Modelica Buildings library

Michael Wetter, Wangda Zuo[a], Thierry S. Nouidui, Xiufeng Pang
Lawrence Berkeley National Laboratory
Energy Technologies Area
Building Technology and Urban Systems Department
Simulation Research Group
Berkeley, CA 94720, USA

October 9, 2015

Abstract

This paper describes the *Buildings* library, a free open-source library that is implemented in Modelica, an equation-based object-oriented modeling language. The library supports rapid prototyping, as well as design and operation of building energy and control systems.

First, we describe the scope of the library, which covers HVAC systems, multi-zone heat transfer and multi-zone airflow and contaminant transport. Next, we describe differentiability requirements and address how we implemented them. We describe the class hierarchy that allows implementing component models by extending partial implementations of base models of heat and mass exchangers, and by instantiating basic models for conservation equations and flow resistances. We also describe associated tools for pre- and post-processing, regression tests, co-simulation and real-time data exchange with building automation systems.

The paper closes with an example of a chilled water plant, with and without water-side economizer, in which we analyzed the system-level efficiency for different control setpoints.

[a]Current address: College of Engineering, University of Miami,
Coral Gables, FL 33146, USA

Disclaimer

This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor the Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or the Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or the Regents of the University of California.

1   Introduction

This paper describes the free open-source Modelica `Buildings` library. Modelica is an equation-based, object-oriented language that has been developed to model complex engineered systems that are described by coupled systems of differential, algebraic and discrete equations.[1] The `Buildings` library is an open-source, free library with component and system models for building energy and control systems. The library is accompanied by Python modules that can be used to automate simulations and post-processing results, and by a user guide.

The library has been developed to support the following use cases in addition to conventional energy analysis:

Support for rapid prototyping: Users can rapidly add new component and system models by modifying existing models or extending basic models that implement, for example, steady-state and dynamic energy and mass balances of a heat and mass exchanger.[2]

Modeling of arbitrary HVAC system topologies: Users can model HVAC systems with non-conventional piping or ducting layout, and user-defined control algorithms. This allows for example the performance assessment of novel hydronic heating system where pumps located at each radiator replace the central pump and a hierarchical controller adjusts the furnace temperature based on the pump speed.[3]

Development, verification and hardware-in-the-loop testing of control algorithms: Users can use models from the library to support development of model-based control algorithms, and then verify the control sequences in a hardware-in-the-loop configuration. This process is described by Nouidui et al. (2012)[4], where a thermal model of a room has been linked to Radiance and to a test cell in order to optimize the control of the blinds. For an example, where a hierarchical supervisory and local loop control algorithm has been verified in simulation, see the implementation of the variable air volume flow control sequence from the ASHRAE Sequences of Operation for Common HVAC Systems.[5] In this example, the supervisory control algorithm is implemented using a state graph and the local loop control is implemented using block diagrams. Provan and Venturini[6] developed a stochastic predictive control algorithm for an underfloor heating system. The authors coupled discrete-time stochastic simulation with a Modelica model that computes room temperature and energy consumption.

Stabilization of feedback control: Users can start with a process model that may be non-linear and based on first principles. Next, they

[1] Mattsson, S. E. and H. Elmqvist (1997, April). Modelica – An international effort to design the next generation modeling language. In L. Boullart, M. Loccufier, and S. E. Mattsson (Eds.), *7th IFAC Symposium on Computer Aided Control Systems Design*, Gent, Belgium

[2] Wetter, M. (2009a). A modelica-based model library for building energy and control systems. In P. A. Strachan, N. J. Kelly, and M. Kummert (Eds.), *Proc. of the 11-th IBPSA Conference*, pp. 652–659. International Building Performance Simulation Association and University of Strathclyde

[3] Wetter, M. (2009a). A modelica-based model library for building energy and control systems. In P. A. Strachan, N. J. Kelly, and M. Kummert (Eds.), *Proc. of the 11-th IBPSA Conference*, pp. 652–659. International Building Performance Simulation Association and University of Strathclyde

[4] Nouidui, T. S., K. Phalak, W. Zuo, and M. Wetter (2012, September). Validation and application of the room model of the modelica Buildings library. In *Proc. of the 9-th International Modelica Conference*, Munich, Germany, pp. 727–736. Modelica Association

[5] ASHRAE (2006). *Sequences of Operation for Common HVAC Systems*. Atlanta, GA: ASHRAE; and Wetter, M. (2011b). Co-simulation of building energy and control systems with the building controls virtual test bed. *Journal of Building Performance Simulation 4*, 185–203

[6] Provan, G. and A. Venturini (2012, September). Stochastic simulation and interference using modelica. In *Proc. of the 9-th International Modelica Conference*, Munich, Germany. Modelica Association

can linearize the model around a set of steady-state operating points in order to analyze in the frequency domain the gain and phase margins.[7] Alternatively, the linearized models may be used as input to a model order reduction algorithm. Model order reduction algorithms compute a low order model that is suitable for gain scheduling in the frequency domain.[8]

Emulation of faults at the whole building-system level: For example, the LearnHVAC[9] and Learn High Performance Buildings[10] interactive eLearning software allows testing a variety of fault scenarios and examining the impact of faults on whole building performance. In these tools, models of faulty equipment or faulty controllers are part of a dynamic whole building model that includes HVAC, building envelope and detailed control algorithms whose feedback loops may compensate for some of the faults.

The above list represents applications in which the `Buildings` library has already been used. Work is currently in progress to use models from the `Buildings` library in fault detection and diagnostics algorithms, and to build models for model-based design of full-scale experiments at the Facility for Low Energy Experiments in Buildings[11] that is currently under construction at the Lawrence Berkeley National Laboratory (LBNL). In the future, we also envision the library to be increasingly used in conjunction with Building Information Models (BIM) and as the underlying technology for simulating HVAC and control systems in next-generation building simulation programs that can model arbitrary configurations of HVAC systems, including their continuous time, discrete time and event-driven control sequences.

If next-generation BIM also contain a modular description of the HVAC systems including their control logic, then advanced modeling and simulation programs will be needed that can solve the coupled systems of differential, algebraic and discrete equations that contain continuous time, discrete time and event-driven dynamics. If such next generation BIM standards contain specification of operational sequences, then more flexible simulation programs will be needed that allow the same semantics as actual control sequences. By same semantics, we mean that the control input will need to be a physical quantity that is measured by a sensor and not a "load" with units of power that is required to meet a set-point under the assumption of idealized control, as is customary in many building simulation programs. Furthermore, the output of the controller will need to be an actuator signal, such as a damper position, and not an air flow rate as is customary in conventional building simulation programs (such as EnergyPlus and DOE-2) that do not compute the pressure

[7] Eisenhower, B., K. Gasljevic, and I. Mezić (2012, August). Control-oriented modeling and calibration of building energy models using modelica. In *Proc. of SimBuild*, Madison, WI. IBPSA-USA

[8] Wetter, M. (2009b, June). Modelica-based modeling and simulation to support research and development in building energy and control systems. *Journal of Building Performance Simulation* 2(2), 143–161

[9] SuPerB (2012). Learnhvac educational software

[10] Deringer, J. J., J. E. Nahman, K. Heming, M. Wetter, X. Pang, and M. Konstantoglou (2012, August). LearnHPB and eLAD - two related online eLearning platforms for high performance buildings. Summer Study on Energy Efficiency in Buildings, Pacific Grove, CA. ACEEE

[11] See http://flexlab.lbl.gov/

distribution in duct networks. This is because the air flow rate is not only a function of the damper position, but also of the damper authority, fan speed, fan curve, duct distribution network and position of other dampers that are part of the same duct network. In addition, it should be possible to model HVAC systems with any topology of the duct and piping network, and not only systems that can be decomposed into primary loops and secondary loops as is assumed in other energy simulation programs.[12]

The use of acausal model formulations for building energy systems can be traced back to Silverman et al. (1981).[13] They describe a network formulation that uses a node and arc formulation, where nodes are components (equipment or zones) that do not have distinct inputs and outputs, and each arc carries fluid properties. The equations are then posed as an optimization problem where the independent variables are the variables of the arcs and the constraints are the equations of the nodes. This work led to the development of the PC-based energy simulation program ENET that was first presented by Low and Sowell in 1982[14] and then by Sowell et al. in 1984.[15] In these papers, the authors describe how ENET selects from a network definition and a set of equipment modules, a suitable set of equations using graph theory. These equations are then translated to high-level language instructions directly acceptable to a compiler. These developments provided the foundation of the SPANK program[16], which was later renamed to SPARK.[17]

[12] Wetter, M. (2011a). *Building Performance Simulation for Design and Automation*, Chapter The Future of Building System Modeling and Simulation, pp. 481–504. Oxon, OX: Taylor & Francis

[13] Silverman, G., S. Jurovics, D. Low, and E. Sowell (1981). Modeling and optimization of HVAC systems using network concepts. *ASHRAE Transactions 87*(2), 585–597

[14] Low, D. and E. Sowell (1982, September). ENET, a PC-based building energy simulation system. In *Energy Programs Conference*, Austin, Texas, pp. 2–7. IBM Real Estate and Construction Division

[15] Sowell, E., K. Taghavi, H. Levy, and D. Low (1984). Generation of building energy system models. *ASHRAE Transactions 90*(1B), 573–586

[16] Sowell, E. F., W. F. Buhl, A. E. Erdem, and F. C. Winkelmann (1986, September). A prototype object-based system for HVAC simulation. Technical Report LBL-22106, Lawrence Berkeley National Laboratory; and Sowell, E. F., W. F. Buhl, and J.-M. Nataf (1989, June). Object-oriented programming, equation-based submodels, and system reduction in SPANK. In *Proceedings of the Second International IBPSA Conference*, Vancouver, BC, Canada, pp. 141–146

[17] Buhl, W., A. Erdem, F. Winkelmann, and E. Sowell (1993, August). Recent improvements in SPARK: Strong component decomposition, multivalued objects,and graphical interface. In *Proc. of the 3-rd IBPSA Conference*, Adelaide, Australia, pp. 283–291

In 1989, Sahlin and Sowell propose a simulator-neutral and natural format for component model expression called Neutral Model Format (NMF), which can be converted to efficient numerical algorithmic form.[18] NMF is still in use in the commercial software IDA/ICE.[19] In 1991, Charlesworth et al. developed the Energy Kernel System, which appears to be the first attempt that adopted an object-oriented programming paradigm that uses both, object-instantiation and object-inheritance, for the creation of building energy performance models.[20]

Equation-based acausal modeling languages have been used earlier for applications other than building energy simulation: In 1978, Elmqvist[21] introduced in his dissertation the Dymola modeling language for large continuous systems in which ordinary differential equations and algebraic equations need not be converted to assignment statements, and which employs equation sorting and formula manipulation. In 1997, Mattsson and Elmqvist report on an international effort to design Modelica, a next generation modeling language, which incorporates many ideas of the algorithms presented in the dissertation of Elmqvist.[22]

Modelica has been applied earlier for building energy modeling applications but an extensive free open-source library with both steady-state and dynamic component models for building energy systems has not yet been available. Other open-source developments of building energy models in Modelica that are freely available have been reported by Merz (2002)[23] and by Felgner et al. (2002)[24] who describe the library ATPlus for thermal building simulation. In 2005, Hoh et al. expanded the components of the ATPlus library to include a room model with heat exchangers that are embedded in wall constructions.[25]

[18] Sahlin, P. and E. F. Sowell (1989, June). A neutral format for building simulation models. In *Proceedings of the Second International IBPSA Conference*, Vancouver, BC, Canada, pp. 147–154

[19] Björsell, N., A. Bring, L. Eriksson, P. Grozman, M. Lindgren, P. Sahlin, A. Shapovalov, and M. Vuolle (1999, September). IDA indoor climate and energy. In N. Nakahara, H. Yoshida, M. Udagawa, and J. Hensen (Eds.), *Proc. of the 6-th IBPSA Conference*, Kyoto, Japan, pp. 1035–1042

[20] Charlesworth, P., J. A. Clarke, G. Hammond, A. Irving, K. James, B. Lee, S. Lockley, R. D. Mac, D. Tang, T. J. Wiltshire, and A. J. Wright (1991, August). The energy kernel system. In J. A. Clarke, J. W. Mitchell, and R. C. V. de Perre (Eds.), *Proc. of the IBPSA Conference*, Nice, France

[21] Elmqvist, H. (1978). *A structured model language for large continuous systems*. Ph. D. thesis, Lund Institute of Technology, Lund, Sweden

[22] Mattsson, S. E. and H. Elmqvist (1997, April). Modelica – An international effort to design the next generation modeling language. In L. Boullart, M. Loccufier, and S. E. Mattsson (Eds.), *7th IFAC Symposium on Computer Aided Control Systems Design*, Gent, Belgium

[23] Merz, R. M. (2002, September). *Objektorientierte Modellierung thermischen Gebäudeverhaltens*. Ph. D. thesis, Universität Kaiserslautern

[24] Felgner, F., S. Agustina, R. C. Bohigas, R. Merz, and L. Litz (2002, March). Simulation of thermal building behaviour in modelica. In M. Otter (Ed.), *Proceedings of the 2nd Modelica conference*, Oberpfaffenhofen, Germany, pp. 147–154. Modelica Association and Deutsches Zentrum fur Luft- und Raumfahrt

[25] Hoh, A., T. Haase, T. Tschirner, and D. Müller (2005, March). A combined thermo-hydraulic approach to simulation of active building components applying modelica. In G. Schmitz (Ed.), *Proceedings of the 4th Modelica conference*, Hamburg, Germany. Modelica Association and Hamburg University of Technology

A hygrothermal building model has been developed for the Modelica `BuildingPhysicsLibrary` library which allows multizone building heat and mass transfer analysis.[26] A prototype CFD package of Modelica models has been coupled to models of conductive heat transfer and controls.[27] Badakhshani and Mueller (2010) developed a Modelica library to conduct exergy analysis of space heating systems.[28] Bonvini and Leva (2011) implemented, in Modelica, object-oriented sub-zonal models for two-dimensional room air flow.[29]

Commercial libraries for steady-state and transient simulation of thermodynamic systems include the `AirConditioning` library and the `TIL` library.[30] Both libraries model thermodynamic machines, including the dynamics of vapor compression cycles, based on the geometry of the equipment. They have been primarily used for product development of thermodynamic machines as opposed to whole building simulation.

2   Nomenclature

1. We denote by $\mathbb{R}$ the set of real numbers.

2. We use $\triangleq$ to denote equal by definition.

3. $f(\cdot)$ denotes a function where $(\cdot)$ stands for the undesignated variables. $f(x)$ denotes the value of $f(\cdot)$ for the argument x. $f\colon A \to B$ indicates that the domain of $f(\cdot)$ is in the space A, and that the image of $f(\cdot)$ is in the space B.

4. We say that a function $f\colon \mathbb{R}^n \to \mathbb{R}$ is once continuously differentiable if $f(\cdot)$ is defined on $\mathbb{R}^n$, and if $f(\cdot)$ has a continuous derivative on $\mathbb{R}^n$.

5. For a function $f\colon \mathbb{R} \to \mathbb{R}$, we say that $f(\cdot)$ is globally bounded if there exists a $\beta > 0$ such that $|f(x)| < \beta$ for all $x \in \mathbb{R}$.

6. We use typewriter font for the names of Modelica classes (packages, models etc.).

3   Scope

The `Buildings` library version 1.2 build 1 contains dynamic and steady-state models for building energy and control systems. The library contains models for air-based HVAC systems, water-based heating systems, controls, heat transfer among rooms and the outside, and multizone airflow, including natural ventilation and contaminant transport. A detailed description of all models can be found online at `http://simulationresearch.lbl.gov/modelica`. The models are organized in the following main packages:

[26] Nytsch-Geusen, C., T. Nouidui, A. Holm, , and W. Haupt (2005, August). A hygrothermal building model based on the object-oriented modeling language modelica. In I. Beausoleil-Morrison and M. Bernier (Eds.), *Proceedings of the Ninth International IBPSA Conference*, Volume 1, Montreal, Canada, pp. 867–876. International Building Performance Simulation Association and Ecole Polytechnique de Montreal; and Nouidui, T., K. Sedlbauer, C. Nytsch-Geusen, and K. Kießl (2009, October). Neue objektorientierte hygrothermische modellbibliothek zur ermittlung des hygrothermischen und hygienischen komforts in räumen. *Bauphysik 31*(5), 271–278

[27] Brown, J. (2010, November). *Computational fluid dynamics in an equation-based, acausal modeling environment*. Ph. D. thesis, Georgia Institute of Technology, Atlanta, GA

[28] Badakhshani, A. and D. Müller (2010, September). Dynamische exergetische Analyse von Gebäuden anhand der Gesamtsystemsimulation in der objekt-orientierten programmiersprache Modelica. In *BauSim*, Vienna, Austria. IBPSA Austria-Germany

[29] Bonvini, M. and A. Leva (2011). Object-oriented sub-zonal modelling for efficient energy-related building simulation. *Mathematical and Computer Modelling of Dynamical Systems 17*(6), 543–559

[30] Tummescheit, H., J. Eborn, and K. Prölss (2005, March). AirConditioning - a Modelica library for dynamic simulation of AC systems. In G. Schmitz (Ed.), *Proceedings of the 4th Modelica conference*, Hamburg, Germany, pp. 185–192. Modelica Association and Hamburg University of Technology; and Schulze, C., M. Gräber, M. Huhn, and U. Grätz (2011, March). Real-time simulation of vapour compression cycles. In *Proc. of the 8-th International Modelica Conference*, Dresden, Germany, pp. 48–55. Modelica Association

User's Guide: This package contains a user's guide for the library. In addition, some individual packages contain additional user's guides that are applicable to components of the respective packages.

Airflow: This package contains models for multizone airflow and contaminant transport that are similar to models implemented in CONTAM.[31] The models compute air flow between adjacent rooms and between rooms and the outside. Each room volume is assumed to be completely mixed. The driving force for the air flow are pressure differences that can be induced, for example, by a flow imbalance of the HVAC system, density differences across large openings such as doors or open windows, stack effects in high rise buildings, and wind pressure on the building facade. The physics of the models in this package is described in Wetter (2006).[32] However, this package is a new implementation that led to simpler models and more robust simulations due to the use of the Modelica stream connectors[33] that have been introduced in the `Modelica.Fluid` package[34] of the Modelica Standard Library.

BoundaryConditions: This package contains models to read TMY3 weather data and to compute boundary conditions, such as solar irradiations and sky temperatures.

Controls: This package contains blocks that model continuous time and discrete time controllers. There are also blocks that can be used to schedule setpoints.

Fluid: This is the largest package of the library. It contains component models for air-based and water-based HVAC systems. The package contains models such as chillers, cooling towers, heat exchangers, flow resistances, valves and air dampers, mass flow boundary conditions, pressure boundary conditions, pumps and fans, sensors and energy storage. Simple component models are typically based on first principles, whereas models of more complex equipment typically use steady-state performance curves such as the ones used in EnergyPlus. In addition, differential equations are added to deduce from the steady-state performance the transient behavior, such as due to evaporation of water on the coil surface of a direct evaporative cooling coil once the compressor is switched off. A recent description of these models can be found in Wetter et al. (2011).[35]

[31] Dols, W. S. and G. N. Walton (2002, November). CONTAM 2.0 user manual, multizone airflow and contaminant transport analysis software. Technical Report NISTIR 6921, National Institute of Standards and Technology

[32] Wetter, M. (2006, September). Multizone airflow model in Modelica. In C. Kral and A. Haumer (Eds.), *Proc. of the 5-th International Modelica Conference*, Volume 2, Vienna, Austria, pp. 431–440. Modelica Association and Arsenal Research

[33] Franke, R., F. Casella, M. Otter, M. Sielemann, H. Elmqvist, S. E. Mattsson, and H. Olsson (2009, September). Stream connectors – an extension of modelica for device-oriented modeling of convective transport phenomena. In F. Casella (Ed.), *Proc. of the 7-th International Modelica Conference*, Como, Italy. Modelica Association

[34] Franke, R., F. Casella, M. Otter, K. Proelss, M. Sielemann, and M. Wetter (2009, September). Standardization of thermo-fluid modeling in `Modelica.Fluid`. In F. Casella (Ed.), *Proc. of the 7-th International Modelica Conference*, Como, Italy. Modelica Association

[35] Wetter, M., W. Zuo, and T. S. Nouidui (2011b, March). Recent developments of the modelica buildings library for building heating, ventilation and air-conditioning systems. In *Proc. of the 8-th International Modelica Conference*, Dresden, Germany, pp. 266–275. Modelica Association

HeatTransfer: This package contains models for steady-state and dynamic heat transfer through opaque constructions such as multi-layered walls. It also contains models for layer-by-layer heat transfer through fenestration systems that are similar to the algorithms of the WINDOW 6 program.[36]

Media: This package contains different implementations for various media. The media models in this package are compatible with models of the `Modelica.Media` package, but the implementation is in general simpler. This often leads to fewer nonlinear equations and system models that are more robust. Due to the simplifications, the media models of this package are accurate for a smaller temperature range than the models in `Modelica.Media`, but the smaller temperature range generally suffices for building HVAC applications.

Rooms: This package contains models for heat transfer in rooms and through the building envelope. Multiple instances of these models can be connected to create a multi-zone building model. To compute the air exchange between rooms and between a room and the exterior, the room models can be connected to multi-zone air exchange models from the package `Buildings.Airflow`. The room models can also be linked to models of HVAC systems that are composed of the components in the package `Buildings.Fluid`. A detailed model description can be found in Wetter et al. (2011).[37]

Utilities: This package contains utility models, such as for computing thermal comfort, for run-time data exchange with the Building Controls Virtual Test Bed[38], or for computing psychometric properties. The package also contains various mathematical functions that are used throughout the library. A package that allows interpreting Python code during run-time from a Modelica model is currently in development.

The above packages also contain user's guides, tutorials and examples. The user's guides explain, for models of the respective package, what their underlying assumptions, limitations and proper use are. The tutorials provide step by step instructions for how to build large system models, the examples illustrate different usages of models. These examples range from small models where individual components are tested and users can quickly understand, to whole building simulations that contain thousands of component models.

There is also a separate user guide[39] that covers general information that is not specific to the use of individual sublibraries or models. Discussed topics include how to get started, best practices, how to post-process results using Python, work-arounds for problems and how to develop new component and system models.

---

[36] Mitchell, R., C. Kohler, L. Zhu, D. Arasteh, J. Carmody, C. Huizenga, and D. Curcija (2011, January). THERM 6.3 / WINDOW 6.3 NFRC simulation manual. Technical Report LBNL-48255, Lawrence Berkeley National Laboratory, Berkeley, CA; and Carli, Inc. (2006, October). *TARCOG: Mathematical Models for Calculation of Thermal Performance of Glazing Systems with or without Shading Devices*. Amherst, MA: Carli, Inc

[37] Wetter, M., W. Zuo, and T. S. Nouidui (2011a, November). Modeling of heat transfer in rooms in the Modelica "Buildings" library. In *Proc. of the 12-th IBPSA Conference*, pp. 1096–1103. International Building Performance Simulation Association

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

[39] LBNL (2012). User's guide of the modelica Buildings library

In total, the `Buildings` library contains 236 models and blocks, and 94 functions. The models, blocks and functions are used in 279 example models. The example models are included in the regression tests that are explained in Section 5.2.

4   Implementation

4.1   Differentiability

The models of the `Buildings` library define systems of coupled differential, algebraic and discrete equations. The differential algebraic systems of equations (DAE) that define the physics of the various components are formulated in such a way that the equations are once continuously differentiable with respect to all variables, and that the derivatives are bounded on compact sets. These properties are necessary and sufficient to establish existence and uniqueness of a solution to the differential equations defined by the model equations.[40] Differentiability is also needed because model equations are differentiated during translation to reduce the index of the DAE system if it is higher than one, and to compute the Jacobian matrix. In addition, boundness of the derivative is needed to avoid situations where the step size of Newton-based solvers for nonlinear equations shrinks to zero.

The implications of this differentiability requirement is that discontinuous functions, continuous but non-differentiable functions and functions with infinite derivatives are approximated by differentiable functions that have bounded derivatives on compact sets. Discontinuous functions arise, for example, by if-then-else equations that select empirical correlations for different ranges of model variables. Continuous but non-differentiable functions arise, for example, when computing the minimum capacity flow rate in an effectiveness-NTU heat exchanger model. Functions with infinite derivatives arise, for example, in free convection heat transfer models that compute $h(\Delta T) = k |\Delta T|^{1/3}$, where $h(\cdot)$ is the convection coefficient, $k > 0$ is a constant and ΔT is the temperature difference.

In a small neighborhood around zero mass flow rate, steady-state energy and mass balances are replaced by differentiable equations in order to avoid singularities that can cause numerical problems if the involved variables are the output of iterative solvers that compute numerical approximations to these variables. For example, consider a steady-state model of a heat exchanger that computes the enthalpy difference between outlet and inlet as $\Delta h = \dot{Q}/\dot{m}$, where $\dot{Q}$ is the heat added to the medium and $\dot{m}$ is the mass flow rate. Clearly, a necessary condition for Δh to be finite is $\dot{Q}/\dot{m} \to 0$ as $\dot{m} \to 0$. However,

[40] Coddington, E. A. and N. Levinson (1955). *Theory of ordinary differential equations*. McGraw-Hill Book Company, Inc., New York-Toronto-London

for some system models, computing $\dot{Q}$ requires an iterative solution. Thus, $\dot{Q}$ can only be approximated by some $\dot{Q}^*(\epsilon)$, that satisfies $|\dot{Q}(\epsilon) - \dot{Q}| \le K\epsilon$, where $K > 0$ is a constant and $\epsilon > 0$ is the solver tolerance. This considerably complicates the treatment of zero flow as the ratio needs to satisfy $\dot{Q}^*(\epsilon)/\dot{m} \to 0$ as $\dot{m} \to 0$. To guarantee this behavior, we provide a special implementation of $\Delta h = \dot{Q}/\dot{m}$ in a reusable base class.

Specifically, instead of implementing

$$\Delta h = \dot{Q}/\dot{m}, \tag{1}$$

we implemented

$$\Delta h \triangleq \dot{Q}\,\phi(\dot{m}_\delta, \dot{m}), \tag{2}$$

with

$$\phi(\delta, x) \triangleq \begin{cases} p(\delta, x), & \text{for } x \in [-\delta, \delta], \\ 1/x, & \text{otherwise}, \end{cases} \tag{3}$$

where $\phi \colon \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is a twice continuously differentiable function that is bounded on compact sets. We constructed the function $p \colon \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ in such a way that it satisfies

$$\begin{align} p(\delta, x) &= -p(\delta, -x), \tag{4a}\\ p(\delta, \delta) &= 1/\delta, \tag{4b}\\ p'(\delta, \delta) &= -1/\delta^2, \tag{4c}\\ p'(\delta, 0) &\neq 0, \tag{4d}\\ p''(\delta, \delta) &= -2/\delta^3, \tag{4e} \end{align}$$

where $p'(\cdot, \cdot)$ and $p''(\cdot, \cdot)$ denote the first and second order partial derivatives with respect to the second argument. The condition $p'(\delta, 0) \neq 0$ has been selected to avoid that the first derivative vanishes at the origin, because a Newton solver that solves $p(\delta, x) = 0$ for x may compute the sequence $x_{n+1} = x_n - p(\delta, x_n)/p'(\delta, x_n)$.

The function that we selected and that satisfies (4) is

$$p(\delta, x) \triangleq \frac{x}{\delta^2} + x\left| \frac{2}{\delta^3}x - \frac{3}{\delta^5}x^3 + \frac{1}{\delta^7}x^5 \right|. \tag{5}$$

Figure 1 shows the graph of this function. The term x/δ^2 is used to ensure that the first derivative is non-zero around a neighborhood of $x = 0$, and therefore by the Implicit Function Theorem, there exists in a neighborhood of the origin an inverse function $p^{-1}(\delta, \cdot)$ that is differentiable.[41]

Note that in (2), for $\dot{m}(t) \in (-\dot{m}_\delta, \dot{m}_\delta) \setminus \{0\}$, energy is not conserved. Therefore, $\dot{m}_\delta$ should be chosen small. However, $\dot{m}_\delta$ should
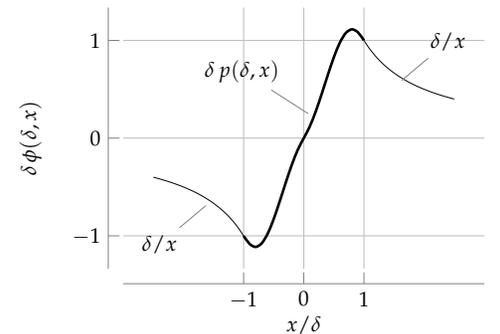

Figure 1: Plot of $\delta\,\phi(\delta, x)$.

[41] Polak, E. (1997). *Optimization, Algorithms and Consistent Approximations*, Volume 124 of *Applied Mathematical Sciences*. Springer Verlag

not be too small compared to the design mass flow rate, as this will cause $\phi(\dot{m}_\delta, \dot{m})$ to be large in magnitude near $\dot{m} \approx \pm\dot{m}_\delta$, which may lead to problems for numerical solvers. The default setting in the `Buildings` library is $\dot{m}_\delta = 10^{-7}\, \dot{m}_0$, where $\dot{m}_0$ is the design mass flow rate. At this low mass flow rate, the impact on the energy consumption is negligible.

To assist component model developers in the regularization of these functions, the package `Buildings.Utilities.Math` provides differentiable approximating functions for many formulations.[42]

4.2   Class Hierarchy

An individual implementation of this special treatment of conservation equations for each model would be cumbersome, error prone and expensive to maintain. We therefore implemented the `Buildings` library in such a way that generic, reusable base classes are provided that implement steady-state and dynamic conservation equations for heat and mass transfer, as well as equations for flow resistances. These base classes are compatible with, and in some cases based on, models from the `Modelica.Fluid` package[43] that are part of the Modelica Standard Library.

Such implementations of reusable base classes are provided for components with two and four fluid ports.[44] For components with two fluid ports, Figure 2 shows the class hierarchy that is used to implement these models. A similar class hierarchy is used for components with four fluid ports. In the figure, white components are base classes, while gray components are component models that can be instantiated by users to build system models. A solid line is used to indicate object-inheritance, while a dashed line is used to indicate object-instantiation. Due to space constraints, only a small part of the object tree is shown.

The three primitive building blocks are as follows:

ConservationEquation: This model provides conservation equations for energy, mass fractions and trace substances for components with any number of fluid ports. The conservation equations can be steady-state or dynamic. There is also a connector that allows adding heat or mass fractions into the components, which then participate in the conservation equation of the fluid. There is no pressure drop between the fluid ports.

StaticTwoPortConservationEquations: This class is similar to `ConservationEquation`. However, the component must have two fluid ports, and the conservation equations are steady-state. For small mass flow rates, the approximating function $p(\dot{m}_\delta, \cdot)$ explained above is used to avoid

[42] Note that Modelica simulation environments can in general properly handle non-differentiabilities. However, a numerically sound treatment requires an event iteration that is computationally costly. When formulating equations for physical phenomena, we therefore replaced non-differentiable equations with approximations that are continuously differentiable.

[43] Franke, R., F. Casella, M. Otter, K. Proelss, M. Sielemann, and M. Wetter (2009, September). Standardization of thermo-fluid modeling in `Modelica.Fluid`. In F. Casella (Ed.), *Proc. of the 7-th International Modelica Conference*, Como, Italy. Modelica Association

[44] In Modelica, a fluid port is an interface for quasi one-dimensional fluid flow. It is used to connect models that carry the same medium. The fluid port propagates pressure, mass flow rate, enthalpy, species concentration and trace substance concentrations among models.
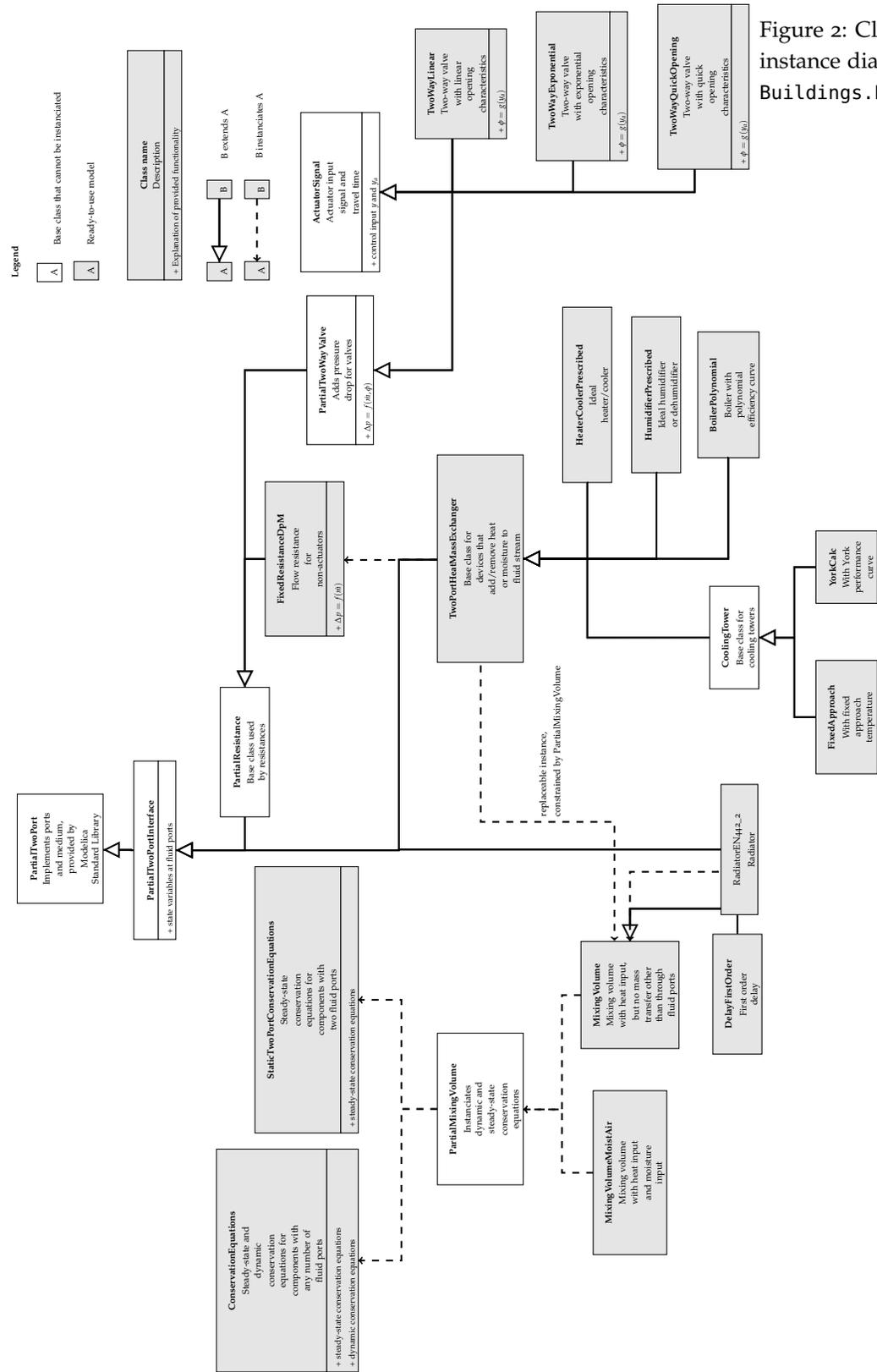
Figure 2: Class inheritance and instance diagram for part of the `Buildings.Fluid` package.

numerical problems near zero mass flow rate.

FixedResistanceDpM: This model computes, for an open thermo-
dynamic system with two fluid ports, the pressure drop due to
flow friction. The mass flow rate is computed as $k = \dot{m}/\sqrt{\Delta p}$ if
$|\dot{m}| > \dot{m}_t$, where $\dot{m}_t > 0$ is a user parameter that indicates where
turbulent flow starts. By default, $\dot{m}_t > 0$ is set to be 30% of the
design mass flow rate, but this value is user-adjustable. The transi-
tion from the turbulent to the laminar flow regime is also required
to ensure that $d\dot{m}/d\Delta p$ has a bounded derivative near zero mass
flow rate. To compute the pressure drop, the mass flow rate as
opposed to the volume flow rate is used to decouple the computa-
tion of the mass balances from the energy balances. In view of the
small variation in density for building HVAC applications, and the
inherent uncertainty in calculating pressure drop of flow networks,
the introduced error is small.

These three primitive models are used as the basic building blocks to
implement models for heat exchangers, valves, fans and pumps. As
shown in Figure 2, such component models extend from the common
base class `PartialTwoPortInterface`, which implements variables to
access thermodynamic properties at the fluid ports. This is extended
by `PartialResistance`, which is the base class for models that are
used to compute flow resistance of pipes, chillers, boilers, valves and
air dampers. The model `PartialTwoPortInterface` is also extended
by `TwoPortHeatMassExchanger`, which is a core model of the library
that is extended to implement models, such as for a heater or cooler
with a prescribed rate of heat transfer (`HeaterCoolerPrescribed`),
a humidifier (`HumidifierPrescribed`), and a boiler with polynomi-
als for the efficiency curve (`BoilerPolynomial`). The conservation
equations in these models are all implemented using an instance of
`PartialMixingVolume`. This provides a means to add heat and ex-
change mass with a fluid volume that can be parametrized to use dy-
namic or steady-state conservation equations.[45] These conservation
equations are provided by the primitive models `ConservationEquations`
and `StaticTwoPortConservationEquations` described above. The
model `PartialMixingVolume` is a *partial* model, which in Modelica
means that it cannot directly be used because it has fewer equations
than unknowns. Rather, the model `TwoPortHeatMassExchanger` re-
declares this model to be a `MixingVolume`, which means that mass
can only be exchanged through the fluid ports. However, compo-
nent model developers can replace this model with any model that
extends from `PartialMixingVolume`. For example, for a humidi-
fier which adds a prescribed amount of water vapor, the model
`MixingVolume` is not applicable as it does not allow adding water

[45] For certain applications, or for certain
models, approximating the dynamics
by a first order response, as done by
an instantaneously mixed volume, is
insufficient. In such a case, a model
developer can implement a finite vol-
ume model by extending the model
`PartialTwoPortInterface`. For exam-
ple, this is how the U-tube borehole
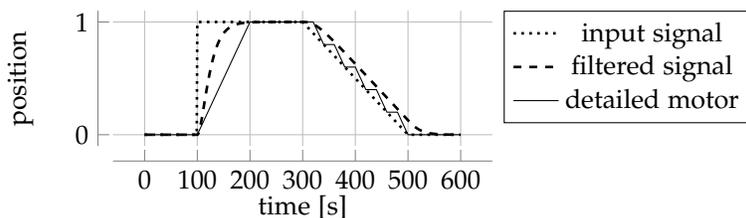heat exchanger model was imple-
mented.

vapor. Therefore, models that need to add or remove water vapor can do so by redeclaring the instance as `MixingVolumeMoistAir`, which contains an input port that allows adding or removing mass to the control volume. In principle, `MixingVolumeMoistAir` can be used for any model by setting the mass exchange to zero. However, `MixingVolume` provides a more efficient implementation than `MixingVolumeMoistAir` if no mass is exchanged with the fluid.

The model `MixingVolume` is also used in various other models, including a model for a transport delay, approximated by a completely mixed volume, and a model for a radiator, which uses a series of `MixingVolume` to implement a dynamic model of a radiator, using the heat transfer equations from the European Norm EN 442-2.

All models for valves extend from the partial model `PartialTwoWayValve`. This partial model implements the pressure drop of the valve based on the flow function, which is defined as

$$\phi(y) = \frac{C_v(y)}{C_v(1)} = \frac{\dot{V}(y)\,\rho/\sqrt{\Delta p}}{\dot{V}(1)\,\rho/\sqrt{\Delta p}} = \frac{\dot{m}(y)}{\dot{m}(1)}, \tag{6}$$

i.e., the ratio of mass flow rate at the current valve position y divided by the mass flow rate at a fully open valve at the same pressure difference across the valve.[46] `PartialTwoWayValve` extends `PartialResistance` and `ActuatorSignal`. `ActuatorSignal` provides the input for the actuator signal, and also an implementation that gives users the option to approximate the travel time of the actuator.

[46] For computing the mass flow rate, we assumed that the variation of mass density is negligible in view of the other idealization of the model and the uncertainty in user-selected parameter values.

Figure 3: Opening position of a valve, computed using different models for the opening position as a function of the input signal. The dotted line shows the input signal. The dashed line shows the input signal filtered by a filter with a rise time of 100 seconds. The solid line shows the output of a detailed motor model with hysteresis and finite travel speed. The opening and closing time of the motor was set to 100 seconds, and the motor hysteresis was set to 0.1.

The travel time is approximated using a second order, critically damped low-pass filter. Using a filter often leads to a more robust simulation, because a step change in the input signal is smoothened by the filter, and hence the flow network receives as an input a signal that is continuously differentiable in time. If the considered application has requirements on the transient response for which neither the instantaneous opening nor the filtered opening is sufficient, then one can use a detailed model of an actuator motor that has a hysteresis, and a finite opening and closing speed. This model, however, generates state events whenever the actuator motor starts or stops adjusting its position, which can significantly increase the computing time of large models. Figure 3 shows the input control signal (dot-

ted), which is equal to the valve position if the valve is configured
using an instantaneous opening, the valve position approximated by
a filter (dashed) and the valve position modeled using the detailed
actuator motor model (solid).

The separation between base classes and ready-to-use compo-
nent models has several advantages: Many component models share
the same parameters, such as parameters that can be used to switch
between steady-state or dynamic models. Developers can add new
components by defining a minimum set of equations, because base
classes already provide a numerically robust and efficient imple-
mentation of steady-state and dynamic conservation equations of
the medium that flows through the component. This leads to faster
model development, and lower risk of introducing errors because
many equations are reused from already tested base classes that
are used across a range of models. Furthermore, code duplication
is largely avoided which reduces maintenance costs. For example,
the following code is sufficient to implement a heat exchanger with
constant effectiveness that exchanges heat between two fluid streams
according to $\dot{Q}(t) = \epsilon\,\dot{Q}_{max}(t)$, where $\epsilon \in (0,1)$ is the heat exchanger
effectiveness and $\dot{Q}_{max}(\cdot)$ is the maximum possible heat transfer.
Line 3 declares that the model extends from a base class that imple-
ments an effective heat transfer model. This base class implements
the four fluid ports and the two flow resistances for the two medium
streams that flow through the heat exchanger. It computes quantities
that are typically required in effectiveness models, such as the capac-
ity flow rates. Lines 4 and 5 declare that the model should only com-
pute sensible heat transfer. Lines 6 and 7 provide equations for the
rate of heat transfer based on the variable QMax_flow which is com-
puted in the base class. Lines 8 and 9 assign the rate of mass transfer
to the variables mXi1_flow and mXi2_flow. This is required for the
model to have an equal number of equations and variables. Lines 10
and 11 declare the parameter for the heat exchanger effectiveness.

```
1   model ConstantEffectiveness
2     "Heat exchanger with constant effectiveness"
3     extends Buildings.Fluid.HeatExchangers.BaseClasses.PartialEffectiveness(
4        sensibleOnly1 = true,
5        sensibleOnly2 = true,
6        Q1_flow = eps * QMax_flow,
7        Q2_flow = -Q1_flow,
8        mXi1_flow = zeros(Medium1.nXi),
9        mXi2_flow = zeros(Medium2.nXi));
10    parameter Real eps(min=0, max=1, unit="1") = 0.8
11      "Heat exchanger effectiveness";
12  end ConstantEffectiveness;
```

Because models extend from the same base class, a model can im-

plement so-called replaceable models to allow replacing one model with a similar one that is compatible. Once users understand what physics is implemented by the base classes, it becomes faster to understand different implementations of component models that extend from the same base class. As a drawback of this model hierarchy, the learning curve for developers in understanding the library structure is somewhat higher. However, we believe that this effort is offset by the above advantages.

5   *Validation*

5.1   *Component and system validation*

Component and system models have been validated using empirical validation, comparative testing and analytical verification. Empirical validation, in which models are compared against measurements, have been used to test the window heat transfer and transmitted solar radiation.[47] Comparative testing, in which results are compared with other simulators, has been used for the following models: In Nouidui et al. (2012)[48], we compared the optical properties and the surface temperatures of window systems with results obtained by the WINDOW 6 program.[49] The building envelope model has also been validated using BESTEST.[50] Air flow rates for the multizone airflow models have been validated with results obtained by the CONTAM program.[51] Analytical verification, in which results are compared with exact solutions, is applicable for component models for which analytical solutions can be derived for certain steady-state or transient boundary conditions. This type of validation has been used to validate most individual component models, such as for heat and mass transfer and storage, for flow resistance elements such as valves and pipes, for fan models and for radiosity transport models.

5.2   *Regression Tests*

Once models are validated, their example models and associated reference results in the form of time series are included in the library and used for regression testing whenever a developer updates the library. The regression tests will run all 279 example models and report various problems, such as translation errors, non-convergence of a model, or a change in results by more than a prescribed tolerance compared to archived reference results. These comparisons to reference results are essential to ensure that changes to base classes have no inadvertent effects, or that developers do not introduce an error in already tested models. The scripts that run the regression tests are included in the BuildingsPy Python package that is described below.

[47] Nouidui, T. S., M. Wetter, , and W. Zuo (2012, August). Validation of the window model of the modelica buildings library. In *Proc. of SimBuild*, Madison, WI. IBPSA-USA

[48] Nouidui, T. S., M. Wetter, , and W. Zuo (2012, August). Validation of the window model of the modelica buildings library. In *Proc. of SimBuild*, Madison, WI. IBPSA-USA

[49] Mitchell, R., C. Kohler, L. Zhu, D. Arasteh, J. Carmody, C. Huizenga, and D. Curcija (2011, January). THERM 6.3 / WINDOW 6.3 NFRC simulation manual. Technical Report LBNL-48255, Lawrence Berkeley National Laboratory, Berkeley, CA

[50] ASHRAE (2007). ANSI/ASHRAE Standard 140-2007, Standard method of test for the evaluation of building energy analysis computer programs; and Nouidui, T. S., K. Phalak, W. Zuo, and M. Wetter (2012, September). Validation and application of the room model of the modelica Buildings library. In *Proc. of the 9-th International Modelica Conference*, Munich, Germany, pp. 727–736. Modelica Association

[51] Wetter, M. (2006, September). Multizone airflow model in Modelica. In C. Kral and A. Haumer (Eds.), *Proc. of the 5-th International Modelica Conference*, Volume 2, Vienna, Austria, pp. 431–440. Modelica Association and Arsenal Research

6   Associated Tools

In some applications, models of this library need to be used within an automated workflow, connected to legacy building simulation programs, or connected to hardware. This section describes tools that have been developed to support such applications.

6.1   Pre- and post-processing

To facilitate the automation of simulation, such as for parametric studies, and to facilitate post-processing, we developed the BuildingsPy Python package. This package contains modules with functions that allow changing from interpreted scripts parameters of models and solvers, translating and simulating models, possibly with the use of the Python multiprocessing package for parallel computing, and reading output files for further processing. For example, after the simulation, time series can be read by a Python script for subsequent visualization and statistical analysis.

6.2   Scripts for Regression Tests

The BuildingsPy package is also used to support developers in running regression tests. Python functions have been implemented that scan the library for executable models, simulate these models using parallel computing, and then compare the new simulation results with archived reference results. If the deviation is larger than a prescribed limit, then a plot will be generated that compares archived versus new results, and the developer will be asked whether the new results should replace the archived results or be rejected. Since there are 330 models, blocks and functions in the library, these regression tests have been shown to be critical in ensuring that developers do not inadvertently introduce an error in validated code.

6.3   Co-simulation and Hardware-in-the-loop Simulation

For certain applications, such as to assess the interaction of facade and HVAC control or to develop model-based control algorithms, Modelica models need to be linked to other simulators such as Radiance or to hardware, such as building automation systems or data acquisition systems for closed loop run-time data exchange. To enable such co-simulation and hardware-in-the-loop simulations, we have been developing the software interfaces that are described below. These interfaces augment the application programming interfaces and model export facilities that are provided by the various Modelica

simulation environments. The authors developed the following two technologies:

To couple Modelica models to other simulators for co-simulation, such as to EnergyPlus or to MATLAB/Simulink, we developed the Building Controls Virtual Test Bed (BCVTB) software.[52] The BCVTB is a software environment that allows expert users to couple different simulation programs for co-simulation and for hardware-in-the-loop simulation. For example, the BCVTB allows users to simulate a building in EnergyPlus while the HVAC and control systems are simulated in Modelica. The BCVTB is based on the Ptolemy II software environment.[53]

While the BCVTB has been primarily developed by LBNL in conjunction with the Ptolemy II developers from the University of California at Berkeley, the MODELISAR project started with the goal that dynamic system models of different software systems can be used together for an integrated simulation, for hardware-in-the-loop simulation and to execute code on processors of embedded systems. This project resulted in the Functional Mockup Interface (FMI) standard.[54] FMI is an open, non-proprietary standard that defines C-header files to interact with the equations of a model or to perform co-simulations with other simulators. It also defines an XML schema that allows an importing tool to inquire information about the model and its interface variables. The FMI interface is independent of the respective tool that exported the model or simulators, which allows exchanging code in an interoperable format. Tools that support FMI can import and/or export a simulator for co-simulation, or a model for model exchange. The exported simulator or model is called a Functional Mockup Unit (FMU). As of 2012, 34 modeling and simulation environments support or plan to support FMI.[55]

In 2012, LBNL released an FMI for co-simulation import interface in EnergyPlus 7.1.[56] Such an interface is currently being implemented in the BCVTB software, and work is ongoing to allow exporting EnergyPlus as an FMU for co-simulation. Benefits of the FMI link, as opposed to the existing simulator coupling through the `Simulator` actor in the BCVTB, are better usability, error handling and reporting.

Figure 4 shows the different connection capabilities in the context of simulators and hardware for buildings. Various Modelica simulation environments allow the import and export of FMUs. This allows, for example, to export an HVAC system from the `Buildings` library, and import it into an EnergyPlus model. Conversely, a building envelope model can be exported from EnergyPlus and imported into Modelica for the development of control sequences. From both environments, FMUs can be exported for import into the BCVTB. From

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

[53] Brooks, C., E. A. Lee, X. Liu, S. Neuendorffer, Y. Zhao, and H. Zheng (2007, January). Ptolemy II – heterogeneous concurrent modeling and design in Java. Technical Report No. UCB/EECS-2007-7, University of California at Berkeley, Berkeley, CA

[54] MODELISAR Consortium (2010b, jan). *Functional Mock-up Interface for Model Exchange* (First ed.). MODELISAR Consortium; and MODELISAR Consortium (2010a, oct). *Functional Mock-up Interface for Co-Simulation* (First ed.). MODELISAR Consortium

[55] FMI (2012). Fmi tool support

[56] EnergyPlus External Interface (2012, May). *External Interface(s) Application Guide, Guide for using EnergyPlus with External Interface(s), version 7.1*
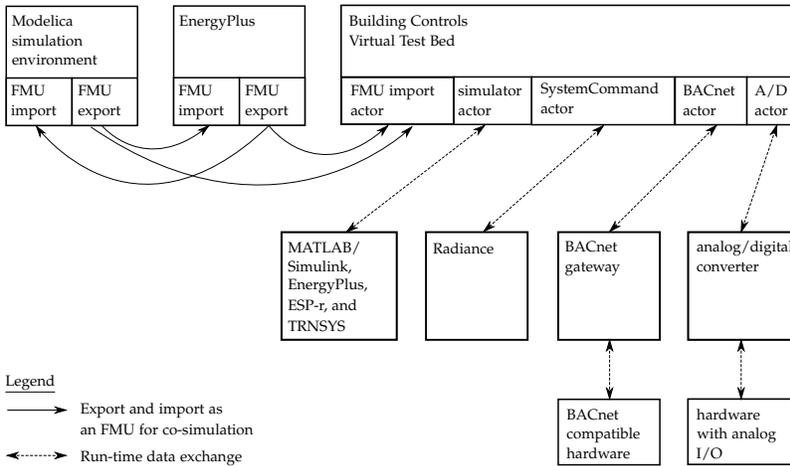
Figure 4: Connectivity between simulators and hardware through simulator export and import as an FMU, and through run-time data exchange with the Building Controls Virtual Test Bed.

the BCVTB, they can be linked to MATLAB/Simulink and Radiance for co-simulation. They can also be linked to hardware through BACnet or analog/digital converters for real-time simulation and for model-based controls.

Modelica models from the Buildings library, exported as FMUs, have also been used in the Learn High Performance Buildings interactive eLearning software.[57] In this project, models are encapsulated as FMUs. Using the FMI interface, model parameters and inputs are set, the time stepping is controlled and model outputs are retrieved. This is implemented in a web-server which uses the simulations to compute realistic responses of building heat transfer, multizone air exchange, HVAC systems, control systems and equipment faults, and visualizes this response at a student's interactive web browser through WebGL.

[57] Deringer, J. J., J. E. Nahman, K. Heming, M. Wetter, X. Pang, and M. Konstantoglou (2012, August). LearnHPB and eLAD - two related online eLearning platforms for high performance buildings. Summer Study on Energy Efficiency in Buildings, Pacific Grove, CA. ACEEE

7 *Example*

We will now present an example that illustrates how Modelica can be used to implement, test and improve the performance of a user-defined control sequence. The control sequence is for a cooling plant with a water-side economizer. It contains a state graph with delay on the state transitions that is used to enable and disable the water side economizer, a continuous time on/off control with event detection and hysteresis to enable and disable the chiller, and discrete time control for the setpoint scheduling. The control sequence is implemented by combining block-diagrams and a state graph with acausal models of the physical components of the chilled water plant.

The objective is to improve the energy efficiency of a chilled water plant by optimizing the control setpoints. The chilled water plant is used to cool a data center in San Francisco, CA, with a constant cooling load of 500 kW.

7.1 *Model of the plant*

We analyzed two models, one with and one without a water-side economizer (WSE). The WSE is a heat exchanger added to the water loop of the chilled water plant to provide free cooling. Figure 5 shows the schematic model and the Modelica system model of the chiller plant with WSE. The room of the data center was modeled using a mixed air volume with a heat source. Heat conduction and air infiltration through the building envelope were neglected since the heat exchange between the room and the ambient environment was small compared to the heat released by the computers. The chiller model is based on performance curves and has a variable speed compressor. The cooling tower uses a performance curve to compute the approach temperature at off-design conditions. The water-side economizer is a steady-state heat exchanger model with constant effectiveness of 0.8. The chilled water flow rate is variable and controlled as described by (15) below. The supply air mass flow rate to the data center is constant, and the supply air is cooled using a finite element cooling coil model.

7.2 *Model of the control algorithm*

The control objective was to maintain the temperature of the supply air to the room, while reducing energy consumption of the chilled water plant. The control was based on the control sequence proposed by Stein (2009).[58] To simplify the implementation, we only applied the controls for the differential pressure of the chilled water loop, the setpoint temperature of the chilled water leaving the chiller, and

[58] Stein, J. (2009). Waterside economizing in data centers: Design and control considerations. *ASHRAE Transactions 115*(2), 192–200

(a) Schematic diagram

(b) Modelica model

Figure 5: Schematic diagram and Modelica system model of the chilled water plant with WSE.

the chiller on/off control. For the system with WSE, there was an additional enable/disable control for the WSE as recommended by Taylor (2010).[59] This control sequence was independent from the other control sequences. The pump for the condenser water loop, the cooling tower fan and the supply air fan were running at constant speeds.

The chiller is switched on if

$$T_{chw,ent}(t) > T_{chi,set}(t) + \Delta T_{chi,ban} \tag{7}$$

and switched off if

$$T_{chw,ent}(t) \leq T_{chi,set}(t), \tag{8}$$

where $T_{chw,ent}(t)$ is the temperature of the chilled water entering the chiller, $T_{chi,set}(t)$ is the setpoint temperature of the chilled water leaving the chiller, and $\Delta T_{chi,ban} = 2.2$ K is the dead-band temperature to prevent chiller short cycling.

Figure 6(a) shows the state graph of the WSE control. The WSE is enabled if

$$\Delta t_{wse,off}(t) \geq 20 \text{ min}, \tag{9}$$

and

$$T_{ws}(t) > 0.9 \ T_{wet}(t) + \Delta T_t(t) + \Delta T_w(t), \tag{10}$$

with temperatures in Kelvin, where $\Delta t_{wse,off}(t)$ is the time of the WSE in the off status, $T_{ws}(t)$ is the temperature of the chilled water leaving the cooling coil, $T_{wet}(t)$ is the wet bulb temperature, $\Delta T_t(t)$ is the temperature difference between the water leaving the cooling tower and the air entering the cooling tower, $\Delta T_w(t)$ is the temperature difference between the chilled water leaving the WSE and the condenser water entering the WSE. The condition (10) must remain true for at least 20 min.

The WSE is disabled if

$$\Delta t_{wse,on}(t) \geq 20 \text{ min}, \tag{11}$$

and

$$T_{ws}(t) < T_{wc}(t) + \Delta T_{wse,off}, \tag{12}$$

where $\Delta t_{wse,on}(t)$ is the time elapsed since the WSE was switched on, $T_{wc}(t)$ is the return water temperature of the cooling tower and $\Delta T_{wse,off} = 0.6$ K is an offset temperature. The condition (12) must remain true for at least 20 min.

The state graph is implemented in the Modelica model graphically as shown in Figure 6(b). On the left are the connectors for the control input signals, in the middle is the state graph and on the right are signal conversions from boolean to real signals, followed by the connectors for the output signals. In this simple state graph there are two states represented by the oval icons. The states are "off" or "on". The curved arrow indicates that "off" is the initial state. The horizontal bars are the transitions between the states. Each transition must have exactly one preceding state and one succeeding state. The transitions fire if a boolean expression, i.e., equations (10) and (12), respectively, remains true for 1200 seconds. This optional wait time is indicated by the red annotation to the right of the transitions.

(a) State graph of the WSE control.
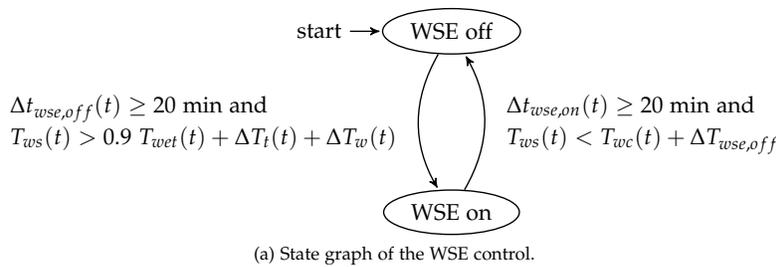
Figure 6: State graph of the WSE and its implementation in Modelica.

(b) Implementation of the state graph of the WSE control.
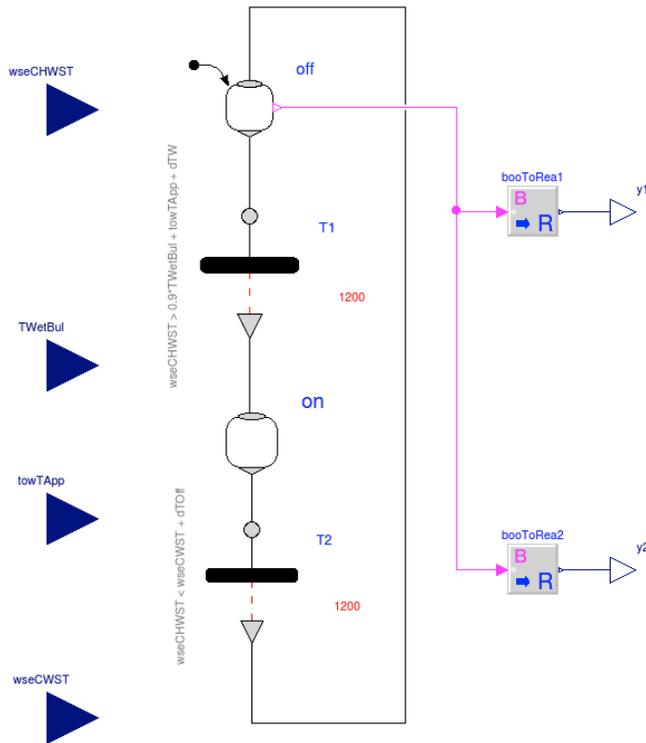
The setpoint reset strategy is to first increase the differential pres-

sure $\Delta p(t)$ of the chilled water loop to increase the chilled water flow rate. If $\Delta p(t)$ reaches the maximum value and further cooling is needed, then the chiller temperature setpoint $T_{chi,set}(t)$ is reduced. If there is too much cooling, then $T_{chi,set}(t)$ and $\Delta p(t)$ will be changed in the reverse direction. This control strategy is realized using the trim and response logic[60], which is as follows: A cooling request is triggered if the input signal $y(t)$ is larger than a threshold y_δ. We used $y_\delta = 0$ and

$$y(t) = T_{air,sup}(t) - T_{air,set}, \tag{13}$$

where $T_{air,sup}(t)$ and $T_{air,set}$ are the actual and the set point temperatures of the supply air to the room. The sampling period was $\Delta t = 120$ seconds. The control signal $u(t)$ is computed as

$$u(t) = \begin{cases} \min(1, \max(0, u(t - \Delta t) + \Delta u_i)), & \text{if } y(t) \geq y_\delta, \\ \min(1, \max(0, u(t - \Delta t) - \Delta u_d)), & \text{otherwise}, \end{cases} \tag{14}$$

where $\Delta u_i = 0.03$ and $\Delta u_d = 0.03$ are parameters for increasing and decreasing $u(t)$.

As shown in Figure 7, the control signal $u(t)$ is converted to control setpoints for $\Delta p(t)$ and $T_{chi,set}(t)$ as

$$\Delta p(t) = \begin{cases} \Delta p_{min} + (\Delta p_{max} - \Delta p_{min}) \frac{u(t)}{x}, & \text{if } u(t) \in [0, x], \\ \Delta p_{max}, & \text{otherwise}, \end{cases} \tag{15}$$

and

$$T_{chi,set}(t) = \begin{cases} T_{chi,max}, & \text{if } u(t) \in [0, x], \\ T_{chi,max} - (T_{chi,max} - T_{chi,min}) \frac{u(t) - x}{1 - x}, & \text{otherwise}, \end{cases} \tag{16}$$

where $x \in (0, 1)$ is a user-selected parameter, Δp_{min} and Δp_{max} are the minimum and maximum differential pressure of the chilled water loop, and $T_{chi,min}$ and $T_{chi,max}$ are the minimum and maximum values of $T_{chi,set}(t)$. For our experiments, we used $x = 0.5$, $\Delta p_{min} = 25.94$ kPa, $\Delta p_{max} = 129.7$ kPa and $T_{chi,min} = 5.56°C$, which is the minimum value in the range of performance curve of the selected chiller. We varied the value of $T_{chi,max}$ for a parametric study to find the optimal value to reduce the system energy consumption. The details will be discussed in the following paragraph.

To optimize the control setpoints for system energy efficiency, we conducted a parametric study in which we varied two control setpoints for the chilled water plants with and without WSE. One parameter was the setpoint temperature for the supply air, $T_{air,set}$.
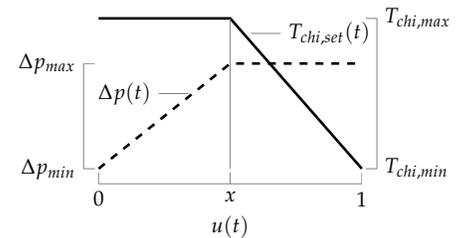
Figure 7: Chilled water loop differential pressure and chiller setpoint temperature reset.

compensated for by the trim and response logic. Second, for the system with WSE, increasing $T_{chi,max}$, and hence $T_{chi,set}(t)$, reduces the PUE because by (7) and (8), this increases the hours when the chiller is off.

For both systems, increasing $T_{air,set}$ also reduces the PUE if $T_{chi,max}$ is larger than a certain value. The reason is that a higher air supply temperature leads to an increase in chilled water temperature, thereby increasing the coefficient of performance of the chiller and the number of hours during which the economizer can be used.

There are some oscillations on the vertical parts of the contour lines, for example at PUE = 1.17 and 1.18 in Figure 9(a) and at PUE = 1.23 to 1.25 in Figure 9(b). This is caused by the large values of $\Delta u_i = 0.03$ and $\Delta u_d = 0.03$ used by the trim and response logic in equation (14). Further numerical experiments showed that reducing the values of Δu_i and Δu_d led to smoother PUE contour lines in Figure 9, but they caused over-heating and over-cooling of the supply air to the room. Since the control objective was that the temperature of the supply air $T_{air,sup}(\cdot)$ tracks the setpoint $T_{air,set}$, we used $\Delta u_i = 0.03$ and $\Delta u_d = 0.03$.
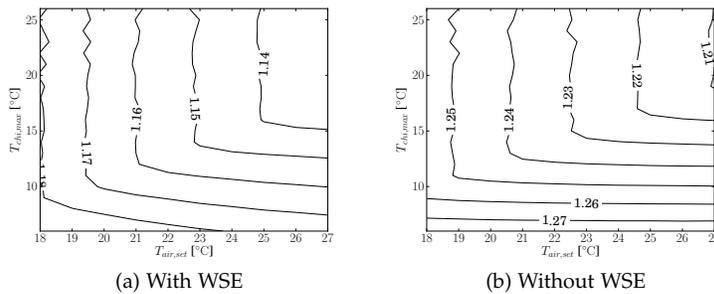

(a) With WSE   (b) Without WSE

Figure 9: PUEs of chilled water plant systems with and without WSE.

Figure 10 shows for the system with WSE the outside wet bulb temperature $T_{wet}(t)$ and the hours of corresponding chiller status for a whole year. When $T_{chi,max} = 6°C$, the chiller was running for the whole year to meet the low setpoint temperature (Figures 10(a) and 10(b)). That explains why the PUEs for $T_{chi,max} = 6°C$ are the highest among the system with WSE in Figure 9(a).

Comparing Figure 10(a) with Figure 10(b) and Figure 10(c) shows that increasing $T_{chi,max}$ from 6°C to 22°C, or increasing $T_{air,set}$ from 18°C to 27°C alone does not increase the hours when the chiller is off. Both setpoints need to be increased to achieve a reduction in chiller operating hours.

The Modelica models can also provide detailed transient information of the chilled water plant control. Figure 11(a) shows the control signals of the system with WSE under two different control setpoints
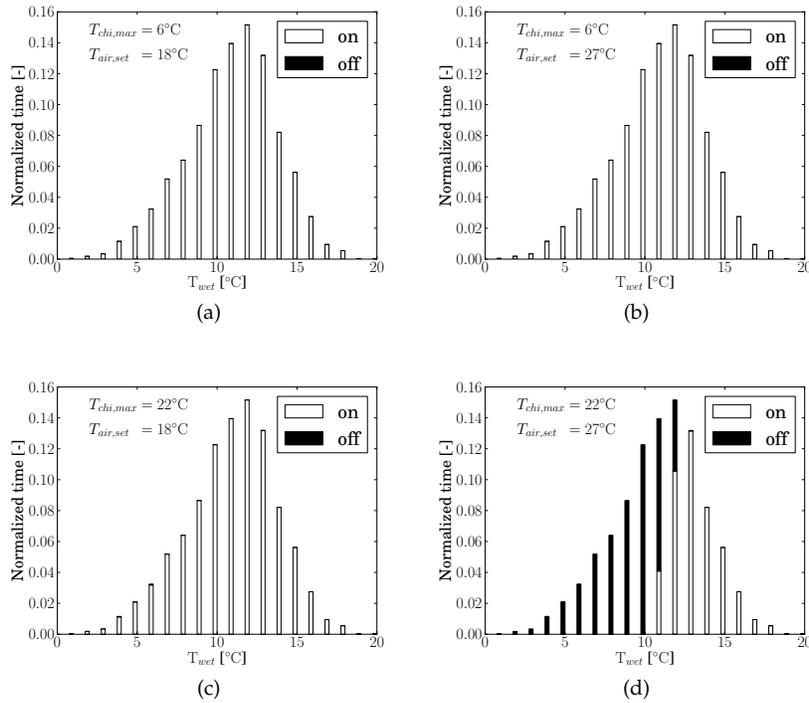
Figure 10: Annual frequency distributions of the chiller status as a function of the wet bulb temperature for different control setpoints. All figures are for the system with WSE.

on a cold day (January 14). On the cold day, the WSE can provide all the cooling needed and the chiller can be off for the whole day if the setpoint temperatures are high. With low setpoint temperatures, the chiller has to be on to cool the chilled water to the low value of $T_{chi,set}(t)$. The small oscillations of $\Delta p(t)$ are caused by the trim and response logic which samples every 120 seconds.

Figure 11(b) shows the same variables for June 30, a hot day. With low setpoint temperatures, the chiller was on all the time. With high setpoint temperatures, the chiller was off in the morning. As the outside wet bulb temperature raises during the morning, the rate of heat transfer of the economizer is reduced. This is compensated for by an increase of the pressure setpoint $\Delta p(t)$ for the chilled water loop, thereby increasing the chilled water flow rate. In the early afternoon, this is insufficient, leading to a reduction of the chilled water set point $T_{chi,set}(t)$, according to Figure 7, after which the chiller switches on and remains on until the early evening.
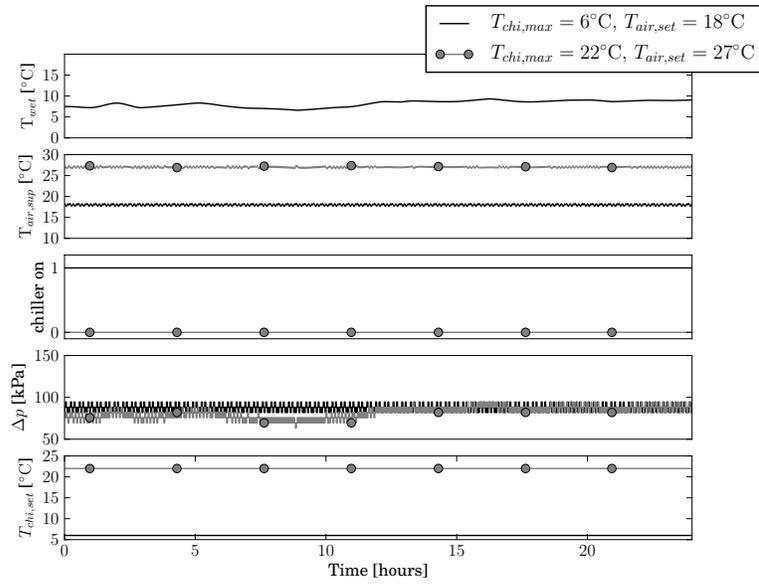
(a) Cold day
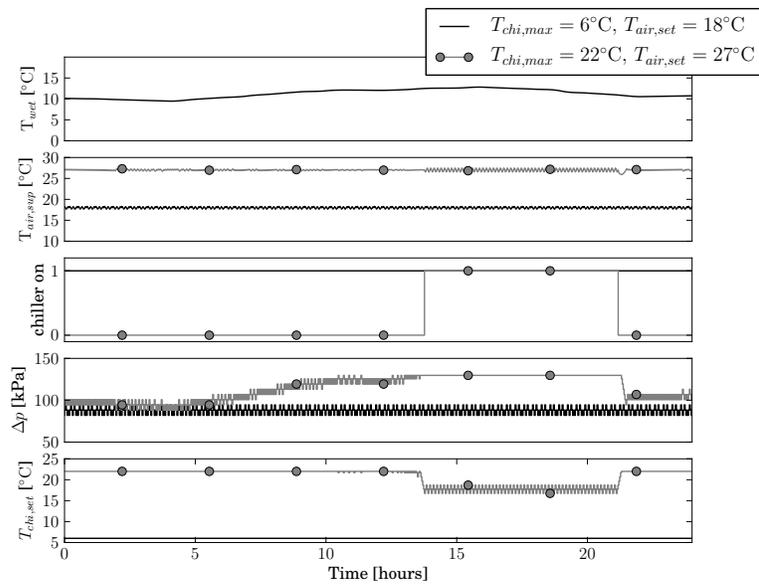
(b) Hot day

Figure 11: Trajectories for the system with WSE.

8 Conclusions

Our experience in developing the `Buildings` library has been that diligent design of the basic models, such as for conservation equations, and of the class inheritance tree is important. Benefits for the users include clarity of modeling assumptions as models of similar equipment use the same base classes. Sharing common base classes also makes it easy to see actual differences between similar models, which would be hard if two similar models were to contain tens or hundreds of lines of duplicated code. For developers, benefits include reduced development time, as adding new models requires little additional code that needs to be validated. Avoiding code duplication also reduces maintenance efforts because errors have to be fixed in only one place and not across a variety of models. The drawback of a large object-inheritance tree is that developers need to invest more time to understand the library structure if they want to reuse existing functionalities when implementing new models.

Also shown to be valuable are the availability of tools for exporting models, interfaces for run-time data exchange with other tools, and scripts for automating simulations and post-processing. Computing and exporting linearized models allows use of models in conjunction with model order reduction algorithms for subsequent stability analysis in the frequency domain. Exporting the full model as a Functional Mockup Unit facilitated embedding Modelica models in the interactive Learn High Performance Buildings eLearning software. Interfaces for co-simulation allowed use of the library in situations where run-time data exchange was needed with other tools that may simulate a domain for which models in Modelica do not exist. Scripting interfaces for automating simulations and post-processing results showed to be useful for both, creating a repeatable processes for use of simulation during project work, and to conduct regression tests during model development.

9 Acknowledgments

10   *References*

ASHRAE (2006). *Sequences of Operation for Common HVAC Systems*. Atlanta, GA: ASHRAE.

ASHRAE (2007). ANSI/ASHRAE Standard 140-2007, Standard method of test for the evaluation of building energy analysis computer programs.

ASHRAE TC 9.9 (2011). Thermal guidelines for data processing environments - expanded data center classes and usage guidance. White paper, ASHRAE.

Badakhshani, A. and D. Müller (2010, September). Dynamische exergetische Analyse von Gebäuden anhand der Gesamtsystemsimulation in der objekt-orientierten programmiersprache Modelica. In *BauSim*, Vienna, Austria. IBPSA Austria-Germany.

Belady, C., A. Rawson, J. Pflueger, and T. Cader (2008). Green grid data center power efficiency metrics: PUE and DCIE. Technical report, The Green Grid. White Paper.

Björsell, N., A. Bring, L. Eriksson, P. Grozman, M. Lindgren, P. Sahlin, A. Shapovalov, and M. Vuolle (1999, September). IDA indoor climate and energy. In N. Nakahara, H. Yoshida, M. Udagawa, and J. Hensen (Eds.), *Proc. of the 6-th IBPSA Conference*, Kyoto, Japan, pp. 1035–1042.

Bonvini, M. and A. Leva (2011). Object-oriented sub-zonal modelling for efficient energy-related building simulation. *Mathematical and Computer Modelling of Dynamical Systems 17*(6), 543–559.

Brooks, C., E. A. Lee, X. Liu, S. Neuendorffer, Y. Zhao, and H. Zheng (2007, January). Ptolemy II – heterogeneous concurrent modeling and design in Java. Technical Report No. UCB/EECS-2007-7, University of California at Berkeley, Berkeley, CA.

Brown, J. (2010, November). *Computational fluid dynamics in an equation-based, acausal modeling environment*. Ph. D. thesis, Georgia Institute of Technology, Atlanta, GA.

Buhl, W., A. Erdem, F. Winkelmann, and E. Sowell (1993, August). Recent improvements in SPARK: Strong component decomposition, multivalued objects,and graphical interface. In *Proc. of the 3-rd IBPSA Conference*, Adelaide, Australia, pp. 283–291.

Carli, Inc. (2006, October). *TARCOG: Mathematical Models for Calculation of Thermal Performance of Glazing Systems with or without Shading Devices*. Amherst, MA: Carli, Inc.

Charlesworth, P., J. A. Clarke, G. Hammond, A. Irving, K. James, B. Lee, S. Lockley, R. D. Mac, D. Tang, T. J. Wiltshire, and A. J. Wright (1991, August). The energy kernel system. In J. A. Clarke, J. W. Mitchell, and R. C. V. de Perre (Eds.), *Proc. of the IBPSA Conference*, Nice, France.

Coddington, E. A. and N. Levinson (1955). *Theory of ordinary differential equations*. McGraw-Hill Book Company, Inc., New York-Toronto-London.

Deringer, J. J., J. E. Nahman, K. Heming, M. Wetter, X. Pang, and M. Konstantoglou (2012, August). LearnHPB and eLAD - two related online eLearning platforms for high performance buildings. Summer Study on Energy Efficiency in Buildings, Pacific Grove, CA. ACEEE.

Dols, W. S. and G. N. Walton (2002, November). CONTAM 2.0 user manual, multizone airflow and contaminant transport analysis software. Technical Report NISTIR 6921, National Institute of Standards and Technology.

Eisenhower, B., K. Gasljevic, and I. Mezić (2012, August). Control-oriented modeling and calibration of building energy models using modelica. In *Proc. of SimBuild*, Madison, WI. IBPSA-USA.

Elmqvist, H. (1978). *A structured model language for large continuous systems*. Ph. D. thesis, Lund Institute of Technology, Lund, Sweden.

EnergyPlus External Interface (2012, May). *External Interface(s) Application Guide, Guide for using EnergyPlus with External Interface(s), version 7.1*.

Felgner, F., S. Agustina, R. C. Bohigas, R. Merz, and L. Litz (2002, March). Simulation of thermal building behaviour in modelica. In M. Otter (Ed.), *Proceedings of the 2nd Modelica conference*,

Oberpfaffenhofen, Germany, pp. 147–154. Modelica Association and Deutsches Zentrum fur Luft- und Raumfahrt.

FMI (2012). Fmi tool support.

Franke, R., F. Casella, M. Otter, K. Proelss, M. Sielemann, and M. Wetter (2009, September). Standardization of thermo-fluid modeling in `Modelica.Fluid`. In F. Casella (Ed.), *Proc. of the 7-th International Modelica Conference*, Como, Italy. Modelica Association.

Franke, R., F. Casella, M. Otter, M. Sielemann, H. Elmqvist, S. E. Mattsson, and H. Olsson (2009, September). Stream connectors – an extension of modelica for device-oriented modeling of convective transport phenomena. In F. Casella (Ed.), *Proc. of the 7-th International Modelica Conference*, Como, Italy. Modelica Association.

Hoh, A., T. Haase, T. Tschirner, and D. Müller (2005, March). A combined thermo-hydraulic approach to simulation of active building components applying modelica. In G. Schmitz (Ed.), *Proceedings of the 4th Modelica conference*, Hamburg, Germany. Modelica Association and Hamburg University of Technology.

LBNL (2012). User's guide of the modelica Buildings library.

Low, D. and E. Sowell (1982, September). ENET, a PC-based building energy simulation system. In *Energy Programs Conference*, Austin, Texas, pp. 2–7. IBM Real Estate and Construction Division.

Mattsson, S. E. and H. Elmqvist (1997, April). Modelica – An international effort to design the next generation modeling language. In L. Boullart, M. Loccufier, and S. E. Mattsson (Eds.), *7th IFAC Symposium on Computer Aided Control Systems Design*, Gent, Belgium.

Merz, R. M. (2002, September). *Objektorientierte Modellierung thermischen Gebäudeverhaltens*. Ph. D. thesis, Universität Kaiserslautern.

Mitchell, R., C. Kohler, L. Zhu, D. Arasteh, J. Carmody, C. Huizenga, and D. Curcija (2011, January). THERM 6.3 / WINDOW 6.3 NFRC simulation manual. Technical Report LBNL-48255, Lawrence Berkeley National Laboratory, Berkeley, CA.

MODELISAR Consortium (2010a, oct). *Functional Mock-up Interface for Co-Simulation* (First ed.). MODELISAR Consortium.

MODELISAR Consortium (2010b, jan). *Functional Mock-up Interface for Model Exchange* (First ed.). MODELISAR Consortium.

Nouidui, T., K. Sedlbauer, C. Nytsch-Geusen, and K. Kießl (2009, October). Neue objektorientierte hygrothermische modellbibliothek

zur ermittlung des hygrothermischen und hygienischen komforts in räumen. *Bauphysik 31*(5), 271–278.

Nouidui, T. S., K. Phalak, W. Zuo, and M. Wetter (2012, September). Validation and application of the room model of the modelica Buildings library. In *Proc. of the 9-th International Modelica Conference*, Munich, Germany, pp. 727–736. Modelica Association.

Nouidui, T. S., M. Wetter, , and W. Zuo (2012, August). Validation of the window model of the modelica buildings library. In *Proc. of SimBuild*, Madison, WI. IBPSA-USA.

Nytsch-Geusen, C., T. Nouidui, A. Holm, , and W. Haupt (2005, August). A hygrothermal building model based on the object-oriented modeling language modelica. In I. Beausoleil-Morrison and M. Bernier (Eds.), *Proceedings of the Ninth International IBPSA Conference*, Volume 1, Montreal, Canada, pp. 867–876. International Building Performance Simulation Association and Ecole Polytechnique de Montreal.

Polak, E. (1997). *Optimization, Algorithms and Consistent Approximations*, Volume 124 of *Applied Mathematical Sciences*. Springer Verlag.

Provan, G. and A. Venturini (2012, September). Stochastic simulation and interference using modelica. In *Proc. of the 9-th International Modelica Conference*, Munich, Germany. Modelica Association.

Sahlin, P. and E. F. Sowell (1989, June). A neutral format for building simulation models. In *Proceedings of the Second International IBPSA Conference*, Vancouver, BC, Canada, pp. 147–154.

Schulze, C., M. Gräber, M. Huhn, and U. Grätz (2011, March). Real-time simulation of vapour compression cycles. In *Proc. of the 8-th International Modelica Conference*, Dresden, Germany, pp. 48–55. Modelica Association.

Silverman, G., S. Jurovics, D. Low, and E. Sowell (1981). Modeling and optimization of HVAC systems using network concepts. *ASHRAE Transactions 87*(2), 585–597.

Sowell, E., K. Taghavi, H. Levy, and D. Low (1984). Generation of building energy system models. *ASHRAE Transactions 90*(1B), 573–586.

Sowell, E. F., W. F. Buhl, A. E. Erdem, and F. C. Winkelmann (1986, September). A prototype object-based system for HVAC simulation. Technical Report LBL-22106, Lawrence Berkeley National Laboratory.

Sowell, E. F., W. F. Buhl, and J.-M. Nataf (1989, June). Object-oriented programming, equation-based submodels, and system reduction in SPANK. In *Proceedings of the Second International IBPSA Conference*, Vancouver, BC, Canada, pp. 141–146.

Stein, J. (2009). Waterside economizing in data centers: Design and control considerations. *ASHRAE Transactions 115*(2), 192–200.

SuPerB (2012). Learnhvac educational software.

Taylor, S. T. (2007, June). Increasing efficiency with VAV system static pressure setpoint reset. *ASHRAE Journal*, 24–32.

Taylor, S. T. (2010, October). Personal communication on October 22, 2010.

Tummescheit, H., J. Eborn, and K. Prölss (2005, March). AirConditioning - a Modelica library for dynamic simulation of AC systems. In G. Schmitz (Ed.), *Proceedings of the 4th Modelica conference*, Hamburg, Germany, pp. 185–192. Modelica Association and Hamburg University of Technology.

Wetter, M. (2006, September). Multizone airflow model in Modelica. In C. Kral and A. Haumer (Eds.), *Proc. of the 5-th International Modelica Conference*, Volume 2, Vienna, Austria, pp. 431–440. Modelica Association and Arsenal Research.

Wetter, M. (2009a). A modelica-based model library for building energy and control systems. In P. A. Strachan, N. J. Kelly, and M. Kummert (Eds.), *Proc. of the 11-th IBPSA Conference*, pp. 652–659. International Building Performance Simulation Association and University of Strathclyde.

Wetter, M. (2009b, June). Modelica-based modeling and simulation to support research and development in building energy and control systems. *Journal of Building Performance Simulation 2*(2), 143–161.

Wetter, M. (2011a). *Building Performance Simulation for Design and Automation*, Chapter The Future of Building System Modeling and Simulation, pp. 481–504. Oxon, OX: Taylor & Francis.

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

Wetter, M., W. Zuo, and T. S. Nouidui (2011a, November). Modeling of heat transfer in rooms in the Modelica "Buildings" library. In *Proc. of the 12-th IBPSA Conference*, pp. 1096–1103. International Building Performance Simulation Association.

Wetter, M., W. Zuo, and T. S. Nouidui (2011b, March). Recent developments of the modelica buildings library for building heating, ventilation and air-conditioning systems. In *Proc. of the 8-th International Modelica Conference*, Dresden, Germany, pp. 266–275. Modelica Association.