>
                                    </xs:simpleType>
                                  </xs:attribute>
                                </xs:extension>
                              </xs:simpleContent>
                            </xs:complexType>
                          </xs:element>
                          <xs:element name="WindSpeed" minOccurs="0">
                            <xs:complexType>
                              <xs:simpleContent>
```

```
                  <xs:extension base="xs:float">
                    <xs:attribute name="ID" fixed="LW1100"
type="xs:string" />
                    <xs:attribute name="TimeStep">
                      <xs:simpleType>
                        <xs:restriction base="xs:integer">
                          <xs:minInclusive value="1" />
                          <xs:maxInclusive value="8760" />
                        </xs:restriction>
                      </xs:simpleType>
                    </xs:attribute>
                  </xs:extension>
                </xs:simpleContent>
              </xs:complexType>
            </xs:element>
            <xs:element name="DirectNormalRadiation" minOccurs="0">
              <xs:complexType>
                <xs:simpleContent>
                  <xs:extension base="xs:float">
                    <xs:attribute name="ID" fixed="LW1200"
type="xs:string" />
                    <xs:attribute name="TimeStep">
                      <xs:simpleType>
                        <xs:restriction base="xs:integer">
                          <xs:minInclusive value="1" />
                          <xs:maxInclusive value="8760" />
                        </xs:restriction>
                      </xs:simpleType>
                    </xs:attribute>
                  </xs:extension>
                </xs:simpleContent>
              </xs:complexType>
            </xs:element>
            <xs:element name="DiffuseHorizontalRadiation"
minOccurs="0">
              <xs:complexType>
                <xs:simpleContent>
                  <xs:extension base="xs:float">
                    <xs:attribute name="ID" fixed="LW1300"
type="xs:string" />
                    <xs:attribute name="TimeStep">
                      <xs:simpleType>
                        <xs:restriction base="xs:integer">
                          <xs:minInclusive value="1" />
                          <xs:maxInclusive value="8760" />
                        </xs:restriction>
                      </xs:simpleType>
                    </xs:attribute>
                  </xs:extension>
                </xs:simpleContent>
              </xs:complexType>
```

```
                          </xs:element>
                          <xs:element name="HorizontalSolarRadiation" minOccurs="0">
                            <xs:complexType>
                              <xs:simpleContent>
                                <xs:extension base="xs:float">
                                  <xs:attribute name="ID" fixed="LW1400"
type="xs:string" />

                                  <xs:attribute name="TimeStep">
                                    <xs:simpleType>
                                      <xs:restriction base="xs:integer">
                                        <xs:minInclusive value="1" />
                                        <xs:maxInclusive value="8760" />
                                      </xs:restriction>
                                    </xs:simpleType>
                                  </xs:attribute>
                                </xs:extension>
                              </xs:simpleContent>
                            </xs:complexType>
                          </xs:element>
                          <xs:element name="SpecificHumidity" minOccurs="0">
                            <xs:complexType>
                              <xs:simpleContent>
                                <xs:extension base="xs:float">
                                  <xs:attribute name="ID" fixed="LW1500"
type="xs:string" />

                                  <xs:attribute name="TimeStep">
                                    <xs:simpleType>
                                      <xs:restriction base="xs:integer">
                                        <xs:minInclusive value="1" />
                                        <xs:maxInclusive value="8760" />
                                      </xs:restriction>
                                    </xs:simpleType>
                                  </xs:attribute>
                                </xs:extension>
                              </xs:simpleContent>
                            </xs:complexType>
                          </xs:element>
                          <xs:element name="AtmosphericPressure" minOccurs="0">
                            <xs:complexType>
                              <xs:simpleContent>
                                <xs:extension base="xs:float">
                                  <xs:attribute name="ID" fixed="LW1600"
type="xs:string" />

                                  <xs:attribute name="TimeStep">
                                    <xs:simpleType>
                                      <xs:restriction base="xs:integer">
                                        <xs:minInclusive value="1" />
                                        <xs:maxInclusive value="8760" />
                                      </xs:restriction>
                                    </xs:simpleType>
                                  </xs:attribute>
```

```
                                </xs:extension>
                              </xs:simpleContent>
                            </xs:complexType>
                          </xs:element>
                          <xs:element name="RelativeHumidity" minOccurs="0">
                            <xs:complexType>
                              <xs:simpleContent>
                                <xs:extension base="xs:float">
                                  <xs:attribute name="ID" fixed="LW1700"
type="xs:string" />
                                  <xs:attribute name="TimeStep">
                                    <xs:simpleType>
                                      <xs:restriction base="xs:integer">
                                        <xs:minInclusive value="1" />
                                        <xs:maxInclusive value="8760" />
                                      </xs:restriction>
                                    </xs:simpleType>
                                  </xs:attribute>
                                </xs:extension>
                              </xs:simpleContent>
                            </xs:complexType>
                          </xs:element>
                        </xs:sequence>
                        <xs:attribute name="ID" fixed="L1200" type="xs:string" />
                      </xs:complexType>
                    </xs:element>
                    <xs:element name="Consumer">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element name="Building" minOccurs="0"
maxOccurs="unbounded">
                            <xs:complexType>
                              <xs:sequence>
                                <xs:element name="Occupancy" minOccurs="0">
                                  <xs:complexType>
                                    <xs:simpleContent>
                                      <xs:extension base="xs:float">
                                        <xs:attribute name="Key">
                                          <xs:simpleType>
                                            <xs:restriction base="xs:string">
                                              <xs:pattern value="[B][S][I][0-9][0-
9][0-9][0-9]" />
                                            </xs:restriction>
                                          </xs:simpleType>
                                        </xs:attribute>
                                        <xs:attribute name="TimeStep">
                                          <xs:simpleType>
                                            <xs:restriction base="xs:integer">
                                              <xs:minInclusive value="1" />
                                              <xs:maxInclusive value="8760" />
                                            </xs:restriction>
```

```
                                          </xs:simpleType>
                                        </xs:attribute>
                                      </xs:extension>
                                    </xs:simpleContent>
                                  </xs:complexType>
                                </xs:element>
                                <xs:element name="ThermalHeatingNeed"
minOccurs="0">

                                  <xs:complexType>
                                    <xs:simpleContent>
                                      <xs:extension base="xs:float">
                                        <xs:attribute name="Key">
                                          <xs:simpleType>
                                            <xs:restriction base="xs:string">
                                              <xs:pattern value="[B][S][I][0-9][0-
9][0-9][0-9]" />

                                            </xs:restriction>
                                          </xs:simpleType>
                                        </xs:attribute>
                                        <xs:attribute name="TimeStep">
                                          <xs:simpleType>
                                            <xs:restriction base="xs:integer">
                                              <xs:minInclusive value="1" />
                                              <xs:maxInclusive value="8760" />
                                            </xs:restriction>
                                          </xs:simpleType>
                                        </xs:attribute>
                                      </xs:extension>
                                    </xs:simpleContent>
                                  </xs:complexType>
                                </xs:element>
                                <xs:element name="ThermalCoolingNeed"
minOccurs="0">

                                  <xs:complexType>
                                    <xs:simpleContent>
                                      <xs:extension base="xs:float">
                                        <xs:attribute name="Key">
                                          <xs:simpleType>
                                            <xs:restriction base="xs:string">
                                              <xs:pattern value="[B][S][I][0-9][0-
9][0-9][0-9]" />

                                            </xs:restriction>
                                          </xs:simpleType>
                                        </xs:attribute>
                                        <xs:attribute name="TimeStep">
                                          <xs:simpleType>
                                            <xs:restriction base="xs:integer">
                                              <xs:minInclusive value="1" />
                                              <xs:maxInclusive value="8760" />
                                            </xs:restriction>
                                          </xs:simpleType>
```

```
                        </xs:attribute>
                      </xs:extension>
                    </xs:simpleContent>
                  </xs:complexType>
                </xs:element>
                <xs:element name="ThermalZoneAirTemperature"
minOccurs="0">
                  <xs:complexType>
                    <xs:simpleContent>
                      <xs:extension base="xs:float">
                        <xs:attribute name="Key">
                          <xs:simpleType>
                            <xs:restriction base="xs:string">
                              <xs:pattern value="[B][S][I][0-9][0-
9][0-9][0-9]" />
                            </xs:restriction>
                          </xs:simpleType>
                        </xs:attribute>
                        <xs:attribute name="TimeStep">
                          <xs:simpleType>
                            <xs:restriction base="xs:integer">
                              <xs:minInclusive value="1" />
                              <xs:maxInclusive value="8760" />
                            </xs:restriction>
                          </xs:simpleType>
                        </xs:attribute>
                      </xs:extension>
                    </xs:simpleContent>
                  </xs:complexType>
                </xs:element>
                <xs:element name="DeliveredEnergyUse"
minOccurs="0">
                  <xs:complexType>
                    <xs:simpleContent>
                      <xs:extension base="xs:float">
                        <xs:attribute name="Key">
                          <xs:simpleType>
                            <xs:restriction base="xs:string">
                              <xs:pattern value="[B][S][I][0-9][0-
9][0-9][0-9]" />
                            </xs:restriction>
                          </xs:simpleType>
                        </xs:attribute>
                        <xs:attribute name="TimeStep">
                          <xs:simpleType>
                            <xs:restriction base="xs:integer">
                              <xs:minInclusive value="1" />
                              <xs:maxInclusive value="8760" />
                            </xs:restriction>
                          </xs:simpleType>
                        </xs:attribute>
```

```
         </xs:extension>
      </xs:simpleContent>
   </xs:complexType>
</xs:element>
<xs:element name="HeatingDeliveredEnergyUse"
minOccurs="0">

   <xs:complexType>
      <xs:simpleContent>
         <xs:extension base="xs:float">
            <xs:attribute name="Key">
               <xs:simpleType>
                  <xs:restriction base="xs:string">
                     <xs:pattern value="[B][S][I][0-9][0-
9][0-9][0-9]" />

                  </xs:restriction>
               </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="TimeStep">
               <xs:simpleType>
                  <xs:restriction base="xs:integer">
                     <xs:minInclusive value="1" />
                     <xs:maxInclusive value="8760" />
                  </xs:restriction>
               </xs:simpleType>
            </xs:attribute>
         </xs:extension>
      </xs:simpleContent>
   </xs:complexType>
</xs:element>
<xs:element name="CoolingDeliveredEnergyUse"
minOccurs="0">

   <xs:complexType>
      <xs:simpleContent>
         <xs:extension base="xs:float">
            <xs:attribute name="Key">
               <xs:simpleType>
                  <xs:restriction base="xs:string">
                     <xs:pattern value="[B][S][I][0-9][0-
9][0-9][0-9]" />

                  </xs:restriction>
               </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="TimeStep">
               <xs:simpleType>
                  <xs:restriction base="xs:integer">
                     <xs:minInclusive value="1" />
                     <xs:maxInclusive value="8760" />
                  </xs:restriction>
               </xs:simpleType>
            </xs:attribute>
         </xs:extension>
```

```
                                    </xs:simpleContent>
                                  </xs:complexType>
                                </xs:element>
                                <xs:element name="LightingDeliveredEnergyUse"
minOccurs="0">
                                  <xs:complexType>
                                    <xs:simpleContent>
                                      <xs:extension base="xs:float">
                                        <xs:attribute name="Key">
                                          <xs:simpleType>
                                            <xs:restriction base="xs:string">
                                              <xs:pattern value="[B][S][I][0-9][0-
9][0-9][0-9]" />
                                            </xs:restriction>
                                          </xs:simpleType>
                                        </xs:attribute>
                                        <xs:attribute name="TimeStep">
                                          <xs:simpleType>
                                            <xs:restriction base="xs:integer">
                                              <xs:minInclusive value="1" />
                                              <xs:maxInclusive value="8760" />
                                            </xs:restriction>
                                          </xs:simpleType>
                                        </xs:attribute>
                                      </xs:extension>
                                    </xs:simpleContent>
                                  </xs:complexType>
                                </xs:element>
                                <xs:element name="FanDeliveredEnergyUse"
minOccurs="0">
                                  <xs:complexType>
                                    <xs:simpleContent>
                                      <xs:extension base="xs:float">
                                        <xs:attribute name="Key">
                                          <xs:simpleType>
                                            <xs:restriction base="xs:string">
                                              <xs:pattern value="[B][S][I][0-9][0-
9][0-9][0-9]" />
                                            </xs:restriction>
                                          </xs:simpleType>
                                        </xs:attribute>
                                        <xs:attribute name="TimeStep">
                                          <xs:simpleType>
                                            <xs:restriction base="xs:integer">
                                              <xs:minInclusive value="1" />
                                              <xs:maxInclusive value="8760" />
                                            </xs:restriction>
                                          </xs:simpleType>
                                        </xs:attribute>
                                      </xs:extension>
                                    </xs:simpleContent>
```

```
                  </xs:complexType>
                </xs:element>
                <xs:element name="PumpDeliveredEnergyUse"
minOccurs="0">
                  <xs:complexType>
                    <xs:simpleContent>
                      <xs:extension base="xs:float">
                        <xs:attribute name="Key">
                          <xs:simpleType>
                            <xs:restriction base="xs:string">
                              <xs:pattern value="[B][S][I][0-9][0-
9][0-9][0-9]" />

                            </xs:restriction>
                          </xs:simpleType>
                        </xs:attribute>
                        <xs:attribute name="TimeStep">
                          <xs:simpleType>
                            <xs:restriction base="xs:integer">
                              <xs:minInclusive value="1" />
                              <xs:maxInclusive value="8760" />
                            </xs:restriction>
                          </xs:simpleType>
                        </xs:attribute>
                      </xs:extension>
                    </xs:simpleContent>
                  </xs:complexType>
                </xs:element>
                <xs:element name="ApplianceDeliveredEnergyUse"
minOccurs="0">
                  <xs:complexType>
                    <xs:simpleContent>
                      <xs:extension base="xs:float">
                        <xs:attribute name="Key">
                          <xs:simpleType>
                            <xs:restriction base="xs:string">
                              <xs:pattern value="[B][S][I][0-9][0-
9][0-9][0-9]" />

                            </xs:restriction>
                          </xs:simpleType>
                        </xs:attribute>
                        <xs:attribute name="TimeStep">
                          <xs:simpleType>
                            <xs:restriction base="xs:integer">
                              <xs:minInclusive value="1" />
                              <xs:maxInclusive value="8760" />
                            </xs:restriction>
                          </xs:simpleType>
                        </xs:attribute>
                      </xs:extension>
                    </xs:simpleContent>
                  </xs:complexType>
```

```
</xs:element>
<xs:element name="DHWDeliveredEnergyUse" minOccurs="0">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:float">
        <xs:attribute name="Key">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:pattern value="[B][S][I][0-9][0-9][0-9][0-9]" />

            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="TimeStep">
          <xs:simpleType>
            <xs:restriction base="xs:integer">
              <xs:minInclusive value="1" />
              <xs:maxInclusive value="8760" />
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
<xs:element name="ElectricityUse" minOccurs="0">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:float">
        <xs:attribute name="Key">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:pattern value="[B][S][I][0-9][0-9][0-9][0-9]" />

            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="TimeStep">
          <xs:simpleType>
            <xs:restriction base="xs:integer">
              <xs:minInclusive value="1" />
              <xs:maxInclusive value="8760" />
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
<xs:element name="NaturalGasUse" minOccurs="0">
```

```
            <xs:complexType>
               <xs:simpleContent>
                  <xs:extension base="xs:float">
                     <xs:attribute name="Key">
                        <xs:simpleType>
                           <xs:restriction base="xs:string">
                              <xs:pattern value="[B][S][I][0-9][0-
9][0-9][0-9]" />
                           </xs:restriction>
                        </xs:simpleType>
                     </xs:attribute>
                     <xs:attribute name="TimeStep">
                        <xs:simpleType>
                           <xs:restriction base="xs:integer">
                              <xs:minInclusive value="1" />
                              <xs:maxInclusive value="8760" />
                           </xs:restriction>
                        </xs:simpleType>
                     </xs:attribute>
                  </xs:extension>
               </xs:simpleContent>
            </xs:complexType>
         </xs:element>
      </xs:sequence>
      <xs:attribute name="Key">
         <xs:simpleType>
            <xs:restriction base="xs:string">
               <xs:pattern value="[B][S][0-9][0-9][0-9][0-9]" />
            </xs:restriction>
         </xs:simpleType>
      </xs:attribute>
   </xs:complexType>
</xs:element>
<xs:element name="ElectricVehicleChargingStation"
minOccurs="0" maxOccurs="unbounded">
   <xs:complexType>
      <xs:sequence>
         <xs:element name="NumberOfCars" minOccurs="0">
            <xs:complexType>
               <xs:simpleContent>
                  <xs:extension base="xs:float">
                     <xs:attribute name="Key">
                        <xs:simpleType>
                           <xs:restriction base="xs:string">
                              <xs:pattern value="[E][V][S][I][0-9][0-
9][0-9][0-9]" />
                           </xs:restriction>
                        </xs:simpleType>
                     </xs:attribute>
                     <xs:attribute name="TimeStep">
                        <xs:simpleType>
```

```
                                                <xs:restriction base="xs:integer">
                                                  <xs:minInclusive value="1" />
                                                  <xs:maxInclusive value="8760" />
                                                </xs:restriction>
                                              </xs:simpleType>
                                            </xs:attribute>
                                          </xs:extension>
                                        </xs:simpleContent>
                                      </xs:complexType>
                                    </xs:element>
                                    <xs:element name="ChargingPower" minOccurs="0">
                                      <xs:complexType>
                                        <xs:simpleContent>
                                          <xs:extension base="xs:float">
                                            <xs:attribute name="Key">
                                              <xs:simpleType>
                                                <xs:restriction base="xs:string">
                                                  <xs:pattern value="[E][V][S][I][0-9][0-
9][0-9][0-9]" />
                                                </xs:restriction>
                                              </xs:simpleType>
                                            </xs:attribute>
                                            <xs:attribute name="TimeStep">
                                              <xs:simpleType>
                                                <xs:restriction base="xs:integer">
                                                  <xs:minInclusive value="1" />
                                                  <xs:maxInclusive value="8760" />
                                                </xs:restriction>
                                              </xs:simpleType>
                                            </xs:attribute>
                                          </xs:extension>
                                        </xs:simpleContent>
                                      </xs:complexType>
                                    </xs:element>
                                  </xs:sequence>
                                  <xs:attribute name="Key">
                                    <xs:simpleType>
                                      <xs:restriction base="xs:string">
                                        <xs:pattern value="[E][S][0-9][0-9][0-9][0-9]" />
                                      </xs:restriction>
                                    </xs:simpleType>
                                  </xs:attribute>
                                </xs:complexType>
                              </xs:element>
                            </xs:sequence>
                          </xs:complexType>
                        </xs:element>
                        <xs:element name="Producer">
                          <xs:complexType>
                            <xs:sequence>
                              <xs:element name="PV" minOccurs="0"
```

```
maxOccurs="unbounded">
                            <xs:complexType>
                              <xs:sequence>
                                <xs:element name="ElectricityGeneration"
minOccurs="0">
                                  <xs:complexType>
                                    <xs:simpleContent>
                                      <xs:extension base="xs:float">
                                        <xs:attribute name="Key">
                                          <xs:simpleType>
                                            <xs:restriction base="xs:string">
                                              <xs:pattern value="[P][S][I][0-9][0-
9][0-9][0-9]" />
                                            </xs:restriction>
                                          </xs:simpleType>
                                        </xs:attribute>
                                        <xs:attribute name="TimeStep">
                                          <xs:simpleType>
                                            <xs:restriction base="xs:integer">
                                              <xs:minInclusive value="1" />
                                              <xs:maxInclusive value="8760" />
                                            </xs:restriction>
                                          </xs:simpleType>
                                        </xs:attribute>
                                      </xs:extension>
                                    </xs:simpleContent>
                                  </xs:complexType>
                                </xs:element>
                              </xs:sequence>
                              <xs:attribute name="Key" use="required">
                                <xs:simpleType>
                                  <xs:restriction base="xs:string">
                                    <xs:pattern value="[P][S][0-9][0-9][0-9][0-9]" />
                                  </xs:restriction>
                                </xs:simpleType>
                              </xs:attribute>
                            </xs:complexType>
                          </xs:element>
                          <xs:element name="WindTurbine" minOccurs="0"
maxOccurs="unbounded">
                            <xs:complexType>
                              <xs:sequence>
                                <xs:element name="ElectricityGeneration"
minOccurs="0">
                                  <xs:complexType>
                                    <xs:simpleContent>
                                      <xs:extension base="xs:float">
                                        <xs:attribute name="Key">
                                          <xs:simpleType>
                                            <xs:restriction base="xs:string">
                                              <xs:pattern value="[W][S][I][0-9][0-
```

```
9][0-9][0-9]" />
                                            </xs:restriction>
                                          </xs:simpleType>
                                        </xs:attribute>
                                        <xs:attribute name="TimeStep">
                                          <xs:simpleType>
                                            <xs:restriction base="xs:integer">
                                              <xs:minInclusive value="1" />
                                              <xs:maxInclusive value="8760" />
                                            </xs:restriction>
                                          </xs:simpleType>
                                        </xs:attribute>
                                      </xs:extension>
                                    </xs:simpleContent>
                                  </xs:complexType>
                                </xs:element>
                              </xs:sequence>
                              <xs:attribute name="Key">
                                <xs:simpleType>
                                  <xs:restriction base="xs:string">
                                    <xs:pattern value="[W][S][0-9][0-9][0-9][0-9]" />
                                  </xs:restriction>
                                </xs:simpleType>
                              </xs:attribute>
                            </xs:complexType>
                          </xs:element>
                          <xs:element name="SolarHotWater" minOccurs="0"
maxOccurs="unbounded">
                            <xs:complexType>
                              <xs:sequence>
                                <xs:element name="DHWGeneration" minOccurs="0">
                                  <xs:complexType>
                                    <xs:simpleContent>
                                      <xs:extension base="xs:float">
                                        <xs:attribute name="Key">
                                          <xs:simpleType>
                                            <xs:restriction base="xs:string">
                                              <xs:pattern value="[S][S][I][0-9][0-
9][0-9][0-9]" />
                                            </xs:restriction>
                                          </xs:simpleType>
                                        </xs:attribute>
                                        <xs:attribute name="TimeStep">
                                          <xs:simpleType>
                                            <xs:restriction base="xs:integer">
                                              <xs:minInclusive value="1" />
                                              <xs:maxInclusive value="8760" />
                                            </xs:restriction>
                                          </xs:simpleType>
                                        </xs:attribute>
                                      </xs:extension>
```

```
                                    </xs:simpleContent>
                                  </xs:complexType>
                                </xs:element>
                              </xs:sequence>
                              <xs:attribute name="Key">
                                <xs:simpleType>
                                  <xs:restriction base="xs:string">
                                    <xs:pattern value="[S][S][0-9][0-9][0-9][0-9]" />
                                  </xs:restriction>
                                </xs:simpleType>
                              </xs:attribute>
                            </xs:complexType>
                          </xs:element>
                        </xs:sequence>
                      </xs:complexType>
                    </xs:element>
                    <xs:element name="DistrictSystem">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element name="DistrictHeating" minOccurs="0"
maxOccurs="unbounded">
                            <xs:complexType>
                              <xs:sequence>
                                <xs:element name="HotWaterGen" minOccurs="0">
                                  <xs:complexType>
                                    <xs:simpleContent>
                                      <xs:extension base="xs:float">
                                        <xs:attribute name="Key">
                                          <xs:simpleType>
                                            <xs:restriction base="xs:string">
                                              <xs:pattern value="[D][H][S][I][0-9][0-
9][0-9][0-9]" />

                                            </xs:restriction>
                                          </xs:simpleType>
                                        </xs:attribute>
                                        <xs:attribute name="TimeStep">
                                          <xs:simpleType>
                                            <xs:restriction base="xs:integer">
                                              <xs:minInclusive value="1" />
                                              <xs:maxInclusive value="8760" />
                                            </xs:restriction>
                                          </xs:simpleType>
                                        </xs:attribute>
                                      </xs:extension>
                                    </xs:simpleContent>
                                  </xs:complexType>
                                </xs:element>
                                <xs:element name="SteamGen" minOccurs="0">
                                  <xs:complexType>
                                    <xs:simpleContent>
                                      <xs:extension base="xs:float">
```

```
                <xs:attribute name="Key">
                  <xs:simpleType>
                    <xs:restriction base="xs:string">
                      <xs:pattern value="[D][H][S][I][0-9][0-
9][0-9][0-9]" />
                    </xs:restriction>
                  </xs:simpleType>
                </xs:attribute>
                <xs:attribute name="TimeStep">
                  <xs:simpleType>
                    <xs:restriction base="xs:integer">
                      <xs:minInclusive value="1" />
                      <xs:maxInclusive value="8760" />
                    </xs:restriction>
                  </xs:simpleType>
                </xs:attribute>
              </xs:extension>
            </xs:simpleContent>
          </xs:complexType>
        </xs:element>
        <xs:element name="ElectricityUse" minOccurs="0">
          <xs:complexType>
            <xs:simpleContent>
              <xs:extension base="xs:float">
                <xs:attribute name="Key">
                  <xs:simpleType>
                    <xs:restriction base="xs:string">
                      <xs:pattern value="[D][H][S][I][0-9][0-
9][0-9][0-9]" />
                    </xs:restriction>
                  </xs:simpleType>
                </xs:attribute>
                <xs:attribute name="TimeStep">
                  <xs:simpleType>
                    <xs:restriction base="xs:integer">
                      <xs:minInclusive value="1" />
                      <xs:maxInclusive value="8760" />
                    </xs:restriction>
                  </xs:simpleType>
                </xs:attribute>
              </xs:extension>
            </xs:simpleContent>
          </xs:complexType>
        </xs:element>
        <xs:element name="NaturalGasUse" minOccurs="0">
          <xs:complexType>
            <xs:simpleContent>
              <xs:extension base="xs:float">
                <xs:attribute name="Key">
                  <xs:simpleType>
                    <xs:restriction base="xs:string">
```

```
                              <xs:pattern value="[D][H][S][I][0-9][0-
9][0-9][0-9]" />
                            </xs:restriction>
                          </xs:simpleType>
                        </xs:attribute>
                        <xs:attribute name="TimeStep">
                          <xs:simpleType>
                            <xs:restriction base="xs:integer">
                              <xs:minInclusive value="1" />
                              <xs:maxInclusive value="8760" />
                            </xs:restriction>
                          </xs:simpleType>
                        </xs:attribute>
                      </xs:extension>
                    </xs:simpleContent>
                  </xs:complexType>
                </xs:element>
                <xs:element name="GasolineUse" minOccurs="0">
                  <xs:complexType>
                    <xs:simpleContent>
                      <xs:extension base="xs:float">
                        <xs:attribute name="Key">
                          <xs:simpleType>
                            <xs:restriction base="xs:string">
                              <xs:pattern value="[D][H][S][I][0-9][0-
9][0-9][0-9]" />
                            </xs:restriction>
                          </xs:simpleType>
                        </xs:attribute>
                        <xs:attribute name="TimeStep">
                          <xs:simpleType>
                            <xs:restriction base="xs:integer">
                              <xs:minInclusive value="1" />
                              <xs:maxInclusive value="8760" />
                            </xs:restriction>
                          </xs:simpleType>
                        </xs:attribute>
                      </xs:extension>
                    </xs:simpleContent>
                  </xs:complexType>
                </xs:element>
                <xs:element name="DieselUse" minOccurs="0">
                  <xs:complexType>
                    <xs:simpleContent>
                      <xs:extension base="xs:float">
                        <xs:attribute name="Key">
                          <xs:simpleType>
                            <xs:restriction base="xs:string">
                              <xs:pattern value="[D][H][S][I][0-9][0-
9][0-9][0-9]" />
                            </xs:restriction>
```

```
                                          </xs:simpleType>
                                        </xs:attribute>
                                        <xs:attribute name="TimeStep">
                                          <xs:simpleType>
                                            <xs:restriction base="xs:integer">
                                              <xs:minInclusive value="1" />
                                              <xs:maxInclusive value="8760" />
                                            </xs:restriction>
                                          </xs:simpleType>
                                        </xs:attribute>
                                      </xs:extension>
                                    </xs:simpleContent>
                                  </xs:complexType>
                                </xs:element>
                                <xs:element name="PropaneUse" minOccurs="0">
                                  <xs:complexType>
                                    <xs:simpleContent>
                                      <xs:extension base="xs:float">
                                        <xs:attribute name="Key">
                                          <xs:simpleType>
                                            <xs:restriction base="xs:string">
                                              <xs:pattern value="[D][H][S][I][0-9][0-
9][0-9][0-9]" />

                                            </xs:restriction>
                                          </xs:simpleType>
                                        </xs:attribute>
                                        <xs:attribute name="TimeStep">
                                          <xs:simpleType>
                                            <xs:restriction base="xs:integer">
                                              <xs:minInclusive value="1" />
                                              <xs:maxInclusive value="8760" />
                                            </xs:restriction>
                                          </xs:simpleType>
                                        </xs:attribute>
                                      </xs:extension>
                                    </xs:simpleContent>
                                  </xs:complexType>
                                </xs:element>
                                <xs:element name="OilUse" minOccurs="0">
                                  <xs:complexType>
                                    <xs:simpleContent>
                                      <xs:extension base="xs:float">
                                        <xs:attribute name="Key">
                                          <xs:simpleType>
                                            <xs:restriction base="xs:string">
                                              <xs:pattern value="[D][H][S][I][0-9][0-
9][0-9][0-9]" />

                                            </xs:restriction>
                                          </xs:simpleType>
                                        </xs:attribute>
                                        <xs:attribute name="TimeStep">
```

```
                                          <xs:simpleType>
                                            <xs:restriction base="xs:integer">
                                              <xs:minInclusive value="1" />
                                              <xs:maxInclusive value="8760" />
                                            </xs:restriction>
                                          </xs:simpleType>
                                        </xs:attribute>
                                      </xs:extension>
                                    </xs:simpleContent>
                                  </xs:complexType>
                                </xs:element>
                              </xs:sequence>
                              <xs:attribute name="Key">
                                <xs:simpleType>
                                  <xs:restriction base="xs:string">
                                    <xs:pattern value="[D][H][S][0-9][0-9][0-9][0-9]"
/>
                                  </xs:restriction>
                                </xs:simpleType>
                              </xs:attribute>
                            </xs:complexType>
                          </xs:element>
                          <xs:element name="DistrictCooling" minOccurs="0"
maxOccurs="unbounded">
                            <xs:complexType>
                              <xs:sequence>
                                <xs:element name="ChilledWaterGen" minOccurs="0">
                                  <xs:complexType>
                                    <xs:simpleContent>
                                      <xs:extension base="xs:float">
                                        <xs:attribute name="Key">
                                          <xs:simpleType>
                                            <xs:restriction base="xs:string">
                                              <xs:pattern value="[D][C][S][I][0-9][0-
9][0-9][0-9]" />
                                            </xs:restriction>
                                          </xs:simpleType>
                                        </xs:attribute>
                                        <xs:attribute name="TimeStep">
                                          <xs:simpleType>
                                            <xs:restriction base="xs:integer">
                                              <xs:minInclusive value="1" />
                                              <xs:maxInclusive value="8760" />
                                            </xs:restriction>
                                          </xs:simpleType>
                                        </xs:attribute>
                                      </xs:extension>
                                    </xs:simpleContent>
                                  </xs:complexType>
                                </xs:element>
                                <xs:element name="ElectricityUse" minOccurs="0">
```

```
                  <xs:complexType>
                    <xs:simpleContent>
                      <xs:extension base="xs:float">
                        <xs:attribute name="Key">
                          <xs:simpleType>
                            <xs:restriction base="xs:string">
                              <xs:pattern value="[D][C][S][I][0-9][0-
9][0-9][0-9]" />

                            </xs:restriction>
                          </xs:simpleType>
                        </xs:attribute>
                        <xs:attribute name="TimeStep">
                          <xs:simpleType>
                            <xs:restriction base="xs:integer">
                              <xs:minInclusive value="1" />
                              <xs:maxInclusive value="8760" />
                            </xs:restriction>
                          </xs:simpleType>
                        </xs:attribute>
                      </xs:extension>
                    </xs:simpleContent>
                  </xs:complexType>
                </xs:element>
                <xs:element name="NaturalGasUse" minOccurs="0">
                  <xs:complexType>
                    <xs:simpleContent>
                      <xs:extension base="xs:float">
                        <xs:attribute name="Key">
                          <xs:simpleType>
                            <xs:restriction base="xs:string">
                              <xs:pattern value="[D][C][S][I][0-9][0-
9][0-9][0-9]" />

                            </xs:restriction>
                          </xs:simpleType>
                        </xs:attribute>
                        <xs:attribute name="TimeStep">
                          <xs:simpleType>
                            <xs:restriction base="xs:integer">
                              <xs:minInclusive value="1" />
                              <xs:maxInclusive value="8760" />
                            </xs:restriction>
                          </xs:simpleType>
                        </xs:attribute>
                      </xs:extension>
                    </xs:simpleContent>
                  </xs:complexType>
                </xs:element>
                <xs:element name="GasolineUse" minOccurs="0">
                  <xs:complexType>
                    <xs:simpleContent>
                      <xs:extension base="xs:float">
```

```
                                        <xs:attribute name="Key">
                                          <xs:simpleType>
                                            <xs:restriction base="xs:string">
                                              <xs:pattern value="[D][C][S][I][0-9][0-
9][0-9][0-9]" />
                                            </xs:restriction>
                                          </xs:simpleType>
                                        </xs:attribute>
                                        <xs:attribute name="TimeStep">
                                          <xs:simpleType>
                                            <xs:restriction base="xs:integer">
                                              <xs:minInclusive value="1" />
                                              <xs:maxInclusive value="8760" />
                                            </xs:restriction>
                                          </xs:simpleType>
                                        </xs:attribute>
                                      </xs:extension>
                                    </xs:simpleContent>
                                  </xs:complexType>
                                </xs:element>
                                <xs:element name="DieselUse" minOccurs="0">
                                  <xs:complexType>
                                    <xs:simpleContent>
                                      <xs:extension base="xs:float">
                                        <xs:attribute name="Key">
                                          <xs:simpleType>
                                            <xs:restriction base="xs:string">
                                              <xs:pattern value="[D][C][S][I][0-9][0-
9][0-9][0-9]" />
                                            </xs:restriction>
                                          </xs:simpleType>
                                        </xs:attribute>
                                        <xs:attribute name="TimeStep">
                                          <xs:simpleType>
                                            <xs:restriction base="xs:integer">
                                              <xs:minInclusive value="1" />
                                              <xs:maxInclusive value="8760" />
                                            </xs:restriction>
                                          </xs:simpleType>
                                        </xs:attribute>
                                      </xs:extension>
                                    </xs:simpleContent>
                                  </xs:complexType>
                                </xs:element>
                                <xs:element name="PropaneUse" minOccurs="0">
                                  <xs:complexType>
                                    <xs:simpleContent>
                                      <xs:extension base="xs:float">
                                        <xs:attribute name="Key">
                                          <xs:simpleType>
                                            <xs:restriction base="xs:string">
```

```
                                              <xs:pattern value="[D][C][S][I][0-9][0-
9][0-9][0-9]" />
                                            </xs:restriction>
                                          </xs:simpleType>
                                        </xs:attribute>
                                        <xs:attribute name="TimeStep">
                                          <xs:simpleType>
                                            <xs:restriction base="xs:integer">
                                              <xs:minInclusive value="1" />
                                              <xs:maxInclusive value="8760" />
                                            </xs:restriction>
                                          </xs:simpleType>
                                        </xs:attribute>
                                      </xs:extension>
                                    </xs:simpleContent>
                                  </xs:complexType>
                                </xs:element>
                                <xs:element name="OilUse" minOccurs="0">
                                  <xs:complexType>
                                    <xs:simpleContent>
                                      <xs:extension base="xs:float">
                                        <xs:attribute name="Key">
                                          <xs:simpleType>
                                            <xs:restriction base="xs:string">
                                              <xs:pattern value="[D][C][S][I][0-9][0-
9][0-9][0-9]" />
                                            </xs:restriction>
                                          </xs:simpleType>
                                        </xs:attribute>
                                        <xs:attribute name="TimeStep">
                                          <xs:simpleType>
                                            <xs:restriction base="xs:integer">
                                              <xs:minInclusive value="1" />
                                              <xs:maxInclusive value="8760" />
                                            </xs:restriction>
                                          </xs:simpleType>
                                        </xs:attribute>
                                      </xs:extension>
                                    </xs:simpleContent>
                                  </xs:complexType>
                                </xs:element>
                              </xs:sequence>
                              <xs:attribute name="Key">
                                <xs:simpleType>
                                  <xs:restriction base="xs:string">
                                    <xs:pattern value="[D][C][S][0-9][0-9][0-9][0-9]"
/>
                                  </xs:restriction>
                                </xs:simpleType>
                              </xs:attribute>
                            </xs:complexType>
```

```
                          </xs:element>
                          <xs:element name="Cogeneration" minOccurs="0"
maxOccurs="unbounded">
                            <xs:complexType>
                              <xs:sequence>
                                <xs:element name="HotWaterGen" minOccurs="0">
                                  <xs:complexType>
                                    <xs:simpleContent>
                                      <xs:extension base="xs:float">
                                        <xs:attribute name="Key">
                                          <xs:simpleType>
                                            <xs:restriction base="xs:string">
                                              <xs:pattern value="[C][S][I][0-9][0-
9][0-9][0-9]" />

                                            </xs:restriction>
                                          </xs:simpleType>
                                        </xs:attribute>
                                        <xs:attribute name="TimeStep">
                                          <xs:simpleType>
                                            <xs:restriction base="xs:integer">
                                              <xs:minInclusive value="1" />
                                              <xs:maxInclusive value="8760" />
                                            </xs:restriction>
                                          </xs:simpleType>
                                        </xs:attribute>
                                      </xs:extension>
                                    </xs:simpleContent>
                                  </xs:complexType>
                                </xs:element>
                                <xs:element name="SteamGen" minOccurs="0">
                                  <xs:complexType>
                                    <xs:simpleContent>
                                      <xs:extension base="xs:float">
                                        <xs:attribute name="Key">
                                          <xs:simpleType>
                                            <xs:restriction base="xs:string">
                                              <xs:pattern value="[C][S][I][0-9][0-
9][0-9][0-9]" />

                                            </xs:restriction>
                                          </xs:simpleType>
                                        </xs:attribute>
                                        <xs:attribute name="TimeStep">
                                          <xs:simpleType>
                                            <xs:restriction base="xs:integer">
                                              <xs:minInclusive value="1" />
                                              <xs:maxInclusive value="8760" />
                                            </xs:restriction>
                                          </xs:simpleType>
                                        </xs:attribute>
                                      </xs:extension>
                                    </xs:simpleContent>
```

```
                                  </xs:complexType>
                                </xs:element>
                                <xs:element name="ElectricityGen" minOccurs="0">
                                  <xs:complexType>
                                    <xs:simpleContent>
                                      <xs:extension base="xs:float">
                                        <xs:attribute name="Key">
                                          <xs:simpleType>
                                            <xs:restriction base="xs:string">
                                              <xs:pattern value="[C][S][I][0-9][0-
9][0-9][0-9]" />

                                            </xs:restriction>
                                          </xs:simpleType>
                                        </xs:attribute>
                                        <xs:attribute name="TimeStep">
                                          <xs:simpleType>
                                            <xs:restriction base="xs:integer">
                                              <xs:minInclusive value="1" />
                                              <xs:maxInclusive value="8760" />
                                            </xs:restriction>
                                          </xs:simpleType>
                                        </xs:attribute>
                                      </xs:extension>
                                    </xs:simpleContent>
                                  </xs:complexType>
                                </xs:element>
                                <xs:element name="ElectricityUse" minOccurs="0">
                                  <xs:complexType>
                                    <xs:simpleContent>
                                      <xs:extension base="xs:float">
                                        <xs:attribute name="Key">
                                          <xs:simpleType>
                                            <xs:restriction base="xs:string">
                                              <xs:pattern value="[C][S][I][0-9][0-
9][0-9][0-9]" />

                                            </xs:restriction>
                                          </xs:simpleType>
                                        </xs:attribute>
                                        <xs:attribute name="TimeStep">
                                          <xs:simpleType>
                                            <xs:restriction base="xs:integer">
                                              <xs:minInclusive value="1" />
                                              <xs:maxInclusive value="8760" />
                                            </xs:restriction>
                                          </xs:simpleType>
                                        </xs:attribute>
                                      </xs:extension>
                                    </xs:simpleContent>
                                  </xs:complexType>
                                </xs:element>
                                <xs:element name="NaturalGasUse" minOccurs="0">
```

```
                   <xs:complexType>
                      <xs:simpleContent>
                         <xs:extension base="xs:float">
                            <xs:attribute name="Key">
                               <xs:simpleType>
                                  <xs:restriction base="xs:string">
                                     <xs:pattern value="[C][S][I][0-9][0-
9][0-9][0-9]" />

                                  </xs:restriction>
                               </xs:simpleType>
                            </xs:attribute>
                            <xs:attribute name="TimeStep">
                               <xs:simpleType>
                                  <xs:restriction base="xs:integer">
                                     <xs:minInclusive value="1" />
                                     <xs:maxInclusive value="8760" />
                                  </xs:restriction>
                               </xs:simpleType>
                            </xs:attribute>
                         </xs:extension>
                      </xs:simpleContent>
                   </xs:complexType>
                </xs:element>
                <xs:element name="GasolineUse" minOccurs="0">
                   <xs:complexType>
                      <xs:simpleContent>
                         <xs:extension base="xs:float">
                            <xs:attribute name="Key">
                               <xs:simpleType>
                                  <xs:restriction base="xs:string">
                                     <xs:pattern value="[C][S][I][0-9][0-
9][0-9][0-9]" />

                                  </xs:restriction>
                               </xs:simpleType>
                            </xs:attribute>
                            <xs:attribute name="TimeStep">
                               <xs:simpleType>
                                  <xs:restriction base="xs:integer">
                                     <xs:minInclusive value="1" />
                                     <xs:maxInclusive value="8760" />
                                  </xs:restriction>
                               </xs:simpleType>
                            </xs:attribute>
                         </xs:extension>
                      </xs:simpleContent>
                   </xs:complexType>
                </xs:element>
                <xs:element name="DieselUse" minOccurs="0">
                   <xs:complexType>
                      <xs:simpleContent>
                         <xs:extension base="xs:float">
```

```
                        <xs:attribute name="Key">
                          <xs:simpleType>
                            <xs:restriction base="xs:string">
                              <xs:pattern value="[C][S][I][0-9][0-
9][0-9][0-9]" />
                            </xs:restriction>
                          </xs:simpleType>
                        </xs:attribute>
                        <xs:attribute name="TimeStep">
                          <xs:simpleType>
                            <xs:restriction base="xs:integer">
                              <xs:minInclusive value="1" />
                              <xs:maxInclusive value="8760" />
                            </xs:restriction>
                          </xs:simpleType>
                        </xs:attribute>
                      </xs:extension>
                    </xs:simpleContent>
                  </xs:complexType>
                </xs:element>
                <xs:element name="PropaneUse" minOccurs="0">
                  <xs:complexType>
                    <xs:simpleContent>
                      <xs:extension base="xs:float">
                        <xs:attribute name="Key">
                          <xs:simpleType>
                            <xs:restriction base="xs:string">
                              <xs:pattern value="[C][S][I][0-9][0-
9][0-9][0-9]" />
                            </xs:restriction>
                          </xs:simpleType>
                        </xs:attribute>
                        <xs:attribute name="TimeStep">
                          <xs:simpleType>
                            <xs:restriction base="xs:integer">
                              <xs:minInclusive value="1" />
                              <xs:maxInclusive value="8760" />
                            </xs:restriction>
                          </xs:simpleType>
                        </xs:attribute>
                      </xs:extension>
                    </xs:simpleContent>
                  </xs:complexType>
                </xs:element>
                <xs:element name="OilUse" minOccurs="0">
                  <xs:complexType>
                    <xs:simpleContent>
                      <xs:extension base="xs:float">
                        <xs:attribute name="Key">
                          <xs:simpleType>
                            <xs:restriction base="xs:string">
```

```
                                                                     <xs:pattern value="[C][S][I][0-9][0-
9][0-9][0-9]" />
                                                                  </xs:restriction>
                                                               </xs:simpleType>
                                                            </xs:attribute>
                                                            <xs:attribute name="TimeStep">
                                                               <xs:simpleType>
                                                                  <xs:restriction base="xs:integer">
                                                                     <xs:minInclusive value="1" />
                                                                     <xs:maxInclusive value="8760" />
                                                                  </xs:restriction>
                                                               </xs:simpleType>
                                                            </xs:attribute>
                                                         </xs:extension>
                                                      </xs:simpleContent>
                                                   </xs:complexType>
                                                </xs:element>
                                             </xs:sequence>
                                             <xs:attribute name="Key">
                                                <xs:simpleType>
                                                   <xs:restriction base="xs:string">
                                                      <xs:pattern value="[C][S][0-9][0-9][0-9][0-9]" />
                                                   </xs:restriction>
                                                </xs:simpleType>
                                             </xs:attribute>
                                          </xs:complexType>
                                       </xs:element>
                                    </xs:sequence>
                                 </xs:complexType>
                              </xs:element>
                              <xs:element name="Storage">
                                 <xs:complexType>
                                    <xs:sequence>
                                       <xs:element name="ElectricityBattery" minOccurs="0"
maxOccurs="unbounded">
                                          <xs:complexType>
                                             <xs:sequence>
                                                <xs:element name="State" minOccurs="0">
                                                   <xs:complexType>
                                                      <xs:simpleContent>
                                                         <xs:extension base="xs:float">
                                                            <xs:attribute name="Key">
                                                               <xs:simpleType>
                                                                  <xs:restriction base="xs:string">
                                                                     <xs:pattern value="[E][B][S][I][0-9][0-
9][0-9][0-9]" />
                                                                  </xs:restriction>
                                                               </xs:simpleType>
                                                            </xs:attribute>
                                                            <xs:attribute name="TimeStep">
                                                               <xs:simpleType>
```

```
                                            <xs:restriction base="xs:integer">
                                              <xs:minInclusive value="1" />
                                              <xs:maxInclusive value="8760" />
                                            </xs:restriction>
                                          </xs:simpleType>
                                        </xs:attribute>
                                      </xs:extension>
                                    </xs:simpleContent>
                                  </xs:complexType>
                                </xs:element>
                                <xs:element name="ChargingPower" minOccurs="0">
                                  <xs:complexType>
                                    <xs:simpleContent>
                                      <xs:extension base="xs:float">
                                        <xs:attribute name="Key">
                                          <xs:simpleType>
                                            <xs:restriction base="xs:string">
                                              <xs:pattern value="[E][B][S][I][0-9][0-
9][0-9][0-9]" />

                                            </xs:restriction>
                                          </xs:simpleType>
                                        </xs:attribute>
                                        <xs:attribute name="TimeStep">
                                          <xs:simpleType>
                                            <xs:restriction base="xs:integer">
                                              <xs:minInclusive value="1" />
                                              <xs:maxInclusive value="8760" />
                                            </xs:restriction>
                                          </xs:simpleType>
                                        </xs:attribute>
                                      </xs:extension>
                                    </xs:simpleContent>
                                  </xs:complexType>
                                </xs:element>
                                <xs:element name="DisChargingPower"
minOccurs="0">
                                  <xs:complexType>
                                    <xs:simpleContent>
                                      <xs:extension base="xs:float">
                                        <xs:attribute name="Key">
                                          <xs:simpleType>
                                            <xs:restriction base="xs:string">
                                              <xs:pattern value="[E][B][S][I][0-9][0-
9][0-9][0-9]" />

                                            </xs:restriction>
                                          </xs:simpleType>
                                        </xs:attribute>
                                        <xs:attribute name="TimeStep">
                                          <xs:simpleType>
                                            <xs:restriction base="xs:integer">
                                              <xs:minInclusive value="1" />
```

```
                                              <xs:maxInclusive value="8760" />
                                            </xs:restriction>
                                          </xs:simpleType>
                                        </xs:attribute>
                                      </xs:extension>
                                    </xs:simpleContent>
                                  </xs:complexType>
                                </xs:element>
                              </xs:sequence>
                              <xs:attribute name="Key">
                                <xs:simpleType>
                                  <xs:restriction base="xs:string">
                                    <xs:pattern value="[E][B][S][0-9][0-9][0-9][0-9]" />
                                  </xs:restriction>
                                </xs:simpleType>
                              </xs:attribute>
                            </xs:complexType>
                          </xs:element>
                        </xs:sequence>
                      </xs:complexType>
                    </xs:element>
                  </xs:sequence>
                  <xs:attribute name="ID" fixed="L1000" type="xs:string" />
                </xs:complexType>
              </xs:element>
              <xs:element name="Controller">
                <xs:complexType>
                  <xs:sequence minOccurs="0" maxOccurs="unbounded">
                    <xs:element name="Condition" type="xs:string" />
                    <xs:element name="Statement" type="xs:string" />
                  </xs:sequence>
                  <xs:attribute name="ID" fixed="L2000" type="xs:string" />
                </xs:complexType>
              </xs:element>
              <xs:element name="Actuator">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="DiscontinuousControl">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element name="Consumer">
                            <xs:complexType>
                              <xs:sequence>
                                <xs:element name="Building" minOccurs="0"
maxOccurs="unbounded">
                                  <xs:complexType>
                                    <xs:sequence>
                                      <xs:element name="WindowOverhangAngle"
minOccurs="0" maxOccurs="16">
                                        <xs:complexType>
                                          <xs:simpleContent>
```

```
                    <xs:extension
base="OverhangFinAngles">
                      <xs:attribute name="Key">
                        <xs:simpleType>
                          <xs:restriction base="xs:string">
                            <xs:pattern value="[B][D][I][0-
9][0-9][0-9][0-9]" />
                          </xs:restriction>
                        </xs:simpleType>
                      </xs:attribute>
                      <xs:attribute name="Orientation">
                        <xs:simpleType>
                          <xs:restriction base="xs:string">
                            <xs:enumeration value="South"
/>
                            <xs:enumeration
value="Southeast" />
                            <xs:enumeration value="East"
/>
                            <xs:enumeration
value="Northeast" />
                            <xs:enumeration value="North"
/>
                            <xs:enumeration
value="Northwest" />
                            <xs:enumeration value="West"
/>
                            <xs:enumeration
value="Southwest" />
                          </xs:restriction>
                        </xs:simpleType>
                      </xs:attribute>
                      <xs:attribute
name="ConstructionType">
                        <xs:simpleType>
                          <xs:restriction base="xs:integer">
                            <xs:minInclusive value="1" />
                            <xs:maxInclusive value="2" />
                          </xs:restriction>
                        </xs:simpleType>
                      </xs:attribute>
                    </xs:extension>
                  </xs:simpleContent>
                </xs:complexType>
              </xs:element>
              <xs:element name="WindowFinAngle"
minOccurs="0" maxOccurs="16">
                <xs:complexType>
                  <xs:simpleContent>
                    <xs:extension
base="OverhangFinAngles">
```

```
                              <xs:attribute name="Key">
                                 <xs:simpleType>
                                    <xs:restriction base="xs:string">
                                       <xs:pattern value="[B][D][I][0-
9][0-9][0-9][0-9]" />

                                    </xs:restriction>
                                 </xs:simpleType>
                              </xs:attribute>
                              <xs:attribute name="Orientation">
                                 <xs:simpleType>
                                    <xs:restriction base="xs:string">
                                       <xs:enumeration value="South"
/>

                                       <xs:enumeration
value="Southeast" />

                                       <xs:enumeration value="East"
/>

                                       <xs:enumeration
value="Northeast" />

                                       <xs:enumeration value="North"
/>

                                       <xs:enumeration
value="Northwest" />

                                       <xs:enumeration value="West"
/>

                                       <xs:enumeration
value="Southwest" />

                                    </xs:restriction>
                                 </xs:simpleType>
                              </xs:attribute>
                              <xs:attribute
name="ConstructionType">

                                 <xs:simpleType>
                                    <xs:restriction base="xs:integer">
                                       <xs:minInclusive value="1" />
                                       <xs:maxInclusive value="2" />
                                    </xs:restriction>
                                 </xs:simpleType>
                              </xs:attribute>
                           </xs:extension>
                        </xs:simpleContent>
                     </xs:complexType>
                  </xs:element>
               </xs:sequence>
               <xs:attribute name="Key">
                  <xs:simpleType>
                     <xs:restriction base="xs:string">
                        <xs:pattern value="[B][D][0-9][0-9][0-9][0-
9]" />

                     </xs:restriction>
                  </xs:simpleType>
```

```
                                    </xs:attribute>
                                  </xs:complexType>
                                </xs:element>
                              </xs:sequence>
                            </xs:complexType>
                          </xs:element>
                          <xs:element name="Producer">
                            <xs:complexType>
                              <xs:sequence>
                                <xs:element name="PV" minOccurs="0"
maxOccurs="unbounded">
                                  <xs:complexType>
                                    <xs:all>
                                      <xs:element name="ModuleTiltAngle"
minOccurs="0">
                                        <xs:complexType>
                                          <xs:simpleContent>
                                            <xs:extension base="Slope">
                                              <xs:attribute name="Key">
                                                <xs:simpleType>
                                                  <xs:restriction base="xs:string">
                                                    <xs:pattern value="[P][D][I][0-
9][0-9][0-9][0-9]" />
                                                  </xs:restriction>
                                                </xs:simpleType>
                                              </xs:attribute>
                                            </xs:extension>
                                          </xs:simpleContent>
                                        </xs:complexType>
                                      </xs:element>
                                      <xs:element name="ModuleOrientation"
minOccurs="0">
                                        <xs:complexType>
                                          <xs:simpleContent>
                                            <xs:extension base="Orientation">
                                              <xs:attribute name="Key">
                                                <xs:simpleType>
                                                  <xs:restriction base="xs:string">
                                                    <xs:pattern value="[P][D][I][0-
9][0-9][0-9][0-9]" />
                                                  </xs:restriction>
                                                </xs:simpleType>
                                              </xs:attribute>
                                            </xs:extension>
                                          </xs:simpleContent>
                                        </xs:complexType>
                                      </xs:element>
                                    </xs:all>
                                    <xs:attribute name="Key">
                                      <xs:simpleType>
                                        <xs:restriction base="xs:string">
```

```
                                          <xs:pattern value="[P][D][0-9][0-9][0-9][0-
9]" />
                                        </xs:restriction>
                                      </xs:simpleType>
                                    </xs:attribute>
                                  </xs:complexType>
                                </xs:element>
                                <xs:element name="SolarHotWater" minOccurs="0"
maxOccurs="unbounded">
                                  <xs:complexType>
                                    <xs:all>
                                      <xs:element name="CollectorTiltAngle"
minOccurs="0">
                                        <xs:complexType>
                                          <xs:simpleContent>
                                            <xs:extension base="Slope">
                                              <xs:attribute name="Key">
                                                <xs:simpleType>
                                                  <xs:restriction base="xs:string">
                                                    <xs:pattern value="[S][D][I][0-
9][0-9][0-9][0-9]" />
                                                  </xs:restriction>
                                                </xs:simpleType>
                                              </xs:attribute>
                                            </xs:extension>
                                          </xs:simpleContent>
                                        </xs:complexType>
                                      </xs:element>
                                      <xs:element name="CollectorOrientation"
minOccurs="0">
                                        <xs:complexType>
                                          <xs:simpleContent>
                                            <xs:extension base="Orientation">
                                              <xs:attribute name="Key">
                                                <xs:simpleType>
                                                  <xs:restriction base="xs:string">
                                                    <xs:pattern value="[S][D][I][0-
9][0-9][0-9][0-9]" />
                                                  </xs:restriction>
                                                </xs:simpleType>
                                              </xs:attribute>
                                            </xs:extension>
                                          </xs:simpleContent>
                                        </xs:complexType>
                                      </xs:element>
                                    </xs:all>
                                    <xs:attribute name="Key">
                                      <xs:simpleType>
                                        <xs:restriction base="xs:string">
                                          <xs:pattern value="[S][D][0-9][0-9][0-9][0-
9]" />
```

```
                                                            </xs:restriction>
                                                         </xs:simpleType>
                                                      </xs:attribute>
                                                   </xs:complexType>
                                                </xs:element>
                                             </xs:sequence>
                                          </xs:complexType>
                                       </xs:element>
                                    </xs:sequence>
                                 </xs:complexType>
                              </xs:element>
                              <xs:element name="ContinuousControl">
                                 <xs:complexType>
                                    <xs:sequence>
                                       <xs:element name="Consumer">
                                          <xs:complexType>
                                             <xs:sequence>
                                                <xs:element name="Building" minOccurs="0"
maxOccurs="unbounded">
                                                   <xs:complexType>
                                                      <xs:all>
                                                         <xs:element name="AppliancePowerDensity"
minOccurs="0">
                                                            <xs:complexType>
                                                               <xs:simpleContent>
                                                                  <xs:extension base="xs:float">
                                                                     <xs:attribute name="Key">
                                                                        <xs:simpleType>
                                                                           <xs:restriction base="xs:string">
                                                                              <xs:pattern value="[B][A][I][0-
9][0-9][0-9][0-9]" />
                                                                           </xs:restriction>
                                                                        </xs:simpleType>
                                                                     </xs:attribute>
                                                                  </xs:extension>
                                                               </xs:simpleContent>
                                                            </xs:complexType>
                                                         </xs:element>
                                                         <xs:element name="LightingPowerDensity"
minOccurs="0">
                                                            <xs:complexType>
                                                               <xs:simpleContent>
                                                                  <xs:extension base="xs:float">
                                                                     <xs:attribute name="Key">
                                                                        <xs:simpleType>
                                                                           <xs:restriction base="xs:string">
                                                                              <xs:pattern value="[B][A][I][0-
9][0-9][0-9][0-9]" />
                                                                           </xs:restriction>
                                                                        </xs:simpleType>
                                                                     </xs:attribute>
```

```
                                            </xs:extension>
                                          </xs:simpleContent>
                                        </xs:complexType>
                                      </xs:element>
                                      <xs:element name="HeatingSetPoint"
minOccurs="0">

                                        <xs:complexType>
                                          <xs:simpleContent>
                                            <xs:extension base="xs:float">
                                              <xs:attribute name="Key">
                                                <xs:simpleType>
                                                  <xs:restriction base="xs:string">
                                                    <xs:pattern value="[B][A][I][0-
9][0-9][0-9][0-9]" />

                                                  </xs:restriction>
                                                </xs:simpleType>
                                              </xs:attribute>
                                            </xs:extension>
                                          </xs:simpleContent>
                                        </xs:complexType>
                                      </xs:element>
                                      <xs:element name="CoolingSetPoint"
minOccurs="0">

                                        <xs:complexType>
                                          <xs:simpleContent>
                                            <xs:extension base="xs:float">
                                              <xs:attribute name="Key">
                                                <xs:simpleType>
                                                  <xs:restriction base="xs:string">
                                                    <xs:pattern value="[B][A][I][0-
9][0-9][0-9][0-9]" />

                                                  </xs:restriction>
                                                </xs:simpleType>
                                              </xs:attribute>
                                            </xs:extension>
                                          </xs:simpleContent>
                                        </xs:complexType>
                                      </xs:element>
                                      <xs:element
name="NaturalVentilationOpeningRatio" minOccurs="0">
                                        <xs:complexType>
                                          <xs:simpleContent>
                                            <xs:extension base="xs:float">
                                              <xs:attribute name="Key">
                                                <xs:simpleType>
                                                  <xs:restriction base="xs:string">
                                                    <xs:pattern value="[B][A][I][0-
9][0-9][0-9][0-9]" />

                                                  </xs:restriction>
                                                </xs:simpleType>
                                              </xs:attribute>
```

```
                                            </xs:extension>
                                          </xs:simpleContent>
                                        </xs:complexType>
                                      </xs:element>
                                    </xs:all>
                                    <xs:attribute name="Key">
                                      <xs:simpleType>
                                        <xs:restriction base="xs:string">
                                          <xs:pattern value="[B][C][0-9][0-9][0-9][0-
9]" />
                                        </xs:restriction>
                                      </xs:simpleType>
                                    </xs:attribute>
                                  </xs:complexType>
                                </xs:element>
                              </xs:sequence>
                            </xs:complexType>
                          </xs:element>
                        </xs:sequence>
                      </xs:complexType>
                    </xs:element>
                  </xs:sequence>
                  <xs:attribute name="ID" fixed="L3000" type="xs:string" />
                </xs:complexType>
              </xs:element>
              <xs:element name="OptimalControl">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="Consumer">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element name="Building" minOccurs="0"
maxOccurs="unbounded">
                            <xs:complexType>
                              <xs:sequence>
                                <xs:element name="TemperatureSetpoint"
minOccurs="0">
                                  <xs:complexType>
                                    <xs:complexContent>
                                      <xs:extension
base="OptimalControlConstraintInput">
                                        <xs:attribute name="Key">
                                          <xs:simpleType>
                                            <xs:restriction base="xs:string">
                                              <xs:pattern value="[B][E][F][M][I][0-
9][0-9][0-9][0-9]" />
                                            </xs:restriction>
                                          </xs:simpleType>
                                        </xs:attribute>
                                      </xs:extension>
                                    </xs:complexContent>
```

```
                                  </xs:complexType>
                                </xs:element>
                                <xs:element name="CoolingSetpointTemperatureFloat"
minOccurs="0">
                                  <xs:complexType>
                                    <xs:complexContent>
                                      <xs:extension
base="OptimalControlConstraintInput">
                                        <xs:attribute name="Key">
                                          <xs:simpleType>
                                            <xs:restriction base="xs:string">
                                              <xs:pattern value="[B][E][F][M][I][0-
9][0-9][0-9][0-9]" />
                                            </xs:restriction>
                                          </xs:simpleType>
                                        </xs:attribute>
                                      </xs:extension>
                                    </xs:complexContent>
                                  </xs:complexType>
                                </xs:element>
                                <xs:element name="LightingDimmer" minOccurs="0">
                                  <xs:complexType>
                                    <xs:complexContent>
                                      <xs:extension
base="OptimalControlConstraintInput">
                                        <xs:attribute name="Key">
                                          <xs:simpleType>
                                            <xs:restriction base="xs:string">
                                              <xs:pattern value="[B][E][F][M][I][0-
9][0-9][0-9][0-9]" />
                                            </xs:restriction>
                                          </xs:simpleType>
                                        </xs:attribute>
                                      </xs:extension>
                                    </xs:complexContent>
                                  </xs:complexType>
                                </xs:element>
                                <xs:element name="VoltageThrottling" minOccurs="0">
                                  <xs:complexType>
                                    <xs:complexContent>
                                      <xs:extension
base="OptimalControlConstraintInput">
                                        <xs:attribute name="Key">
                                          <xs:simpleType>
                                            <xs:restriction base="xs:string">
                                              <xs:pattern value="[B][E][F][M][I][0-
9][0-9][0-9][0-9]" />
                                            </xs:restriction>
                                          </xs:simpleType>
                                        </xs:attribute>
                                      </xs:extension>
```

```
                                    </xs:complexContent>
                                  </xs:complexType>
                                </xs:element>
                                <xs:element name="ApplianceDemandResponse"
minOccurs="0">
                                  <xs:complexType>
                                    <xs:sequence>
                                      <xs:element
name="ControllerableLoadFactor_Weekday" type="xs:float" />
                                      <xs:element
name="ControllerableLoadFactor_Weekend" type="xs:float" />
                                    </xs:sequence>
                                    <xs:attribute name="Key">
                                      <xs:simpleType>
                                        <xs:restriction base="xs:string">
                                          <xs:pattern value="[B][E][F][M][I][0-9][0-
9][0-9][0-9]" />
                                        </xs:restriction>
                                      </xs:simpleType>
                                    </xs:attribute>
                                  </xs:complexType>
                                </xs:element>
                              </xs:sequence>
                              <xs:attribute name="Key">
                                <xs:simpleType>
                                  <xs:restriction base="xs:string">
                                    <xs:pattern value="[B][E][F][M][0-9][0-9][0-9][0-
9]" />
                                  </xs:restriction>
                                </xs:simpleType>
                              </xs:attribute>
                            </xs:complexType>
                          </xs:element>
                        </xs:sequence>
                      </xs:complexType>
                    </xs:element>
                    <xs:element name="Producer">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element name="PV" minOccurs="0"
maxOccurs="unbounded">
                            <xs:complexType>
                              <xs:sequence>
                                <xs:element name="TiltAngle" minOccurs="0">
                                  <xs:complexType>
                                    <xs:complexContent>
                                      <xs:extension
base="OptimalControlConstraintInput">
                                        <xs:attribute name="Key">
                                          <xs:simpleType>
                                            <xs:restriction base="xs:string">
```

```
                                              <xs:pattern value="[P][E][F][M][I][0-
9][0-9][0-9][0-9]" />
                                            </xs:restriction>
                                          </xs:simpleType>
                                        </xs:attribute>
                                      </xs:extension>
                                    </xs:complexContent>
                                  </xs:complexType>
                                </xs:element>
                                <xs:element name="Orientation" minOccurs="0">
                                  <xs:complexType>
                                    <xs:complexContent>
                                      <xs:extension
base="OptimalControlConstraintInput">
                                        <xs:attribute name="Key">
                                          <xs:simpleType>
                                            <xs:restriction base="xs:string">
                                              <xs:pattern value="[P][E][F][M][I][0-
9][0-9][0-9][0-9]" />
                                            </xs:restriction>
                                          </xs:simpleType>
                                        </xs:attribute>
                                      </xs:extension>
                                    </xs:complexContent>
                                  </xs:complexType>
                                </xs:element>
                              </xs:sequence>
                              <xs:attribute name="Key">
                                <xs:simpleType>
                                  <xs:restriction base="xs:string">
                                    <xs:pattern value="[P][E][F][M][0-9][0-9][0-9][0-
9]" />
                                  </xs:restriction>
                                </xs:simpleType>
                              </xs:attribute>
                            </xs:complexType>
                          </xs:element>
                          <xs:element name="SolarHotWater" minOccurs="0"
maxOccurs="unbounded">
                            <xs:complexType>
                              <xs:sequence>
                                <xs:element name="TiltAngle" minOccurs="0">
                                  <xs:complexType>
                                    <xs:complexContent>
                                      <xs:extension
base="OptimalControlConstraintInput">
                                        <xs:attribute name="Key">
                                          <xs:simpleType>
                                            <xs:restriction base="xs:string">
                                              <xs:pattern value="[S][E][F][M][I][0-
9][0-9][0-9][0-9]" />
```

```
                                            </xs:restriction>
                                          </xs:simpleType>
                                        </xs:attribute>
                                      </xs:extension>
                                    </xs:complexContent>
                                  </xs:complexType>
                                </xs:element>
                                <xs:element name="Orientation" minOccurs="0">
                                  <xs:complexType>
                                    <xs:complexContent>
                                      <xs:extension
base="OptimalControlConstraintInput">
                                        <xs:attribute name="Key">
                                          <xs:simpleType>
                                            <xs:restriction base="xs:string">
                                              <xs:pattern value="[S][E][F][M][I][0-
9][0-9][0-9][0-9]" />
                                            </xs:restriction>
                                          </xs:simpleType>
                                        </xs:attribute>
                                      </xs:extension>
                                    </xs:complexContent>
                                  </xs:complexType>
                                </xs:element>
                              </xs:sequence>
                              <xs:attribute name="Key">
                                <xs:simpleType>
                                  <xs:restriction base="xs:string">
                                    <xs:pattern value="[S][E][F][M][0-9][0-9][0-9][0-
9]" />
                                  </xs:restriction>
                                </xs:simpleType>
                              </xs:attribute>
                            </xs:complexType>
                          </xs:element>
                        </xs:sequence>
                      </xs:complexType>
                    </xs:element>
                  </xs:sequence>
                  <xs:attribute name="ID" fixed="L4000" type="xs:string" />
                </xs:complexType>
              </xs:element>
              <xs:element name="DesignOptimization">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="Consumer">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element name="Building" minOccurs="0"
maxOccurs="unbounded">
                            <xs:complexType>
```

```
<xs:sequence>
  <xs:element name="WallInsulationThickness"
minOccurs="0">
    <xs:complexType>
      <xs:complexContent>
        <xs:extension base="DOInput">
          <xs:attribute name="Key">
            <xs:simpleType>
              <xs:restriction base="xs:string">
                <xs:pattern value="[B][E][E][M][I][0-
9][0-9][0-9][0-9]" />

              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
  </xs:element>
  <xs:element name="InfiltrationRate" minOccurs="0">
    <xs:complexType>
      <xs:complexContent>
        <xs:extension base="DOInput">
          <xs:attribute name="Key">
            <xs:simpleType>
              <xs:restriction base="xs:string">
                <xs:pattern value="[B][E][E][M][I][0-
9][0-9][0-9][0-9]" />

              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
  </xs:element>
  <xs:element name="RoofEmissivity" minOccurs="0">
    <xs:complexType>
      <xs:complexContent>
        <xs:extension base="DOInput">
          <xs:attribute name="Key">
            <xs:simpleType>
              <xs:restriction base="xs:string">
                <xs:pattern value="[B][E][E][M][I][0-
9][0-9][0-9][0-9]" />

              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
  </xs:element>
  <xs:element name="WindowSHGC" minOccurs="0">
```

```
    <xs:complexType>
       <xs:complexContent>
          <xs:extension base="DOInput">
             <xs:attribute name="Key">
                <xs:simpleType>
                   <xs:restriction base="xs:string">
                      <xs:pattern value="[B][E][E][M][I][0-
9][0-9][0-9][0-9]" />

                   </xs:restriction>
                </xs:simpleType>
             </xs:attribute>
          </xs:extension>
       </xs:complexContent>
    </xs:complexType>
 </xs:element>
 <xs:element name="WindowOverhangAngle"
minOccurs="0">
    <xs:complexType>
       <xs:complexContent>
          <xs:extension base="DOInput">
             <xs:attribute name="Key">
                <xs:simpleType>
                   <xs:restriction base="xs:string">
                      <xs:pattern value="[B][E][E][M][I][0-
9][0-9][0-9][0-9]" />

                   </xs:restriction>
                </xs:simpleType>
             </xs:attribute>
          </xs:extension>
       </xs:complexContent>
    </xs:complexType>
 </xs:element>
 <xs:element name="WindowFinAngle" minOccurs="0">
    <xs:complexType>
       <xs:complexContent>
          <xs:extension base="DOInput">
             <xs:attribute name="Key">
                <xs:simpleType>
                   <xs:restriction base="xs:string">
                      <xs:pattern value="[B][E][E][M][I][0-
9][0-9][0-9][0-9]" />

                   </xs:restriction>
                </xs:simpleType>
             </xs:attribute>
          </xs:extension>
       </xs:complexContent>
    </xs:complexType>
 </xs:element>
 <xs:element name="NaturalVentilation" minOccurs="0">
    <xs:complexType>
       <xs:complexContent>
```

```
                                                         <xs:extension base="DOInput">
                                                            <xs:attribute name="Key">
                                                               <xs:simpleType>
                                                                  <xs:restriction base="xs:string">
                                                                     <xs:pattern value="[B][E][E][M][I][0-
9][0-9][0-9][0-9]" />
                                                                  </xs:restriction>
                                                               </xs:simpleType>
                                                            </xs:attribute>
                                                         </xs:extension>
                                                      </xs:complexContent>
                                                   </xs:complexType>
                                                </xs:element>
                                             </xs:sequence>
                                             <xs:attribute name="Key">
                                                <xs:simpleType>
                                                   <xs:restriction base="xs:string">
                                                      <xs:pattern value="[B][E][E][M][0-9][0-9][0-9][0-
9]" />
                                                   </xs:restriction>
                                                </xs:simpleType>
                                             </xs:attribute>
                                          </xs:complexType>
                                       </xs:element>
                                    </xs:sequence>
                                 </xs:complexType>
                              </xs:element>
                              <xs:element name="Producer">
                                 <xs:complexType>
                                    <xs:sequence>
                                       <xs:element name="PV" minOccurs="0"
maxOccurs="unbounded">
                                          <xs:complexType>
                                             <xs:sequence>
                                                <xs:element name="ModuleArea">
                                                   <xs:complexType>
                                                      <xs:complexContent>
                                                         <xs:extension base="DOInput">
                                                            <xs:attribute name="Key">
                                                               <xs:simpleType>
                                                                  <xs:restriction base="xs:string">
                                                                     <xs:pattern value="[P][E][E][M][I][0-
9][0-9][0-9][0-9]" />
                                                                  </xs:restriction>
                                                               </xs:simpleType>
                                                            </xs:attribute>
                                                         </xs:extension>
                                                      </xs:complexContent>
                                                   </xs:complexType>
                                                </xs:element>
                                             </xs:sequence>
```

```
                                             <xs:attribute name="Key">
                                                <xs:simpleType>
                                                   <xs:restriction base="xs:string">
                                                      <xs:pattern value="[P][E][E][M][0-9][0-9][0-9][0-
9]" />
                                                   </xs:restriction>
                                                </xs:simpleType>
                                             </xs:attribute>
                                          </xs:complexType>
                                       </xs:element>
                                       <xs:element name="SolarHotWater" minOccurs="0"
maxOccurs="unbounded">
                                          <xs:complexType>
                                             <xs:sequence>
                                                <xs:element name="CollectorArea">
                                                   <xs:complexType>
                                                      <xs:complexContent>
                                                         <xs:extension base="DOInput">
                                                            <xs:attribute name="Key">
                                                               <xs:simpleType>
                                                                  <xs:restriction base="xs:string">
                                                                     <xs:pattern value="[S][E][E][M][I][0-
9][0-9][0-9][0-9]" />
                                                                  </xs:restriction>
                                                               </xs:simpleType>
                                                            </xs:attribute>
                                                         </xs:extension>
                                                      </xs:complexContent>
                                                   </xs:complexType>
                                                </xs:element>
                                             </xs:sequence>
                                             <xs:attribute name="Key">
                                                <xs:simpleType>
                                                   <xs:restriction base="xs:string">
                                                      <xs:pattern value="[S][E][E][M][0-9][0-9][0-9][0-
9]" />
                                                   </xs:restriction>
                                                </xs:simpleType>
                                             </xs:attribute>
                                          </xs:complexType>
                                       </xs:element>
                                    </xs:sequence>
                                 </xs:complexType>
                              </xs:element>
                              <xs:element name="DistrictSystem">
                                 <xs:complexType>
                                    <xs:sequence>
                                       <xs:element name="DistrictHeating" minOccurs="0"
maxOccurs="unbounded">
                                          <xs:complexType>
                                             <xs:sequence>
```

```
                                <xs:element name="HeatingCOP">
                                  <xs:complexType>
                                    <xs:complexContent>
                                      <xs:extension base="DOInput">
                                        <xs:attribute name="Key">
                                          <xs:simpleType>
                                            <xs:restriction base="xs:string">
                                              <xs:pattern value="[H][E][E][M][I][0-
9][0-9][0-9][0-9]" />
                                            </xs:restriction>
                                          </xs:simpleType>
                                        </xs:attribute>
                                      </xs:extension>
                                    </xs:complexContent>
                                  </xs:complexType>
                                </xs:element>
                              </xs:sequence>
                              <xs:attribute name="Key">
                                <xs:simpleType>
                                  <xs:restriction base="xs:string">
                                    <xs:pattern value="[H][E][E][M][0-9][0-9][0-9][0-
9]" />
                                  </xs:restriction>
                                </xs:simpleType>
                              </xs:attribute>
                            </xs:complexType>
                          </xs:element>
                          <xs:element name="DistrictCoioling" minOccurs="0"
maxOccurs="unbounded">
                            <xs:complexType>
                              <xs:sequence>
                                <xs:element name="CoolingCOP">
                                  <xs:complexType>
                                    <xs:complexContent>
                                      <xs:extension base="DOInput">
                                        <xs:attribute name="Key">
                                          <xs:simpleType>
                                            <xs:restriction base="xs:string">
                                              <xs:pattern value="[C][E][E][M][I][0-
9][0-9][0-9][0-9]" />
                                            </xs:restriction>
                                          </xs:simpleType>
                                        </xs:attribute>
                                      </xs:extension>
                                    </xs:complexContent>
                                  </xs:complexType>
                                </xs:element>
                              </xs:sequence>
                              <xs:attribute name="Key">
                                <xs:simpleType>
                                  <xs:restriction base="xs:string">
```

```
                                                      <xs:pattern value="[C][E][E][M][0-9][0-9][0-9][0-
9]" />
                                                   </xs:restriction>
                                                </xs:simpleType>
                                             </xs:attribute>
                                          </xs:complexType>
                                       </xs:element>
                                       <xs:element name="Cogeneration" minOccurs="0"
maxOccurs="unbounded">
                                          <xs:complexType>
                                             <xs:sequence>
                                                <xs:element name="HeatingCOP">
                                                   <xs:complexType>
                                                      <xs:complexContent>
                                                         <xs:extension base="DOInput">
                                                            <xs:attribute name="Key">
                                                               <xs:simpleType>
                                                                  <xs:restriction base="xs:string">
                                                                     <xs:pattern value="[G][E][E][M][I][0-
9][0-9][0-9][0-9]" />
                                                                  </xs:restriction>
                                                               </xs:simpleType>
                                                            </xs:attribute>
                                                         </xs:extension>
                                                      </xs:complexContent>
                                                   </xs:complexType>
                                                </xs:element>
                                             </xs:sequence>
                                             <xs:attribute name="Key">
                                                <xs:simpleType>
                                                   <xs:restriction base="xs:string">
                                                      <xs:pattern value="[G][E][E][M][0-9][0-9][0-9][0-
9]" />
                                                   </xs:restriction>
                                                </xs:simpleType>
                                             </xs:attribute>
                                          </xs:complexType>
                                       </xs:element>
                                    </xs:sequence>
                                 </xs:complexType>
                              </xs:element>
                              <xs:element name="Storage">
                                 <xs:complexType>
                                    <xs:sequence>
                                       <xs:element name="ElectricityBattery" minOccurs="0"
maxOccurs="unbounded">
                                          <xs:complexType>
                                             <xs:sequence>
                                                <xs:element name="Capacity">
                                                   <xs:complexType>
                                                      <xs:complexContent>
```

```
                                      <xs:extension base="DOInput">
                                        <xs:attribute name="Key">
                                          <xs:simpleType>
                                            <xs:restriction base="xs:string">
                                              <xs:pattern value="[E][E][E][M][I][0-
9][0-9][0-9][0-9]" />
                                            </xs:restriction>
                                          </xs:simpleType>
                                        </xs:attribute>
                                      </xs:extension>
                                    </xs:complexContent>
                                  </xs:complexType>
                                </xs:element>
                              </xs:sequence>
                              <xs:attribute name="Key">
                                <xs:simpleType>
                                  <xs:restriction base="xs:string">
                                    <xs:pattern value="[E][E][E][M][0-9][0-9][0-9][0-
9]" />
                                  </xs:restriction>
                                </xs:simpleType>
                              </xs:attribute>
                            </xs:complexType>
                          </xs:element>
                        </xs:sequence>
                      </xs:complexType>
                    </xs:element>
                  </xs:sequence>
                  <xs:attribute name="ID" fixed="L5000" type="xs:string" />
                </xs:complexType>
              </xs:element>
              <xs:element name="LogicalLayerRelationship">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="SendsSensorData">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element name="PhysicalLayerToSensor"
type="LogicalLayerRelationship" minOccurs="0" maxOccurs="unbounded" />
                        </xs:sequence>
                      </xs:complexType>
                    </xs:element>
                    <xs:element name="ProvidesSensorData">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element name="DataSender">
                            <xs:complexType>
                              <xs:attribute name="IDRef" fixed="L1000"
type="xs:IDREF" />
                            </xs:complexType>
                          </xs:element>
```

```
                          <xs:element name="DataReceiver">
                            <xs:complexType>
                              <xs:attribute name="IDRef" fixed="L2000"
type="xs:IDREF" />
                            </xs:complexType>
                          </xs:element>
                        </xs:sequence>
                      </xs:complexType>
                    </xs:element>
                    <xs:element name="ProvidesActuatorData">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element name="DataSender">
                            <xs:complexType>
                              <xs:attribute name="IDRef" fixed="L2000"
type="xs:IDREF" />
                            </xs:complexType>
                          </xs:element>
                          <xs:element name="DataReceiver">
                            <xs:complexType>
                              <xs:attribute name="IDRef" fixed="L3000"
type="xs:IDREF" />
                            </xs:complexType>
                          </xs:element>
                        </xs:sequence>
                      </xs:complexType>
                    </xs:element>
                    <xs:element name="ProvidesVariableData">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element name="PhysicalLayerToOptimalControl"
type="LogicalLayerRelationship" minOccurs="0" maxOccurs="unbounded" />
                          <xs:element name="PhysicalLayerToDesignOptimization"
type="LogicalLayerRelationship" minOccurs="0" maxOccurs="unbounded" />
                        </xs:sequence>
                      </xs:complexType>
                    </xs:element>
                    <xs:element name="ActuatesControl">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element name="ActuatesDiscontinuousControlSignal"
type="LogicalLayerRelationship" minOccurs="0" maxOccurs="unbounded" />
                          <xs:element name="ActuatesContinuousControlSignal"
type="LogicalLayerRelationship" minOccurs="0" maxOccurs="unbounded" />
                          <xs:element name="ConductsOptimalControl"
type="LogicalLayerRelationship" minOccurs="0" maxOccurs="unbounded" />
                        </xs:sequence>
                      </xs:complexType>
                    </xs:element>
                    <xs:element name="OptimalDesign">
                      <xs:complexType>
```

```
                        <xs:sequence>
                          <xs:element name="ConductsDesignOptimization"
type="LogicalLayerRelationship" minOccurs="0" maxOccurs="unbounded" />
                        </xs:sequence>
                      </xs:complexType>
                    </xs:element>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
    <xs:key name="BuildingKey">
      <xs:selector xpath="xs:PhysicalLayer/xs:Consumer/xs:Building" />
      <xs:field xpath="@xs:Key" />
    </xs:key>
    <xs:keyref name="BuildingKeyRef" refer="BuildingKey">
      <xs:selector
xpath=".//xs:SupplyPower/xs:EnergyReceiver|.//xs:SupplyDHW/xs:EnergyReceiver|.//xs:Supply
ChilledWater/xs:EnergyReceiver|.//xs:SupplyHotWater/xs:EnergyReceiver|.//xs:SupplySteam/xs:
EnergyReceiver|.//xs:SupplyNaturalGas/xs:EnergyReceiver|.//xs:SupplyGasoline/xs:EnergyRecei
ver|.//xs:SupplyDiesel/xs:EnergyReceiver|.//xs:SupplyPropane/xs:EnergyReceiver|.//xs:SupplyOil
/xs:EnergyReceiver|.//xs:PhysicalLayerToSensor/xs:DataSender|.//xs:ActuatesDiscontinuousCont
rolSignal/xs:DataReceiver|.//xs:ActuatesContinuousControlSignal/xs:DataReceiver|.//xs:Conduct
sDesignOptimization/xs:DataReceiver|.//xs:PhysicalLayerToOptimalControl/xs:DataSender|.//xs:
PhysicalLayerToDesignOptimization/xs:DataSender" />
      <xs:field xpath="@xs:Key" />
    </xs:keyref>
    <xs:key name="EVKey">
      <xs:selector xpath="xs:PhysicalLayer/xs:Consumer/xs:ElectricVehicleChargingStation"
/>
      <xs:field xpath="@xs:Key" />
    </xs:key>
    <xs:keyref name="EVKeyRef" refer="EVKey">
      <xs:selector xpath=".//xs:SupplyPower/xs:EnergyReceiver" />
      <xs:field xpath="@xs:Key" />
    </xs:keyref>
    <xs:key name="PVKey">
      <xs:selector xpath="xs:PhysicalLayer/xs:Producer/xs:PV" />
      <xs:field xpath="@xs:Key" />
    </xs:key>
    <xs:keyref name="PVKeyRef" refer="PVKey">
      <xs:selector xpath=".//xs:SupplyPower/xs:EnergySender |
.//xs:SellBackPower/xs:EnergySender | .//xs:PhysicalLayerToSensor/xs:EnergySender |
.//xs:ActuatesDiscontinuousControlSignal/xs:DataReceiver|.//xs:ConductsDesignOptimization/xs:
DataReceiver|.//xs:PhysicalLayerToOptimalControl/xs:DataSender|.//xs:PhysicalLayerToOptimal
Control/xs:DataSender|.//xs:PhysicalLayerToDesignOptimization/xs:DataSender" />
      <xs:field xpath="@xs:KeyRef" />
    </xs:keyref>
```

```
    <xs:key name="WindTurbineKey">
      <xs:selector xpath="xs:PhysicalLayer/xs:Producer/xs:WindTurbine" />
      <xs:field xpath="@xs:Key" />
    </xs:key>
    <xs:keyref name="WindTurbineKeyRef" refer="WindTurbineKey">
      <xs:selector
xpath=".//xs:SupplyPower/xs:EnergySender|.//xs:SellBackPower/xs:EnergySender" />
      <xs:field xpath="@xs:KeyRef" />
    </xs:keyref>
    <xs:key name="SolarHotWaterKey">
      <xs:selector xpath="xs:PhysicalLayer/xs:Producer/xs:SolarHotWater" />
      <xs:field xpath="@xs:Key" />
    </xs:key>
    <xs:keyref name="SolarHotWaterKeyRef" refer="SolarHotWaterKey">
      <xs:selector
xpath=".//xs:SupplyDomesticHotWater/xs:EnergySender|.//xs:PhysicalLayerToSensor/xs:Energy
Sender|.//xs:ActuatesDiscontinuousControlSignal/xs:DataReceiver|.//xs:ConductsDesignOptimiza
tion/xs:DataReceiver|.//xs:PhysicalLayerToOptimalControl/xs:DataSender|.//xs:PhysicalLayerTo
DesignOptimization/xs:DataSender" />
      <xs:field xpath="@xs:KeyRef" />
    </xs:keyref>
    <xs:key name="DistrictHeatingKey">
      <xs:selector xpath="xs:PhysicalLayer/xs:DistrictSystem/xs:DistrictHeating" />
      <xs:field xpath="@xs:Key" />
    </xs:key>
    <xs:keyref name="DistrictHeatingKeyRef" refer="DistrictHeatingKey">
      <xs:selector xpath=".//xs:SupplyHotWater/xs:EnergySender
|.//xs:SupplySteam/xs:EnergySender|.//xs:PhysicalLayerToSensor/xs:EnergySender
|.//xs:ActuatesDiscontinuousControlSignal/xs:DataReceiver|.//xs:ConductsDesignOptimization/xs
:DataReceiver
             " />
      <xs:field xpath="@xs:KeyRef" />
    </xs:keyref>
    <xs:key name="DistrictCoolingKey">
      <xs:selector xpath="xs:PhysicalLayer/xs:DistrictSystem/xs:DistrictCooling" />
      <xs:field xpath="@xs:Key" />
    </xs:key>
    <xs:keyref name="DistrictCoolingKeyRef" refer="DistrictCoolingKey">
      <xs:selector
xpath=".//xs:SupplyChilledWater/xs:EnergySender|.//xs:PhysicalLayerToSensor/xs:EnergySende
r|.//xs:ActuatesDiscontinuousControlSignal/xs:DataReceiver|.//xs:ConductsDesignOptimization/x
s:DataReceiver|.//xs:PhysicalLayerToDesignOptimization/xs:DataSender" />
      <xs:field xpath="@xs:KeyRef" />
    </xs:keyref>
    <xs:key name="CogenerationKey">
      <xs:selector xpath="xs:PhysicalLayer/xs:DistrictSystem/xs:Cogeneration" />
      <xs:field xpath="@xs:Key" />
    </xs:key>
    <xs:keyref name="CogenerationKeyRef" refer="CogenerationKey">
      <xs:selector
xpath=".//xs:SupplyPower/xs:EnergySender|.//xs:SupplyHotWater/xs:EnergySender|.//xs:Supply
```

```
Steam/xs:EnergySender|.//xs:PhysicalLayerToSensor/xs:EnergySender|.//xs:ActuatesDiscontinuousControlSignal/xs:DataReceiver|.//xs:ConductsDesignOptimization/xs:DataReceiver|.//xs:PhysicalLayerToDesignOptimization/xs:DataSender" />
      <xs:field xpath="@xs:KeyRef" />
    </xs:keyref>
    <xs:key name="ElectricityBatteryKey">
      <xs:selector xpath="xs:PhysicalLayer/xs:Storage/xs:ElectricityBattery" />
      <xs:field xpath="@xs:Key" />
    </xs:key>
    <xs:keyref name="ElectricityBatteryKeyRef" refer="ElectricityBatteryKey">
      <xs:selector xpath=".//xs:SupplyPower/xs:EnergySender|.//xs:SupplyPower/xs:EnergyReceiver|.//xs:SellBackPower/xs:EnergySender|.//xs:ConductsDesignOptimization/xs:DataReceiver|.//xs:PhysicalLayerToDesignOptimization/xs:DataSender" />
      <xs:field xpath="@xs:KeyRef" />
    </xs:keyref>
    <xs:key name="ElectricityGridKey">
      <xs:selector xpath="xs:PhysicalLayer/xs:Source/xs:ElectricityGrid" />
      <xs:field xpath="@xs:Key" />
    </xs:key>
    <xs:key name="NaturalGasKey">
      <xs:selector xpath="xs:PhysicalLayer/xs:Source/xs:NaturalGas" />
      <xs:field xpath="@xs:Key" />
    </xs:key>
    <xs:key name="GasolineKey">
      <xs:selector xpath="xs:PhysicalLayer/xs:Source/xs:Gasoline" />
      <xs:field xpath="@xs:Key" />
    </xs:key>
    <xs:key name="DieselKey">
      <xs:selector xpath="xs:PhysicalLayer/xs:Source/xs:Diesel" />
      <xs:field xpath="@xs:Key" />
    </xs:key>
    <xs:key name="PropaneKey">
      <xs:selector xpath="xs:PhysicalLayer/xs:Source/xs:Propane" />
      <xs:field xpath="@xs:Key" />
    </xs:key>
    <xs:key name="OilKey">
      <xs:selector xpath="xs:PhysicalLayer/xs:Source/xs:Oil" />
      <xs:field xpath="@xs:Key" />
    </xs:key>
    <xs:keyref name="ElectricityGridKeyRef" refer="ElectricityGridKey">
      <xs:selector xpath=".//xs:SupplyPower/xs:EnergySender|.//xs:SellBackPower/xs:EnergyReceiver" />
      <xs:field xpath="@xs:KeyRef" />
    </xs:keyref>
    <xs:keyref name="NaturalGasKeyRef" refer="NaturalGasKey">
      <xs:selector xpath=".//xs:SupplyNaturalGas/xs:EnergySender" />
      <xs:field xpath="@xs:KeyRef" />
    </xs:keyref>
    <xs:keyref name="GasolineKeyRef" refer="GasolineKey">
      <xs:selector xpath=".//xs:SupplyGasoline/xs:EnergySender" />
```

```
      <xs:field xpath="@xs:KeyRef" />
    </xs:keyref>
    <xs:keyref name="DieselKeyRef" refer="DieselKey">
      <xs:selector xpath=".//xs:SupplyDiesel/xs:EnergySender" />
      <xs:field xpath="@xs:KeyRef" />
    </xs:keyref>
    <xs:keyref name="PropaneKeyRef" refer="PropaneKey">
      <xs:selector xpath=".//xs:SupplyPropane/xs:EnergySender" />
      <xs:field xpath="@xs:KeyRef" />
    </xs:keyref>
    <xs:keyref name="OilKeyRef" refer="OilKey">
      <xs:selector xpath=".//xs:SupplyOil/xs:EnergySender" />
      <xs:field xpath="@xs:KeyRef" />
    </xs:keyref>
    <xs:key name="BuildingSensorKey">
      <xs:selector xpath=".//xs:Sensor/xs:Consumer/xs:Building" />
      <xs:field xpath="@xs:Key" />
    </xs:key>
    <xs:keyref name="BuildingSensorKeyRef" refer="BuildingSensorKey">
      <xs:selector xpath=".//xs:PhysicalLayerToSensor/xs:DataReceiver" />
      <xs:field xpath="@xs:KeyRef" />
    </xs:keyref>
    <xs:key name="EVSensorKey">
      <xs:selector xpath=".//xs:Sensor/xs:Consumer/xs:ElectricVehicleChargingStation" />
      <xs:field xpath="@xs:Key" />
    </xs:key>
    <xs:keyref name="EVSensorKeyRef" refer="EVSensorKey">
      <xs:selector xpath=".//xs:PhysicalLayerToSensor/xs:DataReceiver" />
      <xs:field xpath="@xs:KeyRef" />
    </xs:keyref>
    <xs:key name="PVSensorKey">
      <xs:selector xpath=".//xs:Sensor/xs:Producer/xs:PV" />
      <xs:field xpath="@xs:Key" />
    </xs:key>
    <xs:keyref name="PVSensorKeyRef" refer="PVSensorKey">
      <xs:selector xpath=".//xs:PhysicalLayerToSensor/xs:DataReceiver" />
      <xs:field xpath="@xs:KeyRef" />
    </xs:keyref>
    <xs:key name="WindTurbineSensorKey">
      <xs:selector xpath=".//xs:Sensor/xs:Producer/xs:WindTurbine" />
      <xs:field xpath="@xs:Key" />
    </xs:key>
    <xs:keyref name="WindTurbineSensorKeyRef" refer="WindTurbineSensorKey">
      <xs:selector xpath=".//xs:PhysicalLayerToSensor/xs:DataReceiver" />
      <xs:field xpath="@xs:KeyRef" />
    </xs:keyref>
    <xs:key name="SolarHotWaterSensorKey">
      <xs:selector xpath=".//xs:Sensor/xs:Producer/xs:SolarHotWater" />
      <xs:field xpath="@xs:Key" />
    </xs:key>
    <xs:keyref name="SolarHotWaterSensorKeyRef" refer="SolarHotWaterSensorKey">
```

```
      <xs:selector xpath=".//xs:PhysicalLayerToSensor/xs:DataReceiver" />
      <xs:field xpath="@xs:KeyRef" />
    </xs:keyref>
    <xs:key name="BuildingDiscontinuousActuatorKey">
      <xs:selector xpath=".//xs:DiscontinuousControl/xs:Consumer/xs:Building" />
      <xs:field xpath="@xs:Key" />
    </xs:key>
    <xs:keyref name="BuildingDiscontinuousControlKeyRef"
refer="BuildingDiscontinuousActuatorKey">
      <xs:selector xpath=".//xs:ActuatesDiscontinuousControlSignal/xs:DataSender" />
      <xs:field xpath="@xs:KeyRef" />
    </xs:keyref>
    <xs:key name="PVDiscontinuousActuatorKey">
      <xs:selector xpath=".//xs:DiscontinuousControl/xs:Producer/xs:PV" />
      <xs:field xpath="@xs:Key" />
    </xs:key>
    <xs:keyref name="PVDiscontinuousControlKeyRef"
refer="PVDiscontinuousActuatorKey">
      <xs:selector xpath=".//xs:ActuatesDiscontinuousControlSignal/xs:DataSender" />
      <xs:field xpath="@xs:KeyRef" />
    </xs:keyref>
    <xs:key name="SHWDiscontinuousActuatorKey">
      <xs:selector xpath=".//xs:DiscontinuousControl/xs:Producer/xs:SolarHotWater" />
      <xs:field xpath="@xs:Key" />
    </xs:key>
    <xs:keyref name="SHWDiscontinuousControlKeyRef"
refer="SHWDiscontinuousActuatorKey">
      <xs:selector xpath=".//xs:ActuatesDiscontinuousControlSignal/xs:DataSender" />
      <xs:field xpath="@xs:KeyRef" />
    </xs:keyref>
    <xs:key name="BuildingContinuousActuatorKey">
      <xs:selector xpath=".//xs:ContinuousControl/xs:Consumer/xs:Building" />
      <xs:field xpath="@xs:Key" />
    </xs:key>
    <xs:keyref name="BuildingContinuousControlKeyRef"
refer="BuildingContinuousActuatorKey">
      <xs:selector xpath=".//xs:ActuatesContinuousControlSignal/xs:DataSender" />
      <xs:field xpath="@xs:KeyRef" />
    </xs:keyref>
    <xs:key name="BuildingSensorIdentifierKey">
      <xs:selector
xpath=".//xs:Occupancy|.//xs:ThermalZoneAirTemperature|.//xs:HeatingDeliveredEnergyUse|.//x
s:CoolingDeliveredEnergyUse|.//xs:LightingDeliveredEnergyUse|.//FanDeliveredEnergyUse|.//xs
:PumpDeliveredEnergyUse|.//xs:ApplianceDeliveredEnergyUse|.//xs:DHWDeliveredEnergyUse|.
//xs:Sensor/xs:Consumer/xs:Building/xs:ElectricityUse|.//xs:Sensor/xs:Consumer/xs:Building/xs:
NaturalGasUse|.//xs:Sensor/xs:Consumer/xs:Building/xs:GasolineUse|.//xs:Sensor/xs:Consumer/
xs:Building/xs:DieselUse|.//xs:Sensor/xs:Consumer/xs:Building/xs:PropaneUse|.//xs:Sensor/xs:C
onsumer/xs:Building/xs:OilUse" />
      <xs:field xpath="@xs:Key" />
    </xs:key>
    <xs:key name="EVSensorIdentifierKey">
```

```
            <xs:selector
xpath=".//xs:Sensor/xs:Consumer/xs:ElectricVehicleChargingStation/xs:NumberOfCars|.//xs:Sen
sor/xs:Consumer/xs:ElectricVehicleChargingStation/xs:ChargingPower" />
            <xs:field xpath="@xs:Key" />
        </xs:key>
        <xs:key name="PVSensorIdentifierKey">
            <xs:selector xpath=".//xs:Sensor/xs:Producer/xs:PV/xs:ElectricityGeneration" />
            <xs:field xpath="@xs:Key" />
        </xs:key>
        <xs:key name="WindTurbineSensorIdentifierKey">
            <xs:selector xpath=".//xs:Sensor/xs:Producer/xs:WindTurbine/xs:ElectricityGeneration"
/>
            <xs:field xpath="@xs:Key" />
        </xs:key>
        <xs:key name="SolarHotWaterSensorIdentifierKey">
            <xs:selector xpath=".//xs:Sensor/xs:Producer/xs:SolarHotWater/xs:DHWGeneration" />
            <xs:field xpath="@xs:Key" />
        </xs:key>
        <xs:key name="BuildingDiscontinuousActuatorIdentifierKey">
            <xs:selector
xpath=".//xs:DiscontinuousControl/xs:Consumer/xs:Building/xs:WindowOverhangAngle|.//xs:Di
scontinuousControl/xs:Consumer/xs:Building/xs:WindowFinAngle|.//xs:DiscontinuousControl/xs
:Consumer/xs:Building/xs:NaturalVentilationOpeningRatio" />
            <xs:field xpath="@xs:Key" />
        </xs:key>
        <xs:key name="PVDiscontinuousActuatorIdentifierKey">
            <xs:selector
xpath=".//xs:DiscontinuousControl/xs:Producer/xs:PV/xs:ModuleTiltAngle|.//xs:DiscontinuousC
ontrol/xs:Producer/xs:PV/xs:ModuleOrientation" />
            <xs:field xpath="@xs:Key" />
        </xs:key>
        <xs:key name="SHWDiscontinuousActuatorIdentifierKey">
            <xs:selector
xpath=".//xs:DiscontinuousControl/xs:Producer/xs:SolarHotWater/xs:CollectorTiltAngle|.//xs:Dis
continuousControl/xs:Producer/xs:SolarHotWater/xs:CollectorOrientation" />
            <xs:field xpath="@xs:Key" />
        </xs:key>
        <xs:key name="BuildingContinuousActuatorIdentifierKey">
            <xs:selector
xpath=".//xs:ContinuousControl/xs:Consumer/xs:Building/xs:AppliancePowerDensity|.//xs:Conti
nuousControl/xs:Consumer/xs:Building/xs:LightingPowerDensity|.//xs:ContinuousControl/xs:Co
nsumer/xs:Building/xs:HeatingSetPoint|.//xs:ContinuousControl/xs:Consumer/xs:Building/xs:Co
olingSetPoint" />
            <xs:field xpath="@xs:Key" />
        </xs:key>
        <xs:key name="BuildingOptimalControlIdentifierKey">
            <xs:selector xpath=".//xs:OptimalControl/xs:Consumer/xs:Building" />
            <xs:field xpath="@xs:Key" />
        </xs:key>
        <xs:keyref name="BuildingOptimalControlIdentifierKeyRef"
refer="BuildingOptimalControlIdentifierKey">
```

```
      <xs:selector
xpath=".//xs:ConductsOptimalControl/xs:DataSender|.//xs:PhysicalLayerToOptimalControl/xs:D
ataReceiver" />
      <xs:field xpath="@xs:KeyRef" />
    </xs:keyref>
    <xs:key name="PVOptimalControlIdentifierKey">
      <xs:selector xpath=".//xs:OptimalControl/xs:Consumer/xs:PV" />
      <xs:field xpath="@xs:Key" />
    </xs:key>
    <xs:keyref name="PVOptimalControlIdentifierKeyRef"
refer="PVOptimalControlIdentifierKey">
      <xs:selector
xpath=".//xs:ConductsOptimalControl/xs:DataSender|.//xs:PhysicalLayerToOptimalControl/xs:D
ataReceiver" />
      <xs:field xpath="@xs:KeyRef" />
    </xs:keyref>
    <xs:key name="SHWOptimalControlIdentifierKey">
      <xs:selector xpath=".//xs:OptimalControl/xs:Consumer/xs:SHW" />
      <xs:field xpath="@xs:Key" />
    </xs:key>
    <xs:keyref name="SHWOptimalControlIdentifierKeyRef"
refer="SHWOptimalControlIdentifierKey">
      <xs:selector
xpath=".//xs:ConductsOptimalControl/xs:DataSender|.//xs:PhysicalLayerToOptimalControl/xs:D
ataReceiver" />
      <xs:field xpath="@xs:KeyRef" />
    </xs:keyref>
    <xs:key name="BuildingDesignOptimizationIdentifierKey">
      <xs:selector xpath=".//xs:DesignOptimization/xs:Consumer/xs:Building" />
      <xs:field xpath="@xs:Key" />
    </xs:key>
    <xs:keyref name="BuildingDesignOptimizationIdentifierKeyRef"
refer="BuildingDesignOptimizationIdentifierKey">
      <xs:selector
xpath=".//xs:ConductsDesignOptimization/xs:DataSender|.//xs:PhysicalLayerToDesignOptimizat
ion/xs:DataReceiver" />
      <xs:field xpath="@xs:KeyRef" />
    </xs:keyref>
    <xs:key name="PVDesignOptimizationIdentifierKey">
      <xs:selector xpath=".//xs:DesignOptimization/xs:Producer/xs:PV" />
      <xs:field xpath="@xs:Key" />
    </xs:key>
    <xs:keyref name="PVDesignOptimizationIdentifierKeyRef"
refer="PVDesignOptimizationIdentifierKey">
      <xs:selector
xpath=".//xs:ConductsDesignOptimization/xs:DataSender|.//xs:PhysicalLayerToDesignOptimizat
ion/xs:DataReceiver" />
      <xs:field xpath="@xs:KeyRef" />
    </xs:keyref>
    <xs:key name="SHWDesignOptimizationIdentifierKey">
      <xs:selector xpath=".//xs:DesignOptimization/xs:Producer/xs:SHW" />
```

```
            <xs:field xpath="@xs:Key" />
        </xs:key>
        <xs:keyref name="SHWDesignOptimizationIdentifierKeyRef"
refer="SHWDesignOptimizationIdentifierKey">
            <xs:selector
xpath=".//xs:ConductsDesignOptimization/xs:DataSender|.//xs:PhysicalLayerToDesignOptimizat
ion/xs:DataReceiver" />
            <xs:field xpath="@xs:KeyRef" />
        </xs:keyref>
        <xs:key name="DistrictHeatingDesignOptimizationIdentifierKey">
            <xs:selector xpath=".//xs:DesignOptimization/xs:DistrictSystem/xs:DistrictHeating" />
            <xs:field xpath="@xs:Key" />
        </xs:key>
        <xs:keyref name="DistrictHeatingDesignOptimizationIdentifierKeyRef"
refer="DistrictHeatingDesignOptimizationIdentifierKey">
            <xs:selector
xpath=".//xs:ConductsDesignOptimization/xs:DataSender|.//xs:PhysicalLayerToDesignOptimizat
ion/xs:DataReceiver" />
            <xs:field xpath="@xs:KeyRef" />
        </xs:keyref>
        <xs:key name="DistrictCoolingDesignOptimizationIdentifierKey">
            <xs:selector xpath=".//xs:ConductsDesignOptimization/xs:DataSender" />
            <xs:field xpath="@xs:Key" />
        </xs:key>
        <xs:keyref name="DistrictCoolingDesignOptimizationIdentifierKeyRef"
refer="DistrictCoolingDesignOptimizationIdentifierKey">
            <xs:selector
xpath=".//xs:PhysicalLayerToDesignOptimization/xs:DataSender|.//xs:PhysicalLayerToDesignO
ptimization/xs:DataReceiver" />
            <xs:field xpath="@xs:KeyRef" />
        </xs:keyref>
        <xs:key name="CogenerationDesignOptimizationIdentifierKey">
            <xs:selector xpath=".//xs:ConductsDesignOptimization/xs:DataSender" />
            <xs:field xpath="@xs:Key" />
        </xs:key>
        <xs:keyref name="CogenerationDesignOptimizationIdentifierKeyRef"
refer="CogenerationDesignOptimizationIdentifierKey">
            <xs:selector
xpath=".//xs:PhysicalLayerToDesignOptimization/xs:DataSender|.//xs:PhysicalLayerToDesignO
ptimization/xs:DataReceiver" />
            <xs:field xpath="@xs:KeyRef" />
        </xs:keyref>
        <xs:key name="ElectricityBatteryDesignOptimizationIdentifierKey">
            <xs:selector xpath=".//xs:DesignOptimization/xs:Storage/xs:ElectricityBattery" />
            <xs:field xpath="@xs:Key" />
        </xs:key>
        <xs:keyref name="ElectricityBatteryDesignOptimizationIdentifierKeyRef"
refer="ElectricityBatteryDesignOptimizationIdentifierKey">
            <xs:selector
xpath=".//xs:ConductsDesignOptimization/xs:DataSender|.//xs:PhysicalLayerToDesignOptimizat
ion/xs:DataReceiver" />
```

```
      <xs:field xpath="@xs:KeyRef" />
    </xs:keyref>
  </xs:element>

</xs:schema>
```

D.2 UP Instance of Setting 2

The below code shows the generated UP instance for the Logic(a) with the capacity of the electricity battery 200 kWh subcase in setting 2.

```
<?xml version='1.0' encoding='UTF-8'?>
<UPSchema xmlns:xs="http://www.w3.org/2001/XMLSchema">
 <PhysicalLayer>
  <Consumer>
   <Building Key="B9677">
    <Name>Caddell</Name>
   </Building>
   <Building Key="B4237">
    <Name>Mason</Name>
   </Building>
   <Building Key="B9455">
    <Name>AllenSustain</Name>
   </Building>
   <Building Key="B3215">
    <Name>Chapin</Name>
   </Building>
   <Building Key="B5958">
    <Name>Crecine</Name>
   </Building>
   <Building Key="B1994">
    <Name>Ford</Name>
   </Building>
   <Building Key="B4388">
    <Name>MSE</Name>
   </Building>
   <Building Key="B2526">
    <Name>Klaus</Name>
   </Building>
   <Building Key="B4936">
    <Name>Savant</Name>
   </Building>
   <Building Key="B7632">
    <Name>Smithgall</Name>
   </Building>
   <ElectricVehicleChargingStation Key="EV8264">
```

```
      <Name>Area4</Name>
    </ElectricVehicleChargingStation>
    <ElectricVehicleChargingStation Key="EV6010">
      <Name>W23Deck</Name>
    </ElectricVehicleChargingStation>
  </Consumer>
  <Producer>
    <PV Key="P2708">
      <Name>PV8000</Name>
    </PV>
  </Producer>
  <DistrictSystem>
    <DistrictHeating Key="DH1309">
      <Name>GT_DH</Name>
    </DistrictHeating>
    <DistrictCooling Key="DC3302">
      <Name>GT_DC</Name>
    </DistrictCooling>
  </DistrictSystem>
  <Storage>
    <ElectricityBattery Key="EB5139">
      <Name>PowerPack200</Name>
    </ElectricityBattery>
  </Storage>
  <Source>
    <ElectricityGrid Key="SE1000"/>
    <NaturalGas Key="SN1000"/>
  </Source>
  <PhysicalLayerRelationship>
    <SupplyPower>
      <EnergySender KeyRef="SE1000"/>
      <EnergyReceiver KeyRef="B9677"/>
    </SupplyPower>
    <SupplyPower>
      <EnergySender KeyRef="SE1000"/>
      <EnergyReceiver KeyRef="B4237"/>
    </SupplyPower>
    <SupplyPower>
      <EnergySender KeyRef="SE1000"/>
      <EnergyReceiver KeyRef="B9455"/>
    </SupplyPower>
    <SupplyPower>
      <EnergySender KeyRef="SE1000"/>
      <EnergyReceiver KeyRef="B3215"/>
    </SupplyPower>
    <SupplyPower>
      <EnergySender KeyRef="SE1000"/>
      <EnergyReceiver KeyRef="B5958"/>
    </SupplyPower>
    <SupplyPower>
      <EnergySender KeyRef="SE1000"/>
```

```
     <EnergyReceiver KeyRef="B1994"/>
    </SupplyPower>
    <SupplyPower>
     <EnergySender KeyRef="SE1000"/>
     <EnergyReceiver KeyRef="B4388"/>
    </SupplyPower>
    <SupplyPower>
     <EnergySender KeyRef="SE1000"/>
     <EnergyReceiver KeyRef="B2526"/>
    </SupplyPower>
    <SupplyPower>
     <EnergySender KeyRef="SE1000"/>
     <EnergyReceiver KeyRef="B4936"/>
    </SupplyPower>
    <SupplyPower>
     <EnergySender KeyRef="SE1000"/>
     <EnergyReceiver KeyRef="B7632"/>
    </SupplyPower>
    <SupplyPower>
     <EnergySender KeyRef="SE1000"/>
     <EnergyReceiver KeyRef="EV8264"/>
    </SupplyPower>
    <SupplyPower>
     <EnergySender KeyRef="SE1000"/>
     <EnergyReceiver KeyRef="EV6010"/>
    </SupplyPower>
    <SupplyPower>
     <EnergySender KeyRef="SE1000"/>
     <EnergyReceiver KeyRef="DC3302"/>
    </SupplyPower>
    <SupplyPower>
     <EnergySender KeyRef="P2708"/>
     <EnergyReceiver KeyRef="B9677"/>
    </SupplyPower>
    <SupplyPower>
     <EnergySender KeyRef="P2708"/>
     <EnergyReceiver KeyRef="B4237"/>
    </SupplyPower>
    <SupplyPower>
     <EnergySender KeyRef="P2708"/>
     <EnergyReceiver KeyRef="B9455"/>
    </SupplyPower>
    <SupplyPower>
     <EnergySender KeyRef="P2708"/>
     <EnergyReceiver KeyRef="B3215"/>
    </SupplyPower>
    <SupplyPower>
     <EnergySender KeyRef="P2708"/>
     <EnergyReceiver KeyRef="B5958"/>
    </SupplyPower>
    <SupplyPower>
```

```xml
      <EnergySender KeyRef="P2708"/>
      <EnergyReceiver KeyRef="B1994"/>
    </SupplyPower>
    <SupplyPower>
      <EnergySender KeyRef="P2708"/>
      <EnergyReceiver KeyRef="B4388"/>
    </SupplyPower>
    <SupplyPower>
      <EnergySender KeyRef="P2708"/>
      <EnergyReceiver KeyRef="B2526"/>
    </SupplyPower>
    <SupplyPower>
      <EnergySender KeyRef="P2708"/>
      <EnergyReceiver KeyRef="B4936"/>
    </SupplyPower>
    <SupplyPower>
      <EnergySender KeyRef="P2708"/>
      <EnergyReceiver KeyRef="B7632"/>
    </SupplyPower>
    <SupplyPower>
      <EnergySender KeyRef="P2708"/>
      <EnergyReceiver KeyRef="EV8264"/>
    </SupplyPower>
    <SupplyPower>
      <EnergySender KeyRef="P2708"/>
      <EnergyReceiver KeyRef="EV6010"/>
    </SupplyPower>
    <SupplyPower>
      <EnergySender KeyRef="P2708"/>
      <EnergyReceiver KeyRef="EB5139"/>
    </SupplyPower>
    <SupplyPower>
      <EnergySender KeyRef="EB5139"/>
      <EnergyReceiver KeyRef="B9677"/>
    </SupplyPower>
    <SupplyPower>
      <EnergySender KeyRef="EB5139"/>
      <EnergyReceiver KeyRef="B4237"/>
    </SupplyPower>
    <SupplyPower>
      <EnergySender KeyRef="EB5139"/>
      <EnergyReceiver KeyRef="B9455"/>
    </SupplyPower>
    <SupplyPower>
      <EnergySender KeyRef="EB5139"/>
      <EnergyReceiver KeyRef="B3215"/>
    </SupplyPower>
    <SupplyPower>
      <EnergySender KeyRef="EB5139"/>
      <EnergyReceiver KeyRef="B5958"/>
    </SupplyPower>
```

```
    <SupplyPower>
     <EnergySender KeyRef="EB5139"/>
     <EnergyReceiver KeyRef="B1994"/>
    </SupplyPower>
    <SupplyPower>
     <EnergySender KeyRef="EB5139"/>
     <EnergyReceiver KeyRef="B4388"/>
    </SupplyPower>
    <SupplyPower>
     <EnergySender KeyRef="EB5139"/>
     <EnergyReceiver KeyRef="B2526"/>
    </SupplyPower>
    <SupplyPower>
     <EnergySender KeyRef="EB5139"/>
     <EnergyReceiver KeyRef="B4936"/>
    </SupplyPower>
    <SupplyPower>
     <EnergySender KeyRef="EB5139"/>
     <EnergyReceiver KeyRef="B7632"/>
    </SupplyPower>
    <SupplyPower>
     <EnergySender KeyRef="EB5139"/>
     <EnergyReceiver KeyRef="EV8264"/>
    </SupplyPower>
    <SupplyPower>
     <EnergySender KeyRef="EB5139"/>
     <EnergyReceiver KeyRef="EV6010"/>
    </SupplyPower>
    <SupplyChilledWater>
     <EnergySender KeyRef="DC3302"/>
     <EnergyReceiver KeyRef="B4237"/>
    </SupplyChilledWater>
    <SupplyChilledWater>
     <EnergySender KeyRef="DC3302"/>
     <EnergyReceiver KeyRef="B9455"/>
    </SupplyChilledWater>
    <SupplyChilledWater>
     <EnergySender KeyRef="DC3302"/>
     <EnergyReceiver KeyRef="B3215"/>
    </SupplyChilledWater>
    <SupplyChilledWater>
     <EnergySender KeyRef="DC3302"/>
     <EnergyReceiver KeyRef="B5958"/>
    </SupplyChilledWater>
    <SupplyChilledWater>
     <EnergySender KeyRef="DC3302"/>
     <EnergyReceiver KeyRef="B1994"/>
    </SupplyChilledWater>
    <SupplyChilledWater>
     <EnergySender KeyRef="DC3302"/>
     <EnergyReceiver KeyRef="B4388"/>
```

```
</SupplyChilledWater>
<SupplyChilledWater>
  <EnergySender KeyRef="DC3302"/>
  <EnergyReceiver KeyRef="B2526"/>
</SupplyChilledWater>
<SupplyChilledWater>
  <EnergySender KeyRef="DC3302"/>
  <EnergyReceiver KeyRef="B4936"/>
</SupplyChilledWater>
<SupplyChilledWater>
  <EnergySender KeyRef="DC3302"/>
  <EnergyReceiver KeyRef="B7632"/>
</SupplyChilledWater>
<SupplySteam>
  <EnergySender KeyRef="DH1309"/>
  <EnergyReceiver KeyRef="B4237"/>
</SupplySteam>
<SupplySteam>
  <EnergySender KeyRef="DH1309"/>
  <EnergyReceiver KeyRef="B9455"/>
</SupplySteam>
<SupplySteam>
  <EnergySender KeyRef="DH1309"/>
  <EnergyReceiver KeyRef="B3215"/>
</SupplySteam>
<SupplySteam>
  <EnergySender KeyRef="DH1309"/>
  <EnergyReceiver KeyRef="B5958"/>
</SupplySteam>
<SupplySteam>
  <EnergySender KeyRef="DH1309"/>
  <EnergyReceiver KeyRef="B1994"/>
</SupplySteam>
<SupplySteam>
  <EnergySender KeyRef="DH1309"/>
  <EnergyReceiver KeyRef="B4388"/>
</SupplySteam>
<SupplySteam>
  <EnergySender KeyRef="DH1309"/>
  <EnergyReceiver KeyRef="B2526"/>
</SupplySteam>
<SupplySteam>
  <EnergySender KeyRef="DH1309"/>
  <EnergyReceiver KeyRef="B4936"/>
</SupplySteam>
<SupplySteam>
  <EnergySender KeyRef="DH1309"/>
  <EnergyReceiver KeyRef="B7632"/>
</SupplySteam>
<SupplyNaturalGas>
  <EnergySender KeyRef="SN1000"/>
```

```xml
    <EnergyReceiver KeyRef="DH1309"/>
   </SupplyNaturalGas>
   <SupplyNaturalGas>
    <EnergySender KeyRef="SN1000"/>
    <EnergyReceiver KeyRef="B4237"/>
   </SupplyNaturalGas>
   <SupplyNaturalGas>
    <EnergySender KeyRef="SN1000"/>
    <EnergyReceiver KeyRef="B9455"/>
   </SupplyNaturalGas>
   <SupplyNaturalGas>
    <EnergySender KeyRef="SN1000"/>
    <EnergyReceiver KeyRef="B4388"/>
   </SupplyNaturalGas>
   <SupplyNaturalGas>
    <EnergySender KeyRef="SN1000"/>
    <EnergyReceiver KeyRef="B3215"/>
   </SupplyNaturalGas>
  </PhysicalLayerRelationship>
 </PhysicalLayer>
 <LogicalLayer>
  <Sensor ID="L1000">
   <Time>
    <CumulativeHour>1</CumulativeHour>
    <Hour>1</Hour>
    <DayType>0</DayType>
   </Time>
   <PriceSignal/>
   <Weather>
    <DryBulbTemperature ID="LW1000" TimeStep="1">0</DryBulbTemperature>
   </Weather>
   <Consumer>
    <Building Key="BS2784">
     <ElectricityUse Key="BSI1010" TimeStep="1">0</ElectricityUse>
    </Building>
    <Building Key="BS6307">
     <ElectricityUse Key="BSI1002" TimeStep="1">0</ElectricityUse>
    </Building>
    <Building Key="BS3512">
     <ElectricityUse Key="BSI1006" TimeStep="1">0</ElectricityUse>
    </Building>
    <Building Key="BS9436">
     <ElectricityUse Key="BSI1004" TimeStep="1">0</ElectricityUse>
    </Building>
    <Building Key="BS4331">
     <ElectricityUse Key="BSI1007" TimeStep="1">0</ElectricityUse>
    </Building>
    <Building Key="BS3103">
     <ElectricityUse Key="BSI1008" TimeStep="1">0</ElectricityUse>
    </Building>
    <Building Key="BS2432">
```

```
     <ElectricityUse Key="BSI1009" TimeStep="1">0</ElectricityUse>
    </Building>
    <Building Key="BS7963">
     <ElectricityUse Key="BSI1005" TimeStep="1">0</ElectricityUse>
    </Building>
    <Building Key="BS3130">
     <ElectricityUse Key="BSI1001" TimeStep="1">0</ElectricityUse>
    </Building>
    <Building Key="BS9291">
     <ElectricityUse Key="BSI1011" TimeStep="1">0</ElectricityUse>
    </Building>
   </Consumer>
   <Producer/>
   <DistrictSystem/>
   <Storage/>
  </Sensor>
  <Controller ID="L2000">
   <Condition>if 3623 &lt; CumulativeHour &lt; 6552 and 12 &lt;= Hour &lt;= 18 and
DayType == 1 and Caddell ElectricityUse &gt; 35</Condition>
   <Statement>Caddell CoolingSetPoint = 27;</Statement>
   <Condition>else</Condition>
   <Statement>Caddell CoolingSetPoint = Original;</Statement>
   <Condition>if 3623 &lt; CumulativeHour &lt; 6552 and 12 &lt;= Hour &lt;= 18 and
DayType == 1 and Mason ElectricityUse &gt; 109</Condition>
   <Statement>Mason CoolingSetPoint = 27;</Statement>
   <Condition>else</Condition>
   <Statement>Mason CoolingSetPoint = Original;</Statement>
   <Condition>if 3623 &lt; CumulativeHour &lt; 6552 and 12 &lt;= Hour &lt;= 18 and
DayType == 1 and Chapin ElectricityUse &gt; 21</Condition>
   <Statement>Chapin CoolingSetPoint = 27;</Statement>
   <Condition>else</Condition>
   <Statement>Chapin CoolingSetPoint = Original;</Statement>
   <Condition>if 3623 &lt; CumulativeHour &lt; 6552 and 12 &lt;= Hour &lt;= 18 and
DayType == 1 and Crecine ElectricityUse &gt; 255</Condition>
   <Statement>Crecine CoolingSetPoint = 27;</Statement>
   <Condition>else</Condition>
   <Statement>Crecine CoolingSetPoint = Original;</Statement>
   <Condition>if 3623 &lt; CumulativeHour &lt; 6552 and 12 &lt;= Hour &lt;= 18 and
DayType == 1 and Ford ElectricityUse &gt; 1074</Condition>
   <Statement>Ford CoolingSetPoint = 27;</Statement>
   <Condition>else</Condition>
   <Statement>Ford CoolingSetPoint = Original;</Statement>
   <Condition>if 3623 &lt; CumulativeHour &lt; 6552 and 12 &lt;= Hour &lt;= 18 and
DayType == 1 and MSE ElectricityUse &gt; 405</Condition>
   <Statement>MSE CoolingSetPoint = 27;</Statement>
   <Condition>else</Condition>
   <Statement>MSE CoolingSetPoint = Original;</Statement>
   <Condition>if 3623 &lt; CumulativeHour &lt; 6552 and 12 &lt;= Hour &lt;= 18 and
DayType == 1 and Klaus ElectricityUse &gt; 775</Condition>
   <Statement>Klaus CoolingSetPoint = 27;</Statement>
   <Condition>else</Condition>
```

```
   <Statement>Klaus CoolingSetPoint = Original;</Statement>
   <Condition>if 3623 &lt; CumulativeHour &lt; 6552 and 12 &lt;= Hour &lt;= 18 and
DayType == 1 and Savant ElectricityUse &gt; 48</Condition>
   <Statement>Savant CoolingSetPoint = 27;</Statement>
   <Condition>else</Condition>
   <Statement>Savant CoolingSetPoint = Original;</Statement>
   <Condition>if 3623 &lt; CumulativeHour &lt; 6552 and 12 &lt;= Hour &lt;= 18 and
DayType == 1 and Smithgall ElectricityUse &gt; 79</Condition>
   <Statement>Smithgall CoolingSetPoint = 27;</Statement>
   <Condition>else</Condition>
   <Statement>Smithgall CoolingSetPoint = Original;</Statement>
   <Condition>if 3623 &lt; CumulativeHour &lt; 6552 and 12 &lt;= Hour &lt;= 18 and
DayType == 1 and AllenSustain ElectricityUse &gt; 62</Condition>
   <Statement>AllenSustain CoolingSetPoint = 27;</Statement>
   <Condition>else</Condition>
   <Statement>AllenSustain CoolingSetPoint = Original;</Statement>
  </Controller>
  <Actuator ID="L3000">
   <DiscontinuousControl>
    <Consumer/>
    <Producer/>
   </DiscontinuousControl>
   <ContinuousControl>
    <Consumer>
     <Building Key="BC4610">
      <CoolingSetPoint Key="BAI1010">0</CoolingSetPoint>
     </Building>
     <Building Key="BC7021">
      <CoolingSetPoint Key="BAI1004">0</CoolingSetPoint>
     </Building>
     <Building Key="BC2612">
      <CoolingSetPoint Key="BAI1003">0</CoolingSetPoint>
     </Building>
     <Building Key="BC9916">
      <CoolingSetPoint Key="BAI1002">0</CoolingSetPoint>
     </Building>
     <Building Key="BC3057">
      <CoolingSetPoint Key="BAI1005">0</CoolingSetPoint>
     </Building>
     <Building Key="BC9497">
      <CoolingSetPoint Key="BAI1007">0</CoolingSetPoint>
     </Building>
     <Building Key="BC1415">
      <CoolingSetPoint Key="BAI1008">0</CoolingSetPoint>
     </Building>
     <Building Key="BC3625">
      <CoolingSetPoint Key="BAI1006">0</CoolingSetPoint>
     </Building>
     <Building Key="BC3141">
      <CoolingSetPoint Key="BAI1009">0</CoolingSetPoint>
     </Building>
```

```
     <Building Key="BC7273">
      <CoolingSetPoint Key="BAI1011">0</CoolingSetPoint>
     </Building>
    </Consumer>
   </ContinuousControl>
  </Actuator>
  <OptimalControl ID="L4000">
   <Consumer/>
   <Producer/>
  </OptimalControl>
  <DesignOptimization ID="L5000">
   <Consumer/>
   <Producer/>
   <DistrictSystem/>
   <Storage/>
  </DesignOptimization>
  <LogicalLayerRelationship>
   <SendsSensorData>
    <PhysicalLayerToSensor>
     <DataSender KeyRef="B9677"/>
     <DataReceiver KeyRef="BS2784"/>
    </PhysicalLayerToSensor>
    <PhysicalLayerToSensor>
     <DataSender KeyRef="B4237"/>
     <DataReceiver KeyRef="BS6307"/>
    </PhysicalLayerToSensor>
    <PhysicalLayerToSensor>
     <DataSender KeyRef="B9455"/>
     <DataReceiver KeyRef="BS3512"/>
    </PhysicalLayerToSensor>
    <PhysicalLayerToSensor>
     <DataSender KeyRef="B3215"/>
     <DataReceiver KeyRef="BS9436"/>
    </PhysicalLayerToSensor>
    <PhysicalLayerToSensor>
     <DataSender KeyRef="B5958"/>
     <DataReceiver KeyRef="BS4331"/>
    </PhysicalLayerToSensor>
    <PhysicalLayerToSensor>
     <DataSender KeyRef="B1994"/>
     <DataReceiver KeyRef="BS3103"/>
    </PhysicalLayerToSensor>
    <PhysicalLayerToSensor>
     <DataSender KeyRef="B4388"/>
     <DataReceiver KeyRef="BS2432"/>
    </PhysicalLayerToSensor>
    <PhysicalLayerToSensor>
     <DataSender KeyRef="B2526"/>
     <DataReceiver KeyRef="BS7963"/>
    </PhysicalLayerToSensor>
    <PhysicalLayerToSensor>
```

```
    <DataSender KeyRef="B4936"/>
    <DataReceiver KeyRef="BS3130"/>
  </PhysicalLayerToSensor>
  <PhysicalLayerToSensor>
    <DataSender KeyRef="B7632"/>
    <DataReceiver KeyRef="BS9291"/>
  </PhysicalLayerToSensor>
</SendsSensorData>
<ProvidesSensorData>
  <DataSender IDRef="L1000"/>
  <DataReceiver IDRef="L2000"/>
</ProvidesSensorData>
<ProvidesActuatorData>
  <DataSender IDRef="L2000"/>
  <DataReceiver IDRef="L3000"/>
</ProvidesActuatorData>
<ProvidesVariableData/>
<ActuatesControl>
  <ActuatesContinuousControlSignal>
    <DataSender KeyRef="BC4610"/>
    <DataReceiver KeyRef="B9677"/>
  </ActuatesContinuousControlSignal>
  <ActuatesContinuousControlSignal>
    <DataSender KeyRef="BC7021"/>
    <DataReceiver KeyRef="B4237"/>
  </ActuatesContinuousControlSignal>
  <ActuatesContinuousControlSignal>
    <DataSender KeyRef="BC2612"/>
    <DataReceiver KeyRef="B9455"/>
  </ActuatesContinuousControlSignal>
  <ActuatesContinuousControlSignal>
    <DataSender KeyRef="BC9916"/>
    <DataReceiver KeyRef="B3215"/>
  </ActuatesContinuousControlSignal>
  <ActuatesContinuousControlSignal>
    <DataSender KeyRef="BC3057"/>
    <DataReceiver KeyRef="B5958"/>
  </ActuatesContinuousControlSignal>
  <ActuatesContinuousControlSignal>
    <DataSender KeyRef="BC9497"/>
    <DataReceiver KeyRef="B4388"/>
  </ActuatesContinuousControlSignal>
  <ActuatesContinuousControlSignal>
    <DataSender KeyRef="BC1415"/>
    <DataReceiver KeyRef="B2526"/>
  </ActuatesContinuousControlSignal>
  <ActuatesContinuousControlSignal>
    <DataSender KeyRef="BC3625"/>
    <DataReceiver KeyRef="B4936"/>
  </ActuatesContinuousControlSignal>
  <ActuatesContinuousControlSignal>
```

```
          <DataSender KeyRef="BC3141"/>
          <DataReceiver KeyRef="B7632"/>
        </ActuatesContinuousControlSignal>
        <ActuatesContinuousControlSignal>
          <DataSender KeyRef="BC7273"/>
          <DataReceiver KeyRef="B1994"/>
        </ActuatesContinuousControlSignal>
      </ActuatesControl>
      <OptimalDesign/>
    </LogicalLayerRelationship>
  </LogicalLayer>
</UPSchema>
```

D.3 UP Instance of Setting 3

The below code shows the generated UP instance for the design and control parameters in setting 3.

```
<?xml version='1.0' encoding='UTF-8'?>
<UPSchema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <PhysicalLayer>
    <Consumer>
      <Building Key="B4978">
        <Name>Caddell</Name>
      </Building>
      <Building Key="B6513">
        <Name>Mason</Name>
      </Building>
      <Building Key="B1897">
        <Name>AllenSustain</Name>
      </Building>
      <Building Key="B1575">
        <Name>Chapin</Name>
      </Building>
      <Building Key="B5539">
        <Name>Crecine</Name>
      </Building>
      <Building Key="B4775">
        <Name>Ford</Name>
      </Building>
      <Building Key="B6940">
        <Name>MSE</Name>
      </Building>
      <Building Key="B2137">
        <Name>Klaus</Name>
      </Building>
      <Building Key="B9909">
```

```
      <Name>Savant</Name>
    </Building>
    <Building Key="B7613">
      <Name>Smithgall</Name>
    </Building>
    <ElectricVehicleChargingStation Key="EV2766">
      <Name>Area4</Name>
    </ElectricVehicleChargingStation>
    <ElectricVehicleChargingStation Key="EV8225">
      <Name>W23Deck</Name>
    </ElectricVehicleChargingStation>
  </Consumer>
  <Producer>
    <PV Key="P7442">
      <Name>CommunityPV</Name>
    </PV>
  </Producer>
  <DistrictSystem>
    <DistrictHeating Key="DH7878">
      <Name>GT_DH</Name>
    </DistrictHeating>
    <DistrictCooling Key="DC7512">
      <Name>GT_DC</Name>
    </DistrictCooling>
  </DistrictSystem>
  <Storage>
    <ElectricityBattery Key="EB1359">
      <Name>CommunityBattery</Name>
    </ElectricityBattery>
  </Storage>
  <Source>
    <ElectricityGrid Key="SE1000"/>
    <NaturalGas Key="SN1000"/>
  </Source>
  <PhysicalLayerRelationship>
    <SupplyPower>
      <EnergySender KeyRef="SE1000"/>
      <EnergyReceiver KeyRef="B4978"/>
    </SupplyPower>
    <SupplyPower>
      <EnergySender KeyRef="SE1000"/>
      <EnergyReceiver KeyRef="B6513"/>
    </SupplyPower>
    <SupplyPower>
      <EnergySender KeyRef="SE1000"/>
      <EnergyReceiver KeyRef="B1897"/>
    </SupplyPower>
    <SupplyPower>
      <EnergySender KeyRef="SE1000"/>
      <EnergyReceiver KeyRef="B1575"/>
    </SupplyPower>
```

```
  <SupplyPower>
   <EnergySender KeyRef="SE1000"/>
   <EnergyReceiver KeyRef="B5539"/>
  </SupplyPower>
  <SupplyPower>
   <EnergySender KeyRef="SE1000"/>
   <EnergyReceiver KeyRef="B4775"/>
  </SupplyPower>
  <SupplyPower>
   <EnergySender KeyRef="SE1000"/>
   <EnergyReceiver KeyRef="B6940"/>
  </SupplyPower>
  <SupplyPower>
   <EnergySender KeyRef="SE1000"/>
   <EnergyReceiver KeyRef="B2137"/>
  </SupplyPower>
  <SupplyPower>
   <EnergySender KeyRef="SE1000"/>
   <EnergyReceiver KeyRef="B9909"/>
  </SupplyPower>
  <SupplyPower>
   <EnergySender KeyRef="SE1000"/>
   <EnergyReceiver KeyRef="B7613"/>
  </SupplyPower>
  <SupplyPower>
   <EnergySender KeyRef="SE1000"/>
   <EnergyReceiver KeyRef="EV2766"/>
  </SupplyPower>
  <SupplyPower>
   <EnergySender KeyRef="SE1000"/>
   <EnergyReceiver KeyRef="EV8225"/>
  </SupplyPower>
  <SupplyPower>
   <EnergySender KeyRef="SE1000"/>
   <EnergyReceiver KeyRef="DC7512"/>
  </SupplyPower>
  <SupplyPower>
   <EnergySender KeyRef="P7442"/>
   <EnergyReceiver KeyRef="B4978"/>
  </SupplyPower>
  <SupplyPower>
   <EnergySender KeyRef="P7442"/>
   <EnergyReceiver KeyRef="B6513"/>
  </SupplyPower>
  <SupplyPower>
   <EnergySender KeyRef="P7442"/>
   <EnergyReceiver KeyRef="B1897"/>
  </SupplyPower>
  <SupplyPower>
   <EnergySender KeyRef="P7442"/>
   <EnergyReceiver KeyRef="B1575"/>
```

```xml
    </SupplyPower>
    <SupplyPower>
      <EnergySender KeyRef="P7442"/>
      <EnergyReceiver KeyRef="B5539"/>
    </SupplyPower>
    <SupplyPower>
      <EnergySender KeyRef="P7442"/>
      <EnergyReceiver KeyRef="B4775"/>
    </SupplyPower>
    <SupplyPower>
      <EnergySender KeyRef="P7442"/>
      <EnergyReceiver KeyRef="B6940"/>
    </SupplyPower>
    <SupplyPower>
      <EnergySender KeyRef="P7442"/>
      <EnergyReceiver KeyRef="B2137"/>
    </SupplyPower>
    <SupplyPower>
      <EnergySender KeyRef="P7442"/>
      <EnergyReceiver KeyRef="B9909"/>
    </SupplyPower>
    <SupplyPower>
      <EnergySender KeyRef="P7442"/>
      <EnergyReceiver KeyRef="B7613"/>
    </SupplyPower>
    <SupplyPower>
      <EnergySender KeyRef="P7442"/>
      <EnergyReceiver KeyRef="EV2766"/>
    </SupplyPower>
    <SupplyPower>
      <EnergySender KeyRef="P7442"/>
      <EnergyReceiver KeyRef="EV8225"/>
    </SupplyPower>
    <SupplyPower>
      <EnergySender KeyRef="P7442"/>
      <EnergyReceiver KeyRef="EB1359"/>
    </SupplyPower>
    <SupplyPower>
      <EnergySender KeyRef="EB1359"/>
      <EnergyReceiver KeyRef="B4978"/>
    </SupplyPower>
    <SupplyPower>
      <EnergySender KeyRef="EB1359"/>
      <EnergyReceiver KeyRef="B6513"/>
    </SupplyPower>
    <SupplyPower>
      <EnergySender KeyRef="EB1359"/>
      <EnergyReceiver KeyRef="B1897"/>
    </SupplyPower>
    <SupplyPower>
      <EnergySender KeyRef="EB1359"/>
```

```
   <EnergyReceiver KeyRef="B1575"/>
  </SupplyPower>
  <SupplyPower>
   <EnergySender KeyRef="EB1359"/>
   <EnergyReceiver KeyRef="B5539"/>
  </SupplyPower>
  <SupplyPower>
   <EnergySender KeyRef="EB1359"/>
   <EnergyReceiver KeyRef="B4775"/>
  </SupplyPower>
  <SupplyPower>
   <EnergySender KeyRef="EB1359"/>
   <EnergyReceiver KeyRef="B6940"/>
  </SupplyPower>
  <SupplyPower>
   <EnergySender KeyRef="EB1359"/>
   <EnergyReceiver KeyRef="B2137"/>
  </SupplyPower>
  <SupplyPower>
   <EnergySender KeyRef="EB1359"/>
   <EnergyReceiver KeyRef="B9909"/>
  </SupplyPower>
  <SupplyPower>
   <EnergySender KeyRef="EB1359"/>
   <EnergyReceiver KeyRef="B7613"/>
  </SupplyPower>
  <SupplyPower>
   <EnergySender KeyRef="EB1359"/>
   <EnergyReceiver KeyRef="EV2766"/>
  </SupplyPower>
  <SupplyPower>
   <EnergySender KeyRef="EB1359"/>
   <EnergyReceiver KeyRef="EV8225"/>
  </SupplyPower>
  <SupplyChilledWater>
   <EnergySender KeyRef="DC7512"/>
   <EnergyReceiver KeyRef="B6513"/>
  </SupplyChilledWater>
  <SupplyChilledWater>
   <EnergySender KeyRef="DC7512"/>
   <EnergyReceiver KeyRef="B1897"/>
  </SupplyChilledWater>
  <SupplyChilledWater>
   <EnergySender KeyRef="DC7512"/>
   <EnergyReceiver KeyRef="B1575"/>
  </SupplyChilledWater>
  <SupplyChilledWater>
   <EnergySender KeyRef="DC7512"/>
   <EnergyReceiver KeyRef="B5539"/>
  </SupplyChilledWater>
  <SupplyChilledWater>
```

```xml
  <EnergySender KeyRef="DC7512"/>
  <EnergyReceiver KeyRef="B4775"/>
</SupplyChilledWater>
<SupplyChilledWater>
  <EnergySender KeyRef="DC7512"/>
  <EnergyReceiver KeyRef="B6940"/>
</SupplyChilledWater>
<SupplyChilledWater>
  <EnergySender KeyRef="DC7512"/>
  <EnergyReceiver KeyRef="B2137"/>
</SupplyChilledWater>
<SupplyChilledWater>
  <EnergySender KeyRef="DC7512"/>
  <EnergyReceiver KeyRef="B9909"/>
</SupplyChilledWater>
<SupplyChilledWater>
  <EnergySender KeyRef="DC7512"/>
  <EnergyReceiver KeyRef="B7613"/>
</SupplyChilledWater>
<SupplySteam>
  <EnergySender KeyRef="DH7878"/>
  <EnergyReceiver KeyRef="B6513"/>
</SupplySteam>
<SupplySteam>
  <EnergySender KeyRef="DH7878"/>
  <EnergyReceiver KeyRef="B1897"/>
</SupplySteam>
<SupplySteam>
  <EnergySender KeyRef="DH7878"/>
  <EnergyReceiver KeyRef="B1575"/>
</SupplySteam>
<SupplySteam>
  <EnergySender KeyRef="DH7878"/>
  <EnergyReceiver KeyRef="B5539"/>
</SupplySteam>
<SupplySteam>
  <EnergySender KeyRef="DH7878"/>
  <EnergyReceiver KeyRef="B4775"/>
</SupplySteam>
<SupplySteam>
  <EnergySender KeyRef="DH7878"/>
  <EnergyReceiver KeyRef="B6940"/>
</SupplySteam>
<SupplySteam>
  <EnergySender KeyRef="DH7878"/>
  <EnergyReceiver KeyRef="B2137"/>
</SupplySteam>
<SupplySteam>
  <EnergySender KeyRef="DH7878"/>
  <EnergyReceiver KeyRef="B9909"/>
</SupplySteam>
```

```xml
      <SupplySteam>
        <EnergySender KeyRef="DH7878"/>
        <EnergyReceiver KeyRef="B7613"/>
      </SupplySteam>
      <SupplyNaturalGas>
        <EnergySender KeyRef="SN1000"/>
        <EnergyReceiver KeyRef="DH7878"/>
      </SupplyNaturalGas>
      <SupplyNaturalGas>
        <EnergySender KeyRef="SN1000"/>
        <EnergyReceiver KeyRef="B6513"/>
      </SupplyNaturalGas>
      <SupplyNaturalGas>
        <EnergySender KeyRef="SN1000"/>
        <EnergyReceiver KeyRef="B1897"/>
      </SupplyNaturalGas>
      <SupplyNaturalGas>
        <EnergySender KeyRef="SN1000"/>
        <EnergyReceiver KeyRef="B6940"/>
      </SupplyNaturalGas>
      <SupplyNaturalGas>
        <EnergySender KeyRef="SN1000"/>
        <EnergyReceiver KeyRef="B1575"/>
      </SupplyNaturalGas>
    </PhysicalLayerRelationship>
  </PhysicalLayer>
  <LogicalLayer>
    <Sensor ID="L1000">
      <Time/>
      <PriceSignal/>
      <Weather/>
      <Consumer/>
      <Producer/>
      <DistrictSystem/>
      <Storage/>
    </Sensor>
    <Controller ID="L2000"/>
    <Actuator ID="L3000">
      <DiscontinuousControl>
        <Consumer/>
        <Producer/>
      </DiscontinuousControl>
      <ContinuousControl>
        <Consumer/>
      </ContinuousControl>
    </Actuator>
    <OptimalControl ID="L4000">
      <Consumer/>
      <Producer/>
    </OptimalControl>
    <DesignOptimization ID="L5000">
```

```
<Consumer>
 <Building>
  <RoofEmissivity Key="BEEMI1010">
   <DataType>ContinuousReal</DataType>
   <TechnologyLevel1>
    <Value>0.4</Value>
    <Cost>5664</Cost>
   </TechnologyLevel1>
   <TechnologyLevel2>
    <Value>0.9</Value>
    <Cost>0</Cost>
   </TechnologyLevel2>
  </RoofEmissivity>
 </Building>
 <Building>
  <NaturalVentilation Key="BEEMI1002">
   <DataType>ContinuousInteger</DataType>
   <TechnologyLevel1>
    <Value>0</Value>
    <Cost>0</Cost>
   </TechnologyLevel1>
   <TechnologyLevel2>
    <Value>5</Value>
    <Cost>750</Cost>
   </TechnologyLevel2>
  </NaturalVentilation>
 </Building>
 <Building>
  <NaturalVentilation Key="BEEMI1001">
   <DataType>ContinuousInteger</DataType>
   <TechnologyLevel1>
    <Value>0</Value>
    <Cost>0</Cost>
   </TechnologyLevel1>
   <TechnologyLevel2>
    <Value>5</Value>
    <Cost>750</Cost>
   </TechnologyLevel2>
  </NaturalVentilation>
 </Building>
 <Building>
  <RoofEmissivity Key="BEEMI1009">
   <DataType>ContinuousReal</DataType>
   <TechnologyLevel1>
    <Value>0.4</Value>
    <Cost>67120</Cost>
   </TechnologyLevel1>
   <TechnologyLevel2>
    <Value>0.9</Value>
    <Cost>0</Cost>
   </TechnologyLevel2>
```

```
    </RoofEmissivity>
    <NaturalVentilation Key="BEEMI1007">
      <DataType>ContinuousInteger</DataType>
      <TechnologyLevel1>
        <Value>0</Value>
        <Cost>0</Cost>
      </TechnologyLevel1>
      <TechnologyLevel2>
        <Value>5</Value>
        <Cost>750</Cost>
      </TechnologyLevel2>
    </NaturalVentilation>
  </Building>
  <Building>
    <RoofEmissivity Key="BEEMI1003">
      <DataType>ContinuousReal</DataType>
      <TechnologyLevel1>
        <Value>0.4</Value>
        <Cost>11888</Cost>
      </TechnologyLevel1>
      <TechnologyLevel2>
        <Value>0.9</Value>
        <Cost>0</Cost>
      </TechnologyLevel2>
    </RoofEmissivity>
  </Building>
  <Building>
    <NaturalVentilation Key="BEEMI1011">
      <DataType>ContinuousInteger</DataType>
      <TechnologyLevel1>
        <Value>0</Value>
        <Cost>0</Cost>
      </TechnologyLevel1>
      <TechnologyLevel2>
        <Value>5</Value>
        <Cost>750</Cost>
      </TechnologyLevel2>
    </NaturalVentilation>
  </Building>
  <Building>
    <RoofEmissivity Key="BEEMI1005">
      <DataType>ContinuousReal</DataType>
      <TechnologyLevel1>
        <Value>0.4</Value>
        <Cost>16640</Cost>
      </TechnologyLevel1>
      <TechnologyLevel2>
        <Value>0.9</Value>
        <Cost>0</Cost>
      </TechnologyLevel2>
    </RoofEmissivity>
```

```
    </Building>
  </Consumer>
  <Producer>
    <PV>
      <ModuleArea Key="PEEMI1004">
        <DataType>ContinuousReal</DataType>
        <TechnologyLevel1>
          <Value>0</Value>
          <Cost>0</Cost>
        </TechnologyLevel1>
        <TechnologyLevel2>
          <Value>12000</Value>
          <Cost>4560000</Cost>
        </TechnologyLevel2>
      </ModuleArea>
    </PV>
  </Producer>
  <DistrictSystem/>
  <Storage>
    <ElectricityBattery>
      <Capacity Key="EEEMI1006">
        <DataType>ContinuousReal</DataType>
        <TechnologyLevel1>
          <Value>0</Value>
          <Cost>0</Cost>
        </TechnologyLevel1>
        <TechnologyLevel2>
          <Value>3000</Value>
          <Cost>1200000</Cost>
        </TechnologyLevel2>
      </Capacity>
    </ElectricityBattery>
  </Storage>
</DesignOptimization>
<LogicalLayerRelationship>
  <SendsSensorData/>
  <ProvidesSensorData>
    <DataSender IDRef="L1000"/>
    <DataReceiver IDRef="L2000"/>
  </ProvidesSensorData>
  <ProvidesActuatorData>
    <DataSender IDRef="L2000"/>
    <DataReceiver IDRef="L3000"/>
  </ProvidesActuatorData>
  <ProvidesVariableData>
    <PhysicalLayerToDesignOptimization>
      <DataSender KeyRef="B1575"/>
      <DataReceiver KeyRef="BEEMI1010"/>
    </PhysicalLayerToDesignOptimization>
    <PhysicalLayerToDesignOptimization>
      <DataSender KeyRef="B4978"/>
```

```
      <DataReceiver KeyRef="BEEMI1002"/>
    </PhysicalLayerToDesignOptimization>
    <PhysicalLayerToDesignOptimization>
      <DataSender KeyRef="B6940"/>
      <DataReceiver KeyRef="BEEMI1001"/>
    </PhysicalLayerToDesignOptimization>
    <PhysicalLayerToDesignOptimization>
      <DataSender KeyRef="EB1359"/>
      <DataReceiver KeyRef="EEEMI1006"/>
    </PhysicalLayerToDesignOptimization>
    <PhysicalLayerToDesignOptimization>
      <DataSender KeyRef="P7442"/>
      <DataReceiver KeyRef="PEEMI1004"/>
    </PhysicalLayerToDesignOptimization>
    <PhysicalLayerToDesignOptimization>
      <DataSender KeyRef="B4775"/>
      <DataReceiver KeyRef="BEEMI1009"/>
    </PhysicalLayerToDesignOptimization>
    <PhysicalLayerToDesignOptimization>
      <DataSender KeyRef="B4775"/>
      <DataReceiver KeyRef="BEEMI1007"/>
    </PhysicalLayerToDesignOptimization>
    <PhysicalLayerToDesignOptimization>
      <DataSender KeyRef="B9909"/>
      <DataReceiver KeyRef="BEEMI1003"/>
    </PhysicalLayerToDesignOptimization>
    <PhysicalLayerToDesignOptimization>
      <DataSender KeyRef="B2137"/>
      <DataReceiver KeyRef="BEEMI1011"/>
    </PhysicalLayerToDesignOptimization>
    <PhysicalLayerToDesignOptimization>
      <DataSender KeyRef="B1897"/>
      <DataReceiver KeyRef="BEEMI1005"/>
    </PhysicalLayerToDesignOptimization>
  </ProvidesVariableData>
  <ActuatesControl/>
  <OptimalDesign>
    <ConductsDesignOptimization>
      <DataSender KeyRef="BEEMI1010"/>
      <DataReceiver KeyRef="B1575"/>
    </ConductsDesignOptimization>
    <ConductsDesignOptimization>
      <DataSender KeyRef="BEEMI1002"/>
      <DataReceiver KeyRef="B4978"/>
    </ConductsDesignOptimization>
    <ConductsDesignOptimization>
      <DataSender KeyRef="BEEMI1001"/>
      <DataReceiver KeyRef="B6940"/>
    </ConductsDesignOptimization>
    <ConductsDesignOptimization>
      <DataSender KeyRef="EEEMI1006"/>
```

```xml
     <DataReceiver KeyRef="EB1359"/>
    </ConductsDesignOptimization>
    <ConductsDesignOptimization>
     <DataSender KeyRef="PEEMI1004"/>
     <DataReceiver KeyRef="P7442"/>
    </ConductsDesignOptimization>
    <ConductsDesignOptimization>
     <DataSender KeyRef="BEEMI1009"/>
     <DataReceiver KeyRef="B4775"/>
    </ConductsDesignOptimization>
    <ConductsDesignOptimization>
     <DataSender KeyRef="BEEMI1007"/>
     <DataReceiver KeyRef="B4775"/>
    </ConductsDesignOptimization>
    <ConductsDesignOptimization>
     <DataSender KeyRef="BEEMI1003"/>
     <DataReceiver KeyRef="B9909"/>
    </ConductsDesignOptimization>
    <ConductsDesignOptimization>
     <DataSender KeyRef="BEEMI1011"/>
     <DataReceiver KeyRef="B2137"/>
    </ConductsDesignOptimization>
    <ConductsDesignOptimization>
     <DataSender KeyRef="BEEMI1005"/>
     <DataReceiver KeyRef="B1897"/>
    </ConductsDesignOptimization>
   </OptimalDesign>
  </LogicalLayerRelationship>
 </LogicalLayer>
</UPSchema>
```

D.4 UP Instance of Setting 4

The below code shows the generated UP instance for the “without occupant’s lighting/appliance energy saving” in setting 4.

```xml
<?xml version='1.0' encoding='UTF-8'?>
<UPSchema xmlns:xs="http://www.w3.org/2001/XMLSchema">
 <PhysicalLayer>
  <Consumer>
   <Building Key="B3953">
    <Name>T03</Name>
   </Building>
   <Building Key="B8092">
    <Name>T04</Name>
   </Building>
   <Building Key="B7426">
```

```
      <Name>T05</Name>
    </Building>
  </Consumer>
  <Producer>
    <PV Key="P8889">
      <Name>CommunityPV</Name>
    </PV>
  </Producer>
  <DistrictSystem/>
  <Storage>
    <ElectricityBattery Key="EB6101">
      <Name>CommunityBattery</Name>
    </ElectricityBattery>
  </Storage>
  <Source>
    <ElectricityGrid Key="SE1000"/>
    <NaturalGas Key="SN1000"/>
  </Source>
  <PhysicalLayerRelationship>
    <SupplyPower>
      <EnergySender KeyRef="SE1000"/>
      <EnergyReceiver KeyRef="B3953"/>
    </SupplyPower>
    <SupplyPower>
      <EnergySender KeyRef="SE1000"/>
      <EnergyReceiver KeyRef="B8092"/>
    </SupplyPower>
    <SupplyPower>
      <EnergySender KeyRef="SE1000"/>
      <EnergyReceiver KeyRef="B7426"/>
    </SupplyPower>
    <SupplyPower>
      <EnergySender KeyRef="P8889"/>
      <EnergyReceiver KeyRef="B3953"/>
    </SupplyPower>
    <SupplyPower>
      <EnergySender KeyRef="P8889"/>
      <EnergyReceiver KeyRef="B8092"/>
    </SupplyPower>
    <SupplyPower>
      <EnergySender KeyRef="P8889"/>
      <EnergyReceiver KeyRef="B7426"/>
    </SupplyPower>
    <SupplyPower>
      <EnergySender KeyRef="P8889"/>
      <EnergyReceiver KeyRef="EB6101"/>
    </SupplyPower>
    <SupplyPower>
      <EnergySender KeyRef="EB6101"/>
      <EnergyReceiver KeyRef="B3953"/>
    </SupplyPower>
```

```
      <SupplyPower>
        <EnergySender KeyRef="EB6101"/>
        <EnergyReceiver KeyRef="B8092"/>
      </SupplyPower>
      <SupplyPower>
        <EnergySender KeyRef="EB6101"/>
        <EnergyReceiver KeyRef="B7426"/>
      </SupplyPower>
      <SupplyNaturalGas>
        <EnergySender KeyRef="SN1000"/>
        <EnergyReceiver KeyRef="B3953"/>
      </SupplyNaturalGas>
      <SupplyNaturalGas>
        <EnergySender KeyRef="SN1000"/>
        <EnergyReceiver KeyRef="B8092"/>
      </SupplyNaturalGas>
      <SupplyNaturalGas>
        <EnergySender KeyRef="SN1000"/>
        <EnergyReceiver KeyRef="B7426"/>
      </SupplyNaturalGas>
    </PhysicalLayerRelationship>
  </PhysicalLayer>
  <LogicalLayer>
    <Sensor ID="L1000">
      <Time/>
      <PriceSignal/>
      <Weather/>
      <Consumer/>
      <Producer/>
      <DistrictSystem/>
      <Storage/>
    </Sensor>
    <Controller ID="L2000"/>
    <Actuator ID="L3000">
      <DiscontinuousControl>
        <Consumer/>
        <Producer/>
      </DiscontinuousControl>
      <ContinuousControl>
        <Consumer/>
      </ContinuousControl>
    </Actuator>
    <OptimalControl ID="L4000">
      <Consumer>
        <Building>
          <TemperatureSetpoint Key="BEFMI1003">
            <Min>24</Min>
            <MinCost>0</MinCost>
            <Max>29</Max>
            <MaxCost>0</MaxCost>
            <Integer>No</Integer>
```

```
        </TemperatureSetpoint>
      </Building>
      <Building>
        <TemperatureSetpoint Key="BEFMI1004">
          <Min>24</Min>
          <MinCost>0</MinCost>
          <Max>29</Max>
          <MaxCost>0</MaxCost>
          <Integer>No</Integer>
        </TemperatureSetpoint>
      </Building>
      <Building>
        <TemperatureSetpoint Key="BEFMI1005">
          <Min>24</Min>
          <MinCost>0</MinCost>
          <Max>29</Max>
          <MaxCost>0</MaxCost>
          <Integer>No</Integer>
        </TemperatureSetpoint>
      </Building>
    </Consumer>
    <Producer/>
  </OptimalControl>
  <DesignOptimization ID="L5000">
    <Consumer/>
    <Producer>
      <PV>
        <ModuleArea Key="PEEMI1002">
          <DataType>ContinuousReal</DataType>
          <TechnologyLevel1>
            <Value>0</Value>
            <Cost>0</Cost>
          </TechnologyLevel1>
          <TechnologyLevel2>
            <Value>1000</Value>
            <Cost>537000</Cost>
          </TechnologyLevel2>
        </ModuleArea>
      </PV>
    </Producer>
    <DistrictSystem/>
    <Storage>
      <ElectricityBattery>
        <Capacity Key="EEEMI1003">
          <DataType>ContinuousReal</DataType>
          <TechnologyLevel1>
            <Value>0</Value>
            <Cost>0</Cost>
          </TechnologyLevel1>
          <TechnologyLevel2>
            <Value>150</Value>
```

```
     <Cost>6000</Cost>
    </TechnologyLevel2>
   </Capacity>
  </ElectricityBattery>
 </Storage>
</DesignOptimization>
<LogicalLayerRelationship>
 <SendsSensorData/>
 <ProvidesSensorData>
  <DataSender IDRef="L1000"/>
  <DataReceiver IDRef="L2000"/>
 </ProvidesSensorData>
 <ProvidesActuatorData>
  <DataSender IDRef="L2000"/>
  <DataReceiver IDRef="L3000"/>
 </ProvidesActuatorData>
 <ProvidesVariableData>
  <PhysicalLayerToOptimalControl>
   <DataSender KeyRef="B7426"/>
   <DataReceiver KeyRef="BEFMI1003"/>
  </PhysicalLayerToOptimalControl>
  <PhysicalLayerToOptimalControl>
   <DataSender KeyRef="B8092"/>
   <DataReceiver KeyRef="BEFMI1004"/>
  </PhysicalLayerToOptimalControl>
  <PhysicalLayerToOptimalControl>
   <DataSender KeyRef="B3953"/>
   <DataReceiver KeyRef="BEFMI1005"/>
  </PhysicalLayerToOptimalControl>
  <PhysicalLayerToDesignOptimization>
   <DataSender KeyRef="P8889"/>
   <DataReceiver KeyRef="PEEMI1002"/>
  </PhysicalLayerToDesignOptimization>
  <PhysicalLayerToDesignOptimization>
   <DataSender KeyRef="EB6101"/>
   <DataReceiver KeyRef="EEEMI1003"/>
  </PhysicalLayerToDesignOptimization>
 </ProvidesVariableData>
 <ActuatesControl>
  <ConductsOptimalControl>
   <DataSender KeyRef="BEFMI1003"/>
   <DataReceiver KeyRef="B7426"/>
  </ConductsOptimalControl>
  <ConductsOptimalControl>
   <DataSender KeyRef="BEFMI1004"/>
   <DataReceiver KeyRef="B8092"/>
  </ConductsOptimalControl>
  <ConductsOptimalControl>
   <DataSender KeyRef="BEFMI1005"/>
   <DataReceiver KeyRef="B3953"/>
  </ConductsOptimalControl>
```

```
      </ActuatesControl>
      <OptimalDesign>
        <ConductsDesignOptimization>
          <DataSender KeyRef="PEEMI1002"/>
          <DataReceiver KeyRef="P8889"/>
        </ConductsDesignOptimization>
        <ConductsDesignOptimization>
          <DataSender KeyRef="EEEMI1003"/>
          <DataReceiver KeyRef="EB6101"/>
        </ConductsDesignOptimization>
      </OptimalDesign>
    </LogicalLayerRelationship>
  </LogicalLayer>
</UPSchema>
```

REFERENCES

Alimohammadisagvand, Behrang, Juha Jokisalo, and Kai Sirén. 2018. “Comparison of Four Rule-Based Demand Response Control Algorithms in an Electrically and Heat Pump-Heated Residential Building.” *Applied Energy* 209(November 2017): 167–79. https://doi.org/10.1016/j.apenergy.2017.10.088.

ASHRAE. 2017. *ASHRAE Handbook of Fundamentals*.

Augenbroe, Godfried. 2019. “The Role of Simulation in Performance Based Building.” In *Building Performance Simulation for Design and Operation*,.

Ayodele, T. R., A. S.O. Ogunjuyigbe, K. O. Akpeji, and O. O. Akinola. 2017. “Prioritized Rule Based Load Management Technique for Residential Building Powered by PV/Battery System.” *Engineering Science and Technology, an International Journal* 20(3): 859–73. http://dx.doi.org/10.1016/j.jestch.2017.04.003.

Balaji, Bharathan et al. 2018. “Brick: Metadata Schema for Portable Smart Building Applications.” *Applied Energy* 226(February): 1273–92. http://dx.doi.org/10.1016/j.apenergy.2018.02.091.

Barbose, Galen, and Darghouth Naïm. 2019. *Primary Authors Tracking the Sun*.

Benndorf, G, N Réhault, M Clairembault, and T Rist. 2017. “Describing HVAC Controls in IFC-Method and Application.” In *Energy Procedia*, , 319–24. www.sciencedirect.comAvailableonlineatwww.sciencedirect.comwww.elsevier.com/locate/procedia1876-6102www.elsevier.com/locate/procedia (April 1, 2019).

Bhattacharya, Arka, Joern Ploennigs, and David Culler. 2015. “Short Paper: Analyzing Metadata Schemas for Buildings.” In , 33–34. http://dx.doi.org/10.1145/2821650.2821669. (September 14, 2018).

Bikakis, Nikos et al. 2013. “The XML and Semantic Web Worlds: Technologies, Interoperability and Integration: A Survey of the State of the Art.” *Studies in Computational Intelligence* 418(c): 319–60.

Booth, A T, R Choudhary, and D J Spiegelhalter. 2012. “Handling Uncertainty in Housing Stock Models.” *Building and Environment* 48(1): 35–47. http://www.statistics.gov.uk/cci/nugget.asp?id (December 6, 2018).

buildingSMART. “Industry Foundation Classes (IFC).” https://technical.buildingsmart.org/standards/ifc.

Carneiro, Gustavo Antonio. 2017. “Integrated Assessment of Buildings and Distributed Energy Resources (DER) at the Neighborhood Scale.” Georgia Institute of Technology. http://hdl.handle.net/1853/59272.

CEN. 2007a. “BS EN 15242:2007 Ventilation for Buildings - Calculation Methods for the Determinant of Air Flow Rates in Buildings Including Infiltration.”

———. 2007b. “EN 15316-4-5:2007 Heating Systems in Buildings — Method for Calculation of System Energy Requirements and System Efficiencies Part 4-5: Space Heating Generation Systems, the Performance and Quality of District Heating and Large Volume Systems.”

Charpenay, Victor, Sebastian Kabisch, Darko Anicic, and Harald Kosch. 2015. "An Ontology Design Pattern for IoT Device Tagging Systems." *Proceedings - 2015 5th International Conference on the Internet of Things, IoT 2015*: 138–45.

Chen, Yixing, Tianzhen Hong, Mary Ann Piette, and Lawrence Berkeley. 2017. "City-Scale Building Retrofit Analysis : A Case Study Using CityBES Abstract CityBES." *Proc. Building Simulation 2017* (August): 1084–91.

Cities, C40. 2018. "19 Global Cities Commit to Make New Buildings 'Net-Zero Carbon' by 2030." https://www.c40.org/press_releases/global-cities-commit-to-make-new-buildings-net-zero-carbon-by-2030?utm_content=76190917&utm_medium=social&utm_source=twitter (January 20, 2020).

CityGML. "CityGML." https://www.ogc.org/standards/citygml.

Clark, G, Allen, C. 1978. "The Estimation of Atmospheric Radiation for Clear and Cloudy Skies." In 2nd National Passive Solar Conference (AS/ISES), 675–78.

Davila, Carlos Cerezo, Christoph Reinhart, and Jamie Bemis. 2016. "Modeling Boston: A Workflow for the Generation of Complete Urban Building Energy Demand Models from Existing Urban Geospatial Datasets." *Energy* 117: 237–50. http://dx.doi.org/10.1016/j.energy.2016.10.057.

EPA-NR. 2007. "Functional Specification of the EPA-NR Software." (June).

Fathima, A. Hina, and K. Palanisamy. 2015. "Optimization in Microgrids with Hybrid

Energy Systems - A Review.” *Renewable and Sustainable Energy Reviews* 45: 431–46. https://www.sciencedirect.com/science/article/pii/S1364032115000696 (December 13, 2018).

Fonseca, J. N.B., and Marta J.N. Oliveira Panão. 2017. “Monte Carlo Housing Stock Model to Predict the Energy Performance Indicators.” *Energy and Buildings* 152: 503–15. https://linkinghub.elsevier.com/retrieve/pii/S0378778817308915 (December 10, 2018).

Fonseca, Jimeno A, Thuy An Nguyen, Arno Schlueter, and Francois Marechal. 2016. “City Energy Analyst (CEA): Integrated Framework for Analysis and Optimization of Building Energy Systems in Neighborhoods and City Districts.” *Energy and Buildings* 113: 202–26. http://dx.doi.org/10.1016/j.enbuild.2015.11.055 (November 30, 2018).

Fonseca, Jimeno A, and Arno Schlueter. 2015. “Integrated Model for Characterization of Spatiotemporal Building Energy Consumption Patterns in Neighborhoods and City Districts.” *Applied Energy* 142: 247–65. http://dx.doi.org/10.1016/j.apenergy.2014.12.068 (December 1, 2018).

Fontenot, Hannah, and Bing Dong. 2019. “Modeling and Control of Building-Integrated Microgrids for Optimal Energy Management – A Review.” *Applied Energy* 254(August): 113689. https://doi.org/10.1016/j.apenergy.2019.113689.

Fu, Ran et al. 2018. “2018 U . S . Utility-Scale Photovoltaics- Plus-Energy Storage System Costs Benchmark.” *National Renewable Energy Laboratory* (November):

32. https://www.nrel.gov/docs/fy19osti/71714.pdf.%0Ahttps://www.nrel.gov/docs/fy19osti/71714.pdf.

Gao, Hao, Christian Koch, and Yupeng Wu. 2019. “Building Information Modelling Based Building Energy Modelling: A Review.” *Applied Energy* 238(July 2017): 320–43. https://linkinghub.elsevier.com/retrieve/pii/S0306261919300327 (September 2, 2019).

Georgia Power. 2021. “Electric Service Tariff: Time of Use - Energy Only.” https://www.georgiapower.com/content/dam/georgia-power/pdfs/electric-service-tariff-pdfs/TOU-EO-11.pdf (June 6, 2021).

Grijalva, Santiago. 2014. “Decentralized Control : Benefits , Technologies and Challenges Smart Grid Functionality Restores the Balance.” : 1–12.

Gröger, Gerhard, and Lutz Plümer. 2012. “CityGML - Interoperable Semantic 3D City Models.” *ISPRS Journal of Photogrammetry and Remote Sensing* 71: 12–33.

Gupta, R. A., and Nand Kishor Gupta. 2015. “A Robust Optimization Based Approach for Microgrid Operation in Deregulated Environment.” *Energy Conversion and Management* 93: 121–31. http://dx.doi.org/10.1016/j.enconman.2015.01.008.

Heo, Yeonsook et al. 2015. “Scalable Methodology for Large Scale Building Energy Improvement: Relevance of Calibration in Model-Based Retrofit Analysis.” *Building and Environment* 87: 342–50. http://dx.doi.org/10.1016/j.buildenv.2014.12.016 (January 1, 2019).

Hong, Tianzhen et al. 2015. “An Ontology to Represent Energy-Related Occupant Behavior in Buildings. Part II: Implementation of the DNAS Framework Using an XML Schema.” *Building and Environment* 94(P1): 196–205.

Husein, Munir, and Il Yop Chung. 2018. “Optimal Design and Financial Feasibility of a University Campus Microgrid Considering Renewable Energy Incentives.” *Applied Energy* 225: 273–89. https://linkinghub.elsevier.com/retrieve/pii/S0306261918307293 (November 13, 2018).

Isikdag, Umit, and Sisi Zlatanova. 2009. “Towards Defining a Framework for Automatic Generation of Buildings in CityGML Using Building Information Models.” In *3D Geo-Information Sciences*, , 79–96. http://www.mendeley.com/research/lecture-notes-in-geoinformation-and-cartography-2/%5Cnhttp://books.google.com/books?hl=en&lr=&id=RaXsNdMtGLQC&oi=fnd&pg=PR5&dq=3D+Geo-Information+Sciences&ots=Pj-AEc7azB&sig=VdkihAftNNKxQLk_nulF.

ISO. 2007. “EN 15316-4-6 Heating Systems in Buildings — Method for Calculation of System Energy Requirements and System Efficiencies Part 4-6: Heat Generation Systems, Photovoltaic Systems.”

———. 2008. “ISO 13790:2008 Energy Performance of Buildings — Calculation of Energy Use for Space Heating and Cooling.”

Johari, F. et al. 2020. “Urban Building Energy Modeling: State of the Art and Future

Prospects." *Renewable and Sustainable Energy Reviews* 128(April).

Kavgic, M et al. 2010. "A Review of Bottom-up Building Stock Models for Energy Consumption in the Residential Sector." *Building and Environment* 45(7): 1683–97. https://ac.els-cdn.com/S0360132310000338/1-s2.0-S0360132310000338-main.pdf?_tid=42b92858-28a5-4f36-8f67-54ec874b8e5b&acdnat=1544471371_001dd049fc9139e3d591510c597d6a36 (December 10, 2018).

Lane, Catherine. 2020. "What Is a Solar Tracker and Is It Worth the Investment?" *SolarReviews*.

Lee, Sang Hoon. 2012. "Management of Building Energy Consumption and Energy Supply Network on Campus Scale." Georgia Institute of Technology.

Lee, Sang Hoon, Godfried Augenbroe, Jin Kook Lee, and Fei Zhao. 2013. "A Design Methodology for Energy Infrastructures at the Campus Scale." *Computer-Aided Civil and Infrastructure Engineering* 28(10): 753–68. https://onlinelibrary.wiley.com/doi/pdf/10.1111/mice.12050 (December 6, 2018).

Leland, Blank, and Anthony Tarquin. 2012. *Engineering Economy*. 7th ed. Mcgraw Hill.

Letellier-Duchesne, Samuel, Shreshth Nagpal, Michaël Kummert, and Christoph Reinhart. 2018. "Balancing Demand and Supply: Linking Neighborhood-Level Building Load Calculations with Detailed District Energy Network Analysis Models." *Energy* 150: 913–25. https://linkinghub.elsevier.com/retrieve/pii/S0360544218303724 (November 27, 2018).

Li, Bei, Robin Roche, and Abdellatif Miraoui. 2017. “Microgrid Sizing with Combined Evolutionary Algorithm and MILP Unit Commitment.” *Applied Energy* 188: 547–62. http://dx.doi.org/10.1016/j.apenergy.2016.12.038.

Lim, Hyunwoo, and Zhiqiang John Zhai. 2017. “Review on Stochastic Modeling Methods for Building Stock Energy Prediction.” *Building Simulation* 10(5): 607–24.

LLNL. 2019. “Energy Flow Charts.” https://flowcharts.llnl.gov/commodities/energy (January 20, 2020).

Mazzola, Simone et al. 2017. “Assessing the Value of Forecast-Based Dispatch in the Operation of off-Grid Rural Microgrids.” *Renewable Energy* 108: 116–25.

Moon, Hyeun Jun, Min Seok Choi, Sa Kyum Kim, and Seung Ho Ryu. 2011. “Case Studies for the Evaluation of Interoperability Between a BIM Based Architectural Model and Building Performance Analysis Programs.” *Building Simulation 2011: 12th Conference of International Building Performance Simulation Association*: 1521–26.

Nagpal, Shreshth, Caitlin Mueller, Arfa Aijazi, and Christoph F Reinhart. 2018. “A Methodology for Auto-Calibrating Urban Building Energy Models Using Surrogate Modeling Techniques.” *Journal of Building Performance Simulation*: 1–16. http://www.tandfonline.com/action/journalInformation?journalCode=tbps20 (November 13, 2018).

Nagpal, Shreshth, and Christoph F. Reinhart. 2018. “A Comparison of Two Modeling Approaches for Establishing and Implementing Energy Use Reduction Targets for a University Campus.” *Energy and Buildings* 173: 103–16. https://linkinghub.elsevier.com/retrieve/pii/S0378778817334497 (November 13, 2018).

NAVIGANT. 2018. “Energy Cloud 4.0.”

Nelson, James R, and Nathan G Johnson. 2020. “Model Predictive Control of Microgrids for Real-Time Ancillary Service Market Participation.” *Applied Energy* 269(December 2019): 114963. https://doi.org/10.1016/j.apenergy.2020.114963.

Netherlands Standardization Institute. 1999. “NEN 2916:1998 Energy Performance of Non-Residential Buildings.”

New, Joshua et al. 2018. “Automatic Building Energy Model Creation (AutoBEM) for Urban-Scale Energy Modeling and Assessment of Value Propositions for Electric Utilities.” In *Proceedings of the INternational Conference on Energy Engineering and Smart Grids (ESG)*,.

Northeast Energy Efficiency Partnerships (NEEP). 2020. “Grid-Interactive Efficient Buildings (GEBs) Tri-Region Status Report.” (January): 39.

Orehounig, Kristina, Ralph Evins, and Viktor Dorer. 2015. “Integration of Decentralized Energy Systems in Neighbourhoods Using the Energy Hub Approach.” *Applied Energy* 154: 277–89. http://dx.doi.org/10.1016/j.apenergy.2015.04.114 (December 31, 2018).

Pritoni, Marco et al. 2021. “Metadata Schemas and Ontologies for Building Energy Applications: A Critical Review and Use Case Analysis.” *Energies* 14(7): 2024.

“Project Haystack.” https://www.project-haystack.org/.

Quan, Steven Jige et al. 2015. “Urban Data and Building Energy Modeling: A GIS-Based Urban Building Energy Modeling System Using the Urban-EPC Engine.” In *Planning Support Systems and Smart Cities*, , 447–69. https://www.researchgate.net/publication/281803274 (November 27, 2018).

Rasmussen, Mads Holten, Maxime Lefrançois, Georg Ferdinand Schneider, and Pieter Pauwels. 2019. “BOT: The Building Topology Ontology of the W3C Linked Building Data Group.” *Semantic Web Journal* 0(0). http://www.semantic-web-journal.net/content/bot-building-topology-ontology-w3c-linked-building-data-group-0.

Reinhart, Christoph F., and Carlos Cerezo Davila. 2016. “Urban Building Energy Modeling - A Review of a Nascent Field.” *Building and Environment* 97: 196–202. http://dx.doi.org/10.1016/j.buildenv.2015.12.001.

Remmen, Peter et al. 2018. “TEASER: An Open Tool for Urban Energy Modelling of Building Stocks.” *Journal of Building Performance Simulation* 11(1): 84–98. https://doi.org/10.1080/19401493.2017.1283539.

Roth, Stephen. 2020. “GbXML.” https://www.gbxml.org/ (June 13, 2020).

RS Means. 2017. “Building Construction Cost Data.”

———. 2020. “Building Construction Cost Data.”

Schiefelbein, Jan et al. 2019. “Automated Urban Energy System Modeling and Thermal Building Simulation Based on OpenStreetMap Data Sets.” *Building and Environment* 149: 630–39.

Schneider, Georg Ferdinand, Pieter Pauwels, and Simone Steiger. 2017. “Ontology-Based Modeling of Control Logic in Building Automation Systems.” *IEEE Transactions on Industrial Informatics* 13(6): 3350–60.

Seppänen, O., W. J. Fisk, and D. Faulkner. 2003. “Cost Benefit Analysis of the Night-Time Ventilative Cooling in Office Building.” *Proceedings of the Healthy Buildings 2006*: 243–47.

SMUD. 2021. “Residential Time of Use Rates.” https://www.smud.org/en/Rate-Information/Residential-rates (June 6, 2021).

Soon Industrial Co. Ltd. 2020. “Automatic Outward Opening Window Openers.” https://www.soonkst.com/automatic-outward-opening-windows (September 15, 2020).

Swan, Lukas G, and V Ismet Ugursal. 2009. “Modeling of End-Use Energy Consumption in the Residential Sector: A Review of Modeling Techniques.” *Renewable and Sustainable Energy Reviews* 13(8): 1819–35. https://ac.els-cdn.com/S1364032108001949/1-s2.0-S1364032108001949-main.pdf?_tid=33ab31ff-ff9d-4291-9028-7070c47ebb11&acdnat=1544471529_ca57589f92e247ed65e0030f2719845e.

Thomas, Dimitrios, Olivier Deblecker, and Christos S. Ioakimidis. 2018. "Optimal Operation of an Energy Management System for a Grid-Connected Smart Building Considering Photovoltaics' Uncertainty and Stochastic Electric Vehicles' Driving Schedule." *Applied Energy* 210: 1188–1206. https://doi.org/10.1016/j.apenergy.2017.07.035.

Ton, Dan T., and Merrill A. Smith. 2012. "The U.S. Department of Energy's Microgrid Initiative." *Electricity Journal* 25(8): 84–94. http://dx.doi.org/10.1016/j.tej.2012.09.013 (January 20, 2020).

Tong, Wei. 2010. *Wind Power Generation and Wind Turbine Design*. WIT Press.

U.S. EIA. 2019. "How Much Energy Is Consumed in Residential and Commercial Buildings in the United States?" https://www.eia.gov/tools/faqs/faq.php?id=86&t=1 (July 1, 2020).

———. 2021a. "Natural Gas Prices." https://www.eia.gov/dnav/ng/ng_pri_sum_dcu_SGA_m.htm (June 6, 2021).

———. 2021b. "U.S. Price of Natural Gas Delivered to Residential Consumers." https://www.eia.gov/dnav/ng/hist/n3010us3m.htm (June 6, 2021).

Vejdan, Sadegh, and Santiago Grijalva. 2018. "Analysis of Multiple Revenue Streams for Privately-Owned Energy Storage Systems." In *2018 IEEE Power and Energy Conference at Illinois, PECI 2018*, IEEE, 1–5.

Zhang, Yuna. 2017. "OPTIMAL STRATEGIES FOR DEMAND CHARGE

REDUCTION BY COMMERCIAL BUILDING OWNERS." Georgia Institute of Technology.