

Towards Digital and Performance-Based Supervisory HVAC Control Delivery

Amir Roth, US Department of Energy, Building Technologies Office

Michael Wetter, Lawrence Berkeley National Laboratory

Kyle Benne, National Renewable Energy Laboratory

David Blum, Lawrence Berkeley National Laboratory

Yan Chen, Pacific Northwest National Laboratory

Gabriel Fierro, Colorado School of Mines and National Renewable Energy Laboratory

Marco Pritoni, Lawrence Berkeley National Laboratory

Avijit Saha, National Renewable Energy Laboratory

Draguna Vrabie, Pacific Northwest National Laboratory

ABSTRACT

Upgrading supervisory HVAC control in commercial buildings is one of the most attractive decarbonization tools at our disposal. Modern controls are software programs and can in theory be deployed at scale and with a low up-front carbon “pulse”. In practice, however, control delivery is a disjointed and inefficient process, dominated by manual handoffs of imprecise English language documents. A particularly high barrier exists between control implementation and building energy modeling (BEM) which results in control sequences typically not being tested for correctness or performance before implementation.

Together with industry partners, DOE and the national labs are developing an ecosystem of tools and standards that can support fully digital performance-based control delivery workflows. This paper describes this ecosystem, which consists of three mutually supportive efforts. Semantic models of buildings and their systems enable automatic configuration and installation of control software. Platform-neutral control descriptions separate control algorithms from control platforms and enable the creation of libraries of reference control implementations. Dynamic whole-building energy-control simulation that can execute physically realistic control sequences makes it possible to test and evaluate the performance of control sequences and then directly compile them for installation and execution in control systems.

In addition to digitizing and streamlining project-level control delivery, these standards and related software support benchmarking of control algorithms, both rule-based and optimization-based, and help to both advance the state of the art and to implement ratings and programs that encourage the adoption of high-performance control.

Introduction

Supervisory control of building HVAC systems is a powerful tool for decarbonizing the built environment. Supervisory control upgrades can be highly effective. High-performance sequences such as those described in ASHRAE Guideline 36 (ASHRAE 2021) can deliver an average 31% annual energy savings for a range of building use types, HVAC and water heating system types, and climate zones (Zhang 2021). Optimization-based predictive control can deliver even higher savings. In theory, control upgrades can be deployed at scale faster, at lower cost, with less disruption, and with a lower up-front carbon “pulse” than envelope and most HVAC

system upgrades. And if done with a high degree of interoperability and automation, up-front investments are not sunk by future envelope or system upgrades as the strategies can be efficiently re-parameterized and re-optimized for the new configuration. Supervisory controls can be especially helpful during system transitions to fully electrified configurations as they can maximize use and efficiency of heat-pumps while minimizing that of fuel-fired backup heating.

Unfortunately, the most important words in the previous paragraph are “theory” and “if”. In practice, control installations and upgrades are currently delivered with little to no automation and interoperability, increasing costs and limiting the scale of deployment. Installations are bespoke. Vendor and installer lock-in is common. Control sequences are specified by mechanical designers in (English) language documents, typically without testing or quantitative performance evaluation. Specifications may be ambiguous, incomplete, or faulty. Control implementers may make interpretation errors or simply fall back to low-performance sequences with which they are familiar. Optimization-driven control is limited to large or showcase projects.

Over the past five years, the Department of Energy (DOE) and the national labs have worked with industry partners and ASHRAE to develop standards and tools that can support digital performance-based control delivery. This ecosystem attempts to link together the relevant steps of the building design and operation process: HVAC system design, ii) control strategy selection, optimization, and specification, iii) control implementation and installation, and iv) commissioning. In addition to streamlining delivery for individual projects, it supports development, codification, and broad deployment of high-performance control sequences for common systems such as those described in ASHRAE Guideline 36. Finally, it enables the creation of a level playing field for evaluating control algorithms, supporting both research to advance the state-of-the-art and benchmarking and rating to promote adoption.

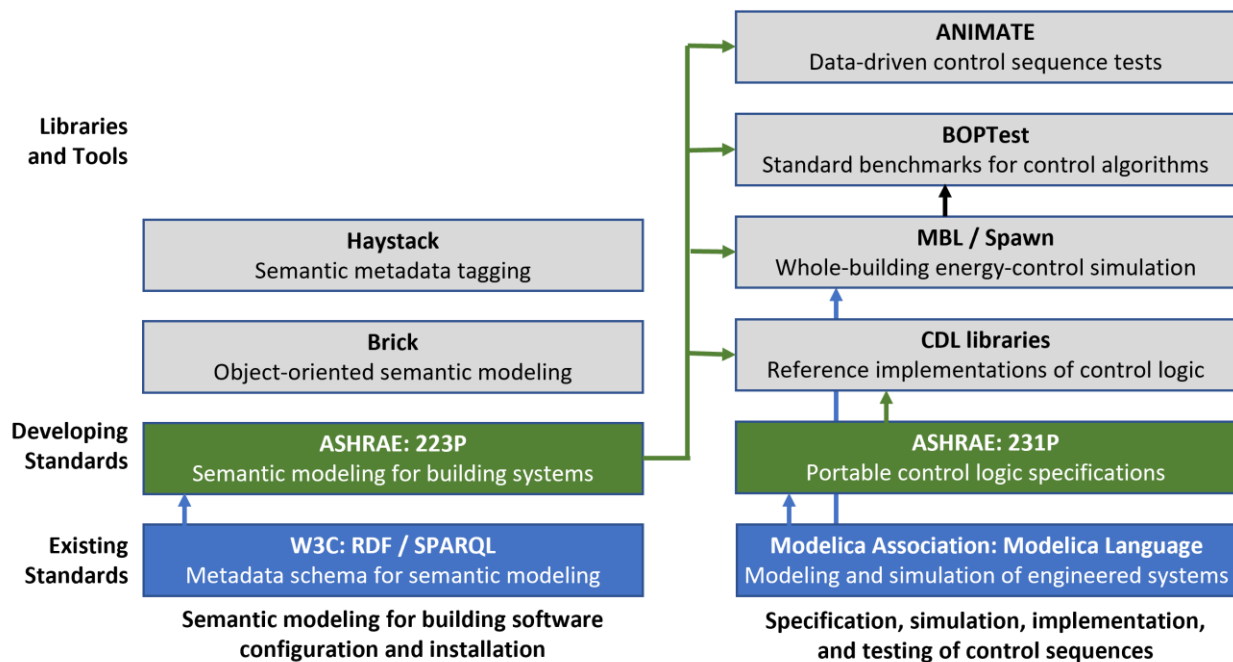


Figure 1. Ecosystem being developed to enable digital performance-driven control delivery. The work centers on two related areas: i) semantic modeling of buildings and systems to automate software configuration and installation, and ii) specification, simulation, implementation, testing, and benchmarking of control sequences.

Figure 1 shows the basic components of this ecosystem. A significant enabling capability is whole-building energy-control simulation that can simulate physically-realistic control logic—this capability relies on the Modelica Language Standard. DOE and the labs are also supporting the development of two new complementary standards: ASHRAE Standard 231P for portable descriptions of control sequences and ASHRAE Standard 223P for semantic modeling of buildings and their systems. 231P also builds on Modelica while 223P builds on the semantic web Resource Descriptor Format (RDF) standards. This paper describes this ecosystem, its intended benefits, and provides a status update on development.

Scale, Potential Impact, and Barriers to Supervisory HVAC Control

Supervisory control is the general term for external “master” control that coordinates the operation of multiple devices or subsystems, such as a cooling plant, multiple air handler units, and multiple room-level terminals. Supervisory control is typically the top level in a hierarchy where each subsystem has its local controller. It typically has high-level objectives such as reducing energy use or shifting demand away from peak periods and considers system-level information such as weather, occupancy, and grid signals. Supervisory control implements its decisions by communicating high-level parameters like outlet temperature setpoints to subsystem controllers. Subsystem controllers—also called local loop controllers—track these high level parameters by adjusting lower-level parameters like valve positions using local information like inlet temperature. In large packaged units, supervisory control may be factory configured. For built-up HVAC systems typical in medium and large commercial buildings, supervisory control is generally specified by an HVAC designer, implemented by a control provider, and verified by a commissioning agent.

Potential performance impact of supervisory control. The potential impact of supervisory control on HVAC system performance is great. Consider multi-zone Variable Air Volume (VAV) systems which save energy by reducing airflow—and fan power—at part load heating and cooling demand. An analysis of the 2012 Commercial Buildings Energy Consumption Survey (CBECS) shows that, in the US, these systems cover 86% of the floor area in large commercial buildings as well over 50% of the floor area in medium office buildings, hotels, schools, and healthcare facilities. Mansson and MacIntyre (1997) found that implementation of strategies including zone temperature set points, supply air temperature reset, boiler temperature reset, economizer control, night-flush ventilation and optimal start can impact annual energy use by 45%. Pang, Piette, and Zhou (2017) found that these strategies and others that make up “good practice” such as static pressure reset, terminal box minimum airflow and optimal start can result in savings of 12% over “common practice” and 64% over “poor practice”. More recently, Zhang et al. (2021) found that the ASHRAE Guideline 36 sequences provide average savings of 31% for multi-zone VAV systems across a wide range of building configurations including varying climates, internal load densities, and operating schedules. Significant savings are also possible for hydronic plants, which are common in large buildings.

Barriers to Deploying High-Performance Supervisory Control at Scale. Fully twenty years ago, Barwig et al. (2002) showed that in commercial buildings with built-up HVAC systems, programming errors are the leading cause of performance and even functional problems related

to supervisory control. The inability to test, debug, and evaluate control sequences virtually leaves these important tasks to installation time, where testing and debugging is more difficult and time consuming.

Since then, HVAC systems, the requirements placed on them in terms of energy use and load management, and the control sequences used to achieve these objectives have only become more complex. Guideline 36 control sequences require six to seven times more code than sequences previously published by ASHRAE (Wetter, 2022). This complexity is only expected to increase with the recent push to decarbonize buildings via a combination of electrification, storage, and load shifting. Systems that operate at higher thermodynamic efficiencies like heat-pumps are more sensitive to temperature setbacks and auxiliary energy use, e.g., by pumps, and are more difficult to control for energy efficiency (Schluck, 2015; Sommer, 2020; Wetter, 2019c). A process that was problematic for simpler control strategies of twenty years ago cannot be expected to work well for the new, more complex systems with which designers, installers and commissioning agents have even less experience.

Performance Evaluation: Whole-Building Energy-Control Simulation

The control delivery process ideally starts with evaluating alternative strategies, choosing the one that most closely meets project goals such as minimizing annual energy use or operational carbon emissions, or occupant discomfort. Control evaluation can include further optimization for the specific project. And it can and should include ensuring that control is robust and stable i.e., that operation does not exhibit oscillatory behavior (Wetter, 2019c). These evaluations require whole-building energy modeling (BEM).

Unfortunately, conventional BEM engines like DOE-2 and EnergyPlus (Crawley, 2001) are not entirely up to the task. Primarily targeted at annual energy use calculations, these engines all employ a simulation approach that involves simulating control in a way that differs from the way control is implemented in the physical world in terms of logic, input/output relationships, and timing. BEM engines use a quasi-static approach to simulating HVAC and controls. Given a starting state and the thermal loads that are experienced over a time interval they can construct the state at the end of the interval. However, they cannot reason about the actions and path taken to get from one state to another. In this regime, control actions are applied retroactively over the interval and are the results of the desired end state, not of control inputs. At a high level, this is the process EnergyPlus takes to simulate the effects of control within a time step: i) it tabulates the heat added to a zone during the time step, ii) if that heat causes the zone temperature to exceed the cooling setpoint, it “reasons” that the cooling system must have activated during the interval to remove the excess heat, iii) it calculates the cooling system’s required average state over the interval to remove the heat, iv) it calculates the zone’s new state using the combined effects of the externally added heat and the heat removed by the cooling system. In short, EnergyPlus *can* simulate the effect of control, but that control is both idealized and expressed differently than it is for physical systems.

The inability to directly simulate physically-realistic control specifications introduces significant friction into the performance-driven control delivery. It means that HVAC engineers who use simulation to select and optimize control strategies cannot directly communicate the intended logic to control implementers. Instead, they specify the intent of the algorithm in detail and then rely on the control engineer to translate that specification to the physical control

platform. Both specification and subsequent translation are manual processes that take time and effort and introduce the possibility of human error. Often the combined cost of time and errors can lead to optimized sequences being discarded in favor of simpler ones with which the implementer is more familiar.

Dynamic HVAC and Control Simulation with Modelica. Simulating physically-realistic control—allowing specifications to be shared between simulation and implementation—requires a shift from quasi-static to dynamic simulation. This shift requires a new simulation approach and substrate, specifically the open Modelica Language Standard. Governed by the Modelica Association, Modelica is a mature and living standard supported by multiple modeling and simulation environments and with a growing number of model libraries. Modelica is not a programming language like C++. It is a modeling—and subsequent simulation—language for engineered systems, and its structure and features cater to it. Most computer languages are *imperative*, programs written in those languages are step-by-step instructions for executing an algorithm. BEM engines like EnergyPlus that are written in imperative languages encode solution methods for the time-evolution of the governing equations of the system they model, in this case a building. Modelica is *declarative*. Modelica models describe the governing equations of the system directly. Modelica compilers and simulation backends analyze these equations and evolve them using domain agnostic solvers.

Modelica makes it possible to simulate feedback control coupled to dynamic HVAC models. The presence of governing equations allows the Modelica simulation environment to compute derivatives, anticipate trigger events such as thermostat setpoint crossings, and dynamically adjust simulation time steps so that state evaluation occurs exactly at these “trigger” events, creating the appearance that control actions are applied immediately.

Dynamic simulation bridges the previously isolated worlds of energy simulation and control implementation. A mechanical engineer can use Modelica-based simulation to test the correctness of and evaluate and optimize the performance of HVAC control sequences, then take those sequences and electronically hand it to a controls implementer who can directly compile and install them into the building’s control system.

Modelica Buildings Library (MBL) and Spawn. For the past ten years, DOE has funded leadership of and significant contributions to the Modelica Buildings Library (MBL), a buildings-domain model library developed and maintained by an international effort that began as an IEA EBC Annex 60 and continues as IBPSA-World Project 1 (Wetter, 2014; Wetter, 2017; Wetter, 2019a). MBL contains models for buildings, HVAC systems, controls and electrical systems, and is compatible with both commercial and open-source Modelica compilers, simulation, environments, and editors.

Modelica, its environments, and equation-based modeling and simulation in general, are unfamiliar to most HVAC engineers and energy modelers. To bridge this gap and aide in the adoption of energy-control simulation, DOE has also funded the development of Spawn, a hybrid engine that uses EnergyPlus to simulate the slow dynamics or weak thermal interactions of weather, lighting, shading, and envelope heat transfer and MBL to simulate HVAC and its control (Wetter, 2020). Spawn also embeds a Modelica compiler and numerical solvers with the goal of presenting a complete package that can easily be incorporated into existing modeling

workflows. Figure 2 shows a Spawn model of one floor of the familiar medium-office prototype building, which uses a multi-zone VAV system. The control logic block implements the ASHRAE Guideline 36 sequences.

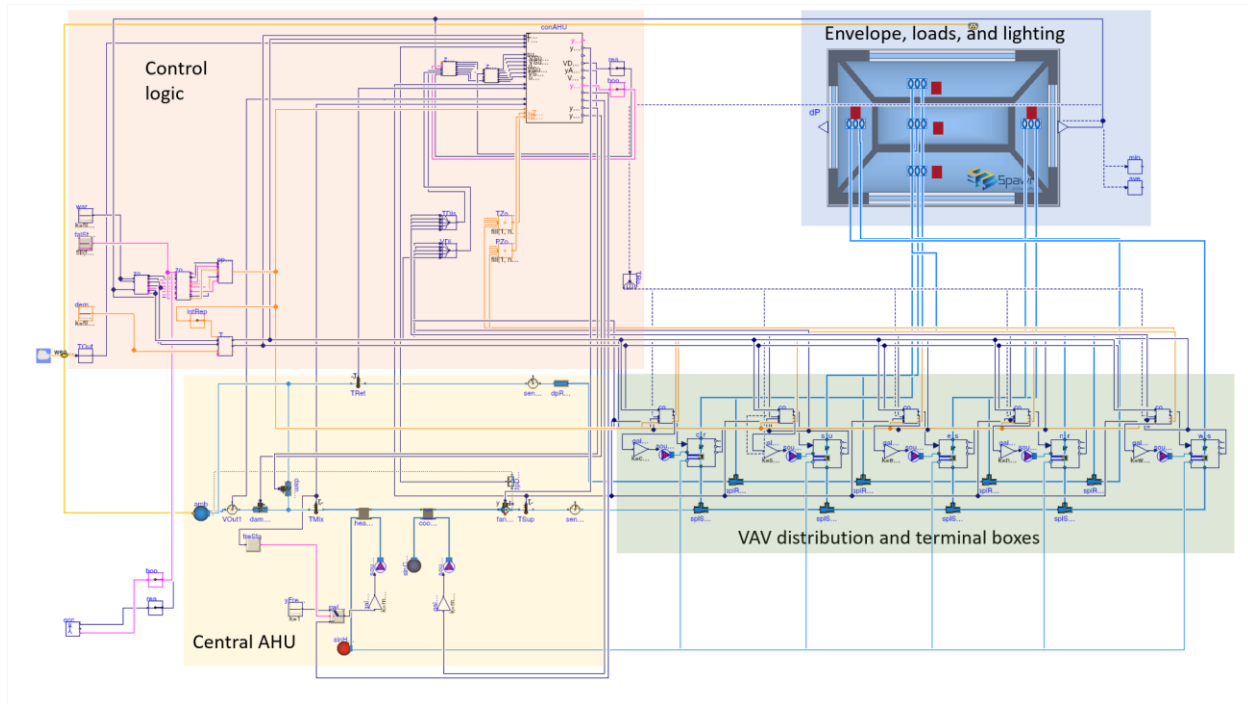


Figure 2. A Spawn model of one floor of the medium office prototype building which uses a five-zone VAV system.

Implementation: Platform-Neutral Control Sequences

In addition to enabling physically-realistic control simulation, the choice of Modelica has another benefit. As a declarative language, Modelica is more abstract—i.e., represents information at higher levels of abstraction—than imperative programming languages like C++ and Python. We will not delve deeply into programming-language theory in this paper, but one important and relevant result in that field is that translating from a more abstract declarative representation to a more concrete imperative one is a deterministic process and can be automated, while the other three translation combinations—concrete to abstract, concrete to concrete, and abstract to abstract—are significantly more difficult. The first uses templates and is something compilers have been doing since the 1950’s. The other three require recognizing patterns and recovering templates which is something that computers cannot do deterministically. This result essentially makes Modelica implementations of control sequences platform-neutral, able to be translated to multiple control frameworks, most of which use lower levels of abstraction and imperative languages.

ASHRAE Standard 231P – Control Description Language (CDL). ASHRAE Standard 231P “Control Description Language (CDL)” codifies platform-neutral digital specification of control logic. CDL describes control logic in a mathematical way. The standard defines primitive

functions—e.g., sum of two inputs, the value of a signal sampled every 30 seconds—rules for composing functions to declare control logic, and a convention for updating function outputs.

Every formal description needs syntax. CDL’s syntax is—you guessed it—a small subset of the Modelica language. CDL control logic can be viewed, authored, edited, inspected and simulated using any Modelica-compliant environment. MBL, Spawn, or other Modelica models can be used to test and evaluate CDL control logic in a whole building energy model.

In addition to performance simulation, CDL’s design also allows translation to established and emerging control platforms CDL’s function primitives are simple and can be implemented in any programming language (Wetter, 2019b) and CDL’s declarative structure supports both textual and graphical editing—each primitive function is represented by an icon and functions are composed using arrows. Figure 3 (at the end of the next section on Semantic Modeling) shows VAV terminal box control logic written in CDL and then translated automatically to Automated Logic Corporation’s platform.

Ecosystem Benefits of Platform-Neutral Control Sequences. The ability to specify control sequences in a digital platform-neutral way creates an interface that separates specification of control algorithms from their implementation in control platforms. This type of layering is common in computer systems and typically confers benefits on the entire ecosystem. Well-defined interfaces enable interoperability—the ability of multiple systems or technologies to exchange information. This affords flexibility in implementations as long as the interface is respected, which in turn creates opportunities for whole ecosystems of competing and complementary technologies. We have already alluded to one instance of it: energy-control simulation tools like the Modelica Buildings Library and Spawn can be used with any control implementation platform. In general, any tool that deals only with control algorithms—e.g., a graphical editor—can be used with any control platform.

Layering and interoperability can extend from tools to products and business models. They facilitate an emerging class of businesses that provide high-performance control algorithms on top of established control platforms. Because control algorithms are software, these products can be offered over the internet.

Another beneficial use of platform-neutral specifications is the creation of shared content, in this case repositories of reference implementations of common control sequences. These serve multiple purposes. During design, they allow different strategies to be explored quickly and automatically. During implementation, they make high-performance sequences more likely to be adopted. A control engineer may be less likely to discard a complex high-performance sequence in favor of a simpler more familiar one if the former comes from a trusted, well tested library. As part of its work in this area, DOE is funding the development of libraries of CDL implementations of common high-performance control sequences, including those from ASHRAE Guideline 36.

Installation and Upgrade: Semantic Modeling

The most widespread barrier to the deployment of high-performance supervisory HVAC control is a barrier to the deployment of supervisory control of any kind—the lack of standard, queryable, and fit-for-purpose descriptions of the building’s HVAC systems, sensing and actuation “points”, and their relationship to the building. Such descriptions are now referred to as

semantic models (Bergmann, 2020). The promise of semantic models is that they can support automated configuration and installation of building automation software, enabling the same kind of “plug-and-play” behavior in that domain as currently exists in other software domains such as mobile applications and e-Commerce. In the absence of a semantic model, building software has to be configured and “point mapped”, i.e., connected to sensing and actuation points, manually.

RDF and SPARQL. Modern semantic modeling frameworks build on two technologies originally developed for the web. Resource Descriptor Framework (RDF) (W3C, 2022) is a metadata schema for explicitly and generically describing objects, their properties, and their relationships. For instance, RDF could be used to describe a person as an object that has a first name which is a character string, a height which is a number of inches between 24 and 96, and eye color which is one of blue, green, or brown. RDF could also describe every person as having a parent where a parent is a different person. RDF could describe that the parent relationship is bidirectional, i.e., that if Joe is Mary’s parent then Mary is Joe’s child. And it could describe constraints on relationships such as every person has two (biological) parents and that a person cannot cyclically be their own biological parent. The same framework can be used to describe buildings and their systems in terms of components and their relationships.

A companion to RDF is SPARQL Protocol and RDF Query Language (SPARQL)¹, a standard language for querying RDF models. A SPARQL query can be used to determine Mary’s height, to list the names of her parents, or to enumerate all of her ancestors that had blue eyes. For a building, a SPARQL query can be used to determine the number and identity of zones served by an air-handling unit, the “point names” of thermostats in those zones, and the “point names” of the outlet air temperature setpoint actuators of the VAV terminals serving those zones. Given a semantic model of a building, it is straightforward to imagine how SPARQL can be used to “discover” a specific building and its HVAC system, to customize a generic control sequence for that specific building—a generic sequence is specified for an arbitrary number of zones, a SPARQL query can be used to create a sequence for the actual number of zones—and to map generic descriptors for the relevant sensors and actuators to their physical counterparts.

Semantic models can “live” in building automation systems (BAS) or in systems that interface with a BAS, e.g., Skyspark². The ability to query a semantic model not only supports automated configuration and mapping of application software, it also enables analysis of the model itself for correctness and “completeness for purpose”. The ability to specify and subsequently check for the presence of specific information allows semantic model installations to be certified and building owners and operators to procure them with confidence.

Brick Schema, ASHRAE Standard 223P, Project Haystack, and XKCD 927. Via different mechanisms, DOE currently supports three semantic modeling efforts. ASHRAE Standard 223P “Semantic Modeling for Building Monitoring and Control Applications” (ASHRAE, 2018), currently under development, is a detailed low-level semantic modeling framework for the building and building-system domains. Brick Schema is positioned as a higher level object-oriented abstraction on top of ASHRAE 223P (Balaji, 2016). Brick has a type hierarchy—e.g.,

¹ Computer scientists are fond of recursive acronyms.

² <https://skyfoundry.com/product>

there is a Brick type “Sensor”, “TemperatureSensor” is a subtype of type “Sensor” and “AirTemperatureSensor” is a subtype of type “TemperatureSensor”—and objects of Brick types represent collections of 223P objects and properties.

A third DOE-sponsored project is modernization of the Project Haystack metadata framework. The utility of standard descriptive building and system models has been recognized for many years. Project Haystack was the first collective effort in this space and has gained acceptance among practitioners. Haystack is not a true semantic modeling framework because it is not RDF-based and does not support SPARQL queries. Haystack also uses sets of tags to identify object types rather than a formal type system. Haystack has made it significantly easier for practitioners to configure, install, and maintain building energy management software but it falls short of supporting fully automated configuration and installation. The modernization project is expanding Haystack in the direction of true semantic modeling, aligning it with Brick, and developing a certification framework for Haystack installations.

To some, this situation may evoke the infamous XKCD “Standards” cartoon³. A more charitable interpretation is that there is great interest in semantic modeling and an increased understanding of its potential value, that people are experimenting with different approaches, that this experimentation will clarify the strengths and weaknesses of different design choices, and that this will lead to alignment if not convergence and increased adoption.

Semantic Models and Platform-Neutral Control Sequences. Semantic models are control platform-agnostic in the same way that ASHRAE Standard 231P/CDL control specifications are. This is important because control specifications need to reference various sensors, actuators, and other information about the underlying system. A semantic model provides a way of doing so in a way that is both platform independent *and* installation independent, as all information required by the algorithm is explicitly represented in the model and can be extracted in a standardized way. In the current implementation of CDL libraries, control logic inputs, outputs, and other parameters are annotated with metadata that allows them to be more easily mapped and installed. In the future, these annotations may be complemented by SPARQL queries that extract the relevant parameters from a semantic model.

Workflow Example. Figure 3 shows an example that ties the various pieces of the ecosystem together. At the top left is CDL control logic for a VAV terminal box—CDL is a computer language, but its elements have graphical annotations that allow them to be displayed and edited graphically. This is a generic sequence that is available from a library. The inset shows the CDL code that describes the VAV damper position output y_{Dam} . Note, in addition to describing this signal in CDL, the description also contains an annotation that states that this signal corresponds to a Brick object of type `Damper_Control`. These annotations allow CDL objects to find their counterparts in Brick descriptions of building systems.

At the top right is a partial Brick semantic model for a building HVAC system that includes three VAV terminal boxes. The code inset here shows the Brick description of the damper position control “point” for VAV box 2-4, `VAV2-4 . DPRPOS`. That description includes the BACnet point name, `DMP_CM_2-4`.

³ <https://xkcd.com/927/>

The bottom part of the figure shows some of the steps of instantiating the CDL logic for a specific building and translating it into a vendor-specific control language, in this case Automated Logic Corporation’s WebCTRL. Among other things, this process requires discovering how many VAV boxes are in the building, instantiating the corresponding number of copies of the control logic, and then “binding” the logic inputs and outputs to the corresponding BACnet control points. This is done using SPARQL to query the Brick model. The full translation requires multiple queries—in fact, queries can be used to determine what system exists in the building and to suggest control sequences that are appropriate to the system—but here we show only two. The first (i.e., step 1) retrieves the names of all VAV boxes in the Brick model; this query returns the list { “VAV2-4”, “VAV2-1”, “VAV2-3” }. The second query (step 2) retrieves the Brick object name and the BACnet point name for the damper position control of VAV box VAV2-4; this returns { “VAV2-4 . DPRPOS”, “DMP_CM_2-4” }. The latter is set as the damper control output signal in the translated sequence (step 3).

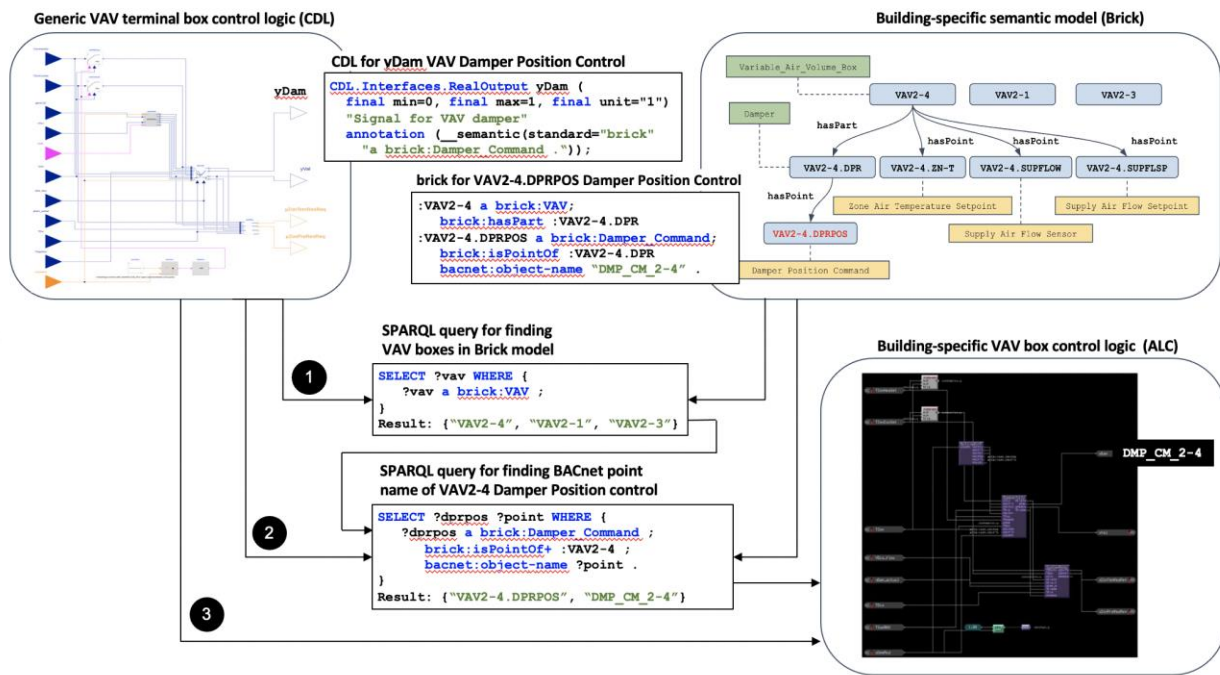


Figure 3. Use of a semantic model (in this case Brick) to automatically instantiate a generic VAV box control sequence for a specific VAV box in a specific building controlled by a specific vendor control platform, in this case ALC’s EIKON. (Wetter, 2022). The top of the figure shows the inputs to the CDL logic on the left and the building specific Brick model on the right. As part of the translation, SPARQL queries are used to query the Brick model and translate generic signal names to physical signal names, i.e., BACnet “points”.

Semantic Models and Energy-Control Simulation. The overlap between semantic models needed to configure building control software and models used to describe buildings for energy-control simulation is obvious and adds to the value proposition of both. In one direction, an HVAC engineer using the MBL or Spawn to evaluate and optimize control sequences could hand over not only the control specification itself, but also a semantic model for the building and its systems. Exporting semantic models from MBL and Spawn is the subject of near future work. In

the reverse direction, a semantic model may provide some of the information required to create a whole-building energy model which could be used for model-predictive control, (MPC) (Drgoña, 2020), fault detection, retrofit planning or other “digital twin” functions.

Verification and Commissioning: Data- and Simulation-Driven Testing

Quality assurance and quality control (QA/QC) are important aspects of building control installation, maintenance, and upgrade. The QA/QC process can be partitioned into three tasks: i) validation, which tests whether specified building systems and their controls meet project requirements including energy codes, ii) verification, which tests whether control implementation matches its specification, and iii) commissioning, which tests operation in the field (Wetter, 2022). Validation and verification are “one time” events whose purpose is to detect errors early in the design and installation process when testing—usually via simulation—can be more comprehensive and correcting errors is cheaper. Initial commissioning is also a onetime event, but commissioning can be repeated periodically and even done continuously as it can detect physical issues with sensors, actuators, and equipment that naturally develop over time. Control sequences do not naturally change over time, but the underlying building may change—its occupancy and use patterns may change—and sequences can be modified and patched in an attempt to track and account for these underlying changes. The commissioning process provides an opportunity to test that sequences are still appropriate.

Simulation-driven verification. Simulation has a key role in validation. It can also play a role in verification and commissioning. The control sequences are simulated using building sensor data streams as inputs. Simulation outputs such as set points and actuator positions, are compared against trend data from the building. This process is useful for control providers who do not have CDL translators for their product lines (Wetter, 2019). Manufacturers can use a similar process to test their control implementation prior to deploying the controller to a building.

Data-driven validation and commissioning. An alternative to using simulation to validate and commission control sequences is to analyze the stream of inputs and subsequent actions taken by the control sequence and check whether conditions and requirements implied by the control sequence are met. This is the approach taken by dAta-driveN building perforMance verificATIOn framEwork (ANIMATE) (Chen, 2021). In ANIMATE, control sequences are coupled with a suite of checks that test whether the intent of the sequence is obeyed. These checks encode the domain knowledge required for manual verification. ANIMATE checks can be applied to both a simulation for sequence validation during design and to the control system of a real building for commissioning. As with control sequences themselves, ANIMATE checks can also query semantic models to automate instantiation and mapping of specific simulated and physical system configurations.

Digital, Performance-Based Control Delivery Processes

Declarative standards that support multiple use cases help automate individual steps in the control delivery process. More importantly, they digitally link these steps together, reducing

the effort and translation and interpretation errors associated with handoffs and allowing assets created in previous project stages to be directly leveraged in later ones.

New Construction. Figure 4 shows the envisioned control delivery process for the new construction case. The project starts with a mechanical engineer—who may be an energy modeler or work with one—using simulation to select and size a building’s HVAC system. Working with MBL or Spawn, the engineer selects different control strategies from a library, evaluates their performance and optionally optimizes their performance. The engineer exports a semantic model of the building’s systems and uses that model to configure ANIMATE which tests the mechanical system and sequences for compliance.

At the end of the design step, the engineer transfers to the control implementer both the semantic model of the system and the CDL specifications of the chosen control sequences. The control implementer can install the semantic model into the building automation system, and compile the CDL specification for execution on their chosen platform. If a translation tool is not available, they can use CDL as a reference implementation to verify their native implementation.

The commissioning agent can again use ANIMATE to test the mechanical system and control sequences in operation. They may also have access to the CDL and even the full MBL or Spawn model which they could use to compare actual behavior to that predicted by simulation.

The semantic model, the CDL specifications and their documentation, and even the full simulation are assets that are available for future use to configure and install additional software, recommission the building periodically or continuously, explore upgrades, or other applications.

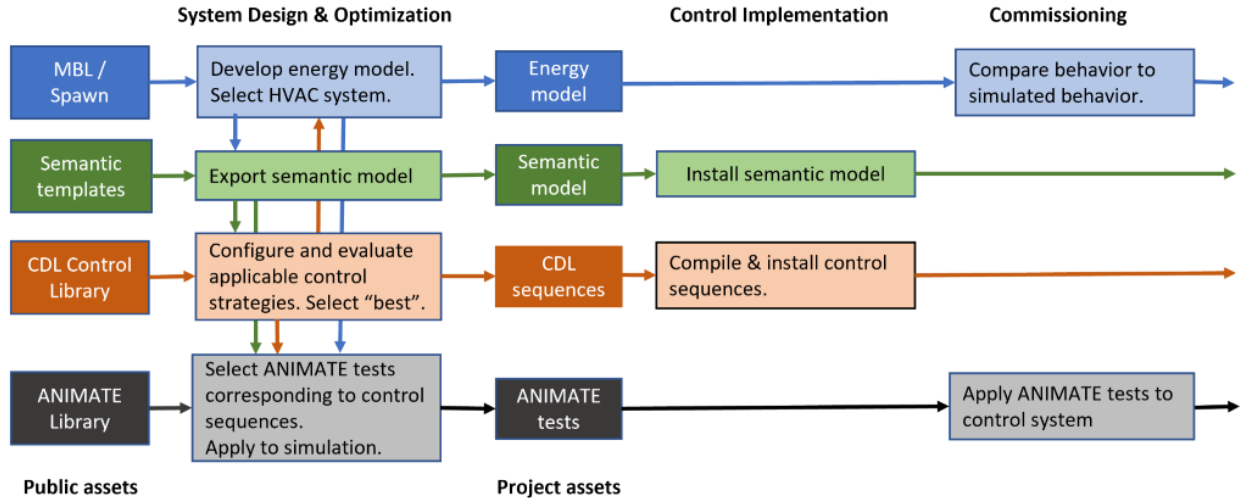


Figure 4. The ecosystem of standards and tools being developed links the steps of control design and optimization, specification, implementation, and commissioning through a collection of digital assets that are created during the design process and subsequently reused in later steps. The key assets are a CDL specification of the chosen control logic and a semantic model that describes the building and its systems. These allow sequences selected and optimized during the design phase to be compiled, configured, and mapped during installation.

Control retrofit. The control retrofit scenario will vary depending on whether the building already has some or all of the digital assets described above. In their absence, the process would ideally begin with the control implementer creating a semantic model of the existing HVAC

system. This model would enable the installation of commissioning software like ANIMATE to test and understand the existing control sequences if these are not well documented.

Next, the semantic model can be used to help create a building energy model. Equipment performance, geometry and envelope characteristics, and internal load schedules must be added. The combination of building model and semantic model can be used to further diagnose existing control strategies and to experiment with different strategies. From this point, the presence of the semantic and energy-control models makes this scenario resemble the new construction one.

Research, Guidelines, and Incentives: Standard Benchmarks and Metrics

In addition to use cases that apply to individual specific building projects, building energy simulation also has use cases that are broadly applicable to all buildings. These include development of codes, guidelines, certificates, programs, and products. EnergyPlus is used in the development of ASHRAE Standard 90.1, ASHRAE Advanced Energy Design and Retrofit Guides, DOE’s Commercial Building Asset Score, and a number of energy-efficiency utility programs. It is also used to support education and research that advance the state of the art in both simulation itself and in the design and operation of artifacts being simulated, i.e., buildings and their systems. Energy-control simulation can serve a similar function for building control.

In addition to simulation, the previous use cases share another common piece of infrastructure—standard benchmarks that represent buildings and systems of interest. For building energy simulation, these are the commercial and residential Reference Building Models which represent existing building stock, and Prototype Building Models which represent new construction and are used to analyze whole-building codes. In the building energy world, many applications define threshold performance levels relative to code and so the latter are used frequently. Supporting use cases like certification, program development, and research requires similar energy-control standard benchmarks.

BOPTest. BOPTest (Building Optimization Performance Testing Framework) is designed to support testing and benchmarking of control algorithms (Blum, 2021). It is a set of Modelica models, an environment for simulating them, and functions for computing control performance metrics—annual energy use, peak energy use, thermal comfort, and others—over these simulations. BOPTest uses the Alfalfa framework to embed a Modelica model within a server process and present its control interface via an API (application programming interface) that mirrors the one presented by a BAS. Control algorithms can interact with a BOPTest simulation as if it is a BAS of a physical building.

ML-Driven Control. Thus far we have focused on control sequences and touched on physics based MPC. However, it is currently not possible to write a paper about building control without mentioning Machine Learning (ML). ML has captured the attention of many control researchers and practitioners who see ML-driven models as promising replacements for or complements to physics-based models in optimization-driven control frameworks. Are the standards and capabilities we have described relevant to optimization-driven control, either physics- or ML-based, given that it does not use hand-crafted sequences? In short, yes.

To begin with, optimization-driven control typically sits “above” rule-based conventional control. Optimization is typically used to determine temperature setpoints of different zones and

components, while rules are used to implement those setpoints. Ensuring robustness and building in safeguards is currently too challenging in optimization-driven control. Semantic modeling, portable control sequences, and energy-control simulation are still needed to deliver the underlying conventional algorithm that both trains and backstops it.

Second, semantic modeling supports configuration and installation of any algorithm to a specific building. Separately, it may also be useful to ML-based control frameworks that use hybrid or gray-box modeling approaches.

Finally, energy-control simulation is relevant to ML-based control in several additional ways. Simulation can be used to train ML models. You may be wondering why an ML model would be necessary if a physics model is available. The answer is that the physics model need not match the actual building exactly. A generic physics model may be used to fast-forward the training of the ML model and get it close enough to the target building before the latter starts learning from the real thing. Simulation can also be used to benchmark ML-based control. As with any new technology, especially a non-deterministic one there can be significant uncertainty about performance, both vis-à-vis deterministic conventional approaches and among different ML-based offerings. Simulation can be used to directly compare ML-based control algorithms against one another and conventional algorithms under a range of configurations and encourage the adoption of high-performance ML-based control. Finally, standard simulation benchmarks can help advance the state-of-the-art in optimization-driven control. ML models are in vogue, but they are not superior to physics models along every dimension. Researchers are experimenting with different hybrids that attempt to use the strengths of one approach to offset the weaknesses of the other. A standard benchmarking framework like BOPTest can help obtain a detailed understanding of the relative merits of different combinations and guide future research.

Progress Check

We described a vision for a digital and performance-driven supervisory control ecosystem that spans all stages of project delivery and supports benchmarking that helps advance the state-of-the-art and encourage adoption. How far along are we towards achieving this vision?

Energy-control simulation has received the longest and most sustained DOE investment and is the area in which the most progress has been made. Backed by an international effort, MBL has good model coverage and is supported by multiple compilers and tools. A co-simulation enabled unit that contains EnergyPlus' weather, lighting, internal loads, and envelope heat transfer modules was released as part of MBL 8.0 in April 2021, allowing users to run Spawn models. A packaged version of Spawn that includes a Modelica compiler and simulation backend will hopefully be released later this year, and development of OpenStudio SDK support for Spawn will begin shortly thereafter. Several control providers and equipment manufacturers are actively using MBL and Spawn to develop new product lines, as are several consultants for project work. ASHRAE Standard 231P is expected to be released for public review in 2022. Software is being developed around CDL's Modelica syntax, including reference libraries of control sequence implementations and translators from CDL to commercial control platforms.

BOPTest is already in use for research and training and DOE is evaluating its suitability for benchmarking smart thermostat algorithms.

ASHRAE Standard 223P will also hopefully be released for public review within the next year. The co-evolution of 223P, Brick, and Haystack and their ultimate relationship to one

another is still uncertain. However, development on all three is proceeding. As part of this effort, tools are being developed for translating from one schema to another. This helps identify areas of fundamental misalignment that need attention. Work has already started on exporting semantic models from Modelica models. As the semantic modeling framework crystalizes, this work will accelerate and SPARQL queries will be added to CDL and ANIMATE libraries.

Acknowledgments

This work was supported by U.S. Department of Energy, Office of Energy Efficiency and Renewable Energy, Building Technologies Office under contracts DE-AC02-05CH11231, DE-AC05-76RL01830, and DE-AC36-08GO28308. The authors thank the many people who have contributed both directly and indirectly to the standards and tools described in this paper.

References

- ASHRAE. 2018. “ASHRAE's BACnet Committee, Project Haystack and Brick Schema Collaborating to Provide Unified Data Semantic Modeling Solution”. Retrieved at <https://www.ashrae.org/about/news/2018/ashrae-s-bacnet-committee-project-haystack-and-brick-schema-collaborating-to-provide-unified-data-semantic-modeling-solution>
- ASHRAE. 2021. “Guideline 36-2021, high-performance sequences of operation for HVAC systems”. ASHRAE, Atlanta, GA. June.
- Balaji, B., A. Bhattacharya, G. Fierro, J. Gao, J. Gluck, D. Hong, A. Johansen, J. Koh, J. Ploennigs, Y. Agarwal and M. Berges. 2016. “Brick: Towards a unified metadata schema for buildings”. Proc. of 3rd ACM International Conference on Systems for Energy-Efficient Built Environments (pp. 41-50). Nov. <https://doi.org/10.1145/2993422.2993577>
- Barwig, F.E., J.M. House, C.J. Klaassen, M.M. Ardehali and T.F. Smith. 2002. “The national building controls information program”. In: ACEEE Summer Study on Energy Efficiency in Buildings, Pacific Grove, CA. http://aceee.org/files/proceedings/2002/data/papers/SS02_Panel3_Paper01.pdf
- Bergmann H., C. Mosiman, A. Saha, S. Haile, W. Livingood, S. Bushby, G. Fierro, J. Bender, M. Poplawski, J. Granderson, and M. Pritoni. 2020. “Semantic Interoperability to Enable Smart, Grid-Interactive Efficient Buildings.” ACEEE Summer Study on Energy Efficiency in Buildings 2020. <https://doi.org/10.20357/B7S304>
- Blum D., J. Arroyo, S. Huang, J. Drgona, F. Jorissen, H. Walnam, Y. Chen, K. Benne, D. Vrabie, M. Wetter, and L. Helsen. 2021. “Building optimization testing framework (BOPTTEST) for simulation-based benchmarking of control strategies in buildings”. In Journal of Building Performance Simulation. Volume 14. Issue 5. pp 586-610. Oct 2021. <https://www.tandfonline.com/doi/full/10.1080/19401493.2021.1986574>
- Blochwitz, T., M. Otter, M. Arnold, C. Bausch, C. Clauß, H. Elmqvist, A. Junghanns, J. Mauss, M. Monteiro, T. Neidhold, D. Neumerkel, H. Olsson, J.-V. Peetz, S. Wolf. (2011). “The Functional Mockup Interface for Tool independent Exchange of Simulation Models.” Proc. of 8th International Modelica Conference. 105-114. <https://doi.org/10.3384/ecp11063105>.
- Chen Y., J. Lerond, X. Lei, M.I. Rosenberg., and D. Vrabie. 2021. “A Knowledge-based Framework for Building Energy Model Performance Verification.” Proc. of BS2021: 17th International Conference of the International Building Performance Simulation Association

- Crawley D.B., L. K. Lawrie, F. C. Winkelmann, W. F. Buhl, Y. J. Huang, C. O. Pedersen, R. K. Strand, R. J. Liesen, D. E. Fisher, M. J. Witte, J. Glazer. 2001. EnergyPlus: creating a new-generation building energy simulation program. *Energy and Buildings*, Volume 33, Issue 4. Pages 319-331. [https://doi.org/10.1016/S0378-7788\(00\)00114-6](https://doi.org/10.1016/S0378-7788(00)00114-6)
- Drgoňa J., J. Arroyo, I. C. Figueroa, D. Blum, K. Arendt, D. Kim, E. P. Ollé, J. Oravec, M. Wetter, D. L. Vrabie, L. Helsen. 2020. “All you need to know about model predictive control for buildings.” *Annual Reviews in Control*. <https://doi.org/10.1016/j.arcontrol.2020.09.001>
- Fierro, G., A.K. Prakash, C. Mosiman, M. Pritoni, P. Raftery, M. Wetter, D. Culler. “Shepherding Metadata Through the Building Lifecycle.” *Proc. of 7th International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*. Nov, 2020. Pages 70-79. <https://doi.org/10.1145/3408308.3427627>
- Project Haystack. 2022. “Project Haystack”. Accessed March. <https://project-haystack.org/>
- Schluck T., P. Kräuchi, and M. Sulzer. 2015. “Non-linear thermal networks - how can a meshed network improve energy efficiency?” *Proc. of International Conference CISBAT*, 779–785. Lausanne, Switzerland, Sept. <https://doi.org/10.5075/epfl-cisbat2015-779-784>.
- Sommer T., M. Sulzer, M. Wetter, A. Sotnikov, S. Mennel and C. Stettler. 2020. “The reservoir network: A new network topology for district heating and cooling.” *Energy*, 199. <https://doi.org/10.1016/j.energy.2020.117418>
- W3C (World Wide Web Consortium). 2022. “Resource Descriptor Framework (RDF).” Accessed March. <https://www.w3.org/RDF/>
- Wetter M. and C. van Treeck. 2017. IEA EBC Annex 60: New Generation Computing Tools for Building and Community Energy Systems. Final report. ISBN 978-0-692-89748-5. <http://www.iea-annex60.org/final-report.html>
- Wetter M., W. Zuo, T. S. Noudui and X. Pang. 2014. “Modelica Buildings library.” *Journal of Building Performance Simulation*, 7(4):253-270. <http://dx.doi.org/10.1080/19401493.2013.765506>
- Wetter M., P. Ehrlich, A. Gautier, M. Grahovac, P. Haves, J. Hu, A. Prakash, D. Robin and K. Zhang. 2022. [OpenBuildingControl: Digitizing the control delivery from building energy modeling to specification, implementation and formal verification](#). *Energy*, Volume 238, Part A.
- Wetter M., M. Grahovac and J. Hu. 2018. [Control Description Language](#). *1st American Modelica Conference*, Cambridge, MA, USA, August 2018. <https://doi.org/10.3384/ECP1815417>
- Wetter M., C. van Treeck, L. Helsen, A. Maccarini, D. Saelens, D. Robinson and G. Schweiger. 2019a. “IBPSA Project 1: BIM/GIS and Modelica framework for building and community energy system design and operation - ongoing developments, lessons learned and challenges.” *Sustainable Built Environment Conference*, Graz, Austria, September. <https://www.doi.org/10.1088/1755-1315/323/1/012114>
- Wetter M., A. Gautier, M. Grahovac and J. Hu. 2019b. [Verification of Control Sequences within OpenBuildingControl](#). *Proc. of 16th IBPSA Conference*, p. 885-892, Rome, Italy, September.
- Wetter M., and J. Hu. 2019c. [Quayside Energy System Analysis](#). LBNL-2001197.
- Wetter M., K. Benne, A. Gautier, T. S. Noudui, A. Ramle, A. Roth, H. Tummeseit, S. Mentzer and C. Winther. 2020. [Lifting the Garage Door on Spawn, an Open-Source BEM-Controls Engine](#). *Proc. of Building Performance Modeling Conference and SimBuild*, p. 518--525, Chicago, IL, USA, September.