IBPSA-USA
SimBuild
2026

12th National Conference of IBPSA-USA
Minneapolis, Minnesota
May 20 – 22, 2026

Digitalizing Building Control Deployment for Retrofits: A Case Study on Demand-Flexible Control Sequences

Marco Pritoni[1]*, Anand Prakash[1]*, Lazlo Paul[1], Weiping Huang[1], Roman Kukharchuk[2]
Stephen Dawson-Haggerty[2], Matthias Sulzer[1] and Michael Wetter[1]
[1]Lawrence Berkeley National Lab, Berkeley, CA
[2]Normal Software, Pittsburgh, PA

Abstract

The traditional approach to designing, specifying, deploying, and testing building HVAC controls in large commercial buildings is labor-intensive and error-prone, leading to high cost and poor operational performance. Recent efforts have enabled a comprehensive, digital workflow for model-based control design, deployment, and verification using open-source technology and software standards. This paper presents a case study that applies this workflow to retrofit a demand-flexible "zone ratcheting" control sequence onto a commercial control platform connected to a virtual building. The study demonstrates that automated semantic binding successfully mapped the control logic to the correct building points, eliminating the need for manual programming. Once deployed, the sequence functioned as intended: it correctly ratcheted the cooling setpoint and transmitted the corresponding signals to the virtual building. Ultimately, this work aims to catalyze transformation in the building industry by enhancing grid flexibility, efficiency, and comfort through improved control sequences using a scalable, digitalized control deployment that allows end-to-end quality control, leading to widespread reliable and cost-effective adoption of advanced control.

Introduction

The building sector plays a substantial role in global energy usage, with building operations accounting for approximately 30% of global energy consumption (International Energy Agency (IEA) 2022). In addition, buildings serve as the primary location for human activity, with an estimated 90% of people's time spent indoors (Klepeis et al. 2001). As occupant comfort expectations grow, regulations become more stringent, and weather patterns become more extreme, buildings must now deliver increasingly higher levels of efficiency, Demand Flexibility (DF), and occupant satisfaction (Schiller et al. 2022; Satchwell et al. 2021; Becerik-Gerber et al. 2022). To meet these demands, building systems are becoming more integrated with other energy systems (Wetter and Sulzer 2024). This growing complexity has created a pressing need for sophisticated control solutions that can deliver high-performance operation(Wetter et al. 2022). However, research has consistently shown that large commercial building operations are often hindered by incomplete or incorrect translation of mechanical planning into automation principles, programming errors and control-related issues, even with relatively simple control sequences (Barwig et al. 2002; Crowe et al. 2020; Torabi et al. 2022). The root of the problem lies in the current control design and implementation process, which is error-prone and often relies on HVAC designers who lack the necessary resources or tools to develop effective control logic. Instead of providing a formal, executable specification of the control logic, HVAC designers typically create a control intent document written in ambiguous English. The control provider then interprets this document and develops software code, often reusing code from previous projects (Paliaga et al. 2020; Wetter et al. 2022). Moreover, control logic development is typically handled as a siloed discipline that is appended at the end of the mechanical design process. This prevents effective and holistic system design (Sulzer et al. 2023) The process also defers testing of the control logic to the point where the building starts operating, thereby making any control troubleshooting or improvements very costly. Moreover, the process also lacks formal verification of as-implemented sequences relative to the design specification, with commissioning often restricted to limited checking (Wetter et al. 2019).

Gap and objective of the paper

To overcome these challenges, a novel digital workflow for design, verification, and deployment, with formal end-to-end quality control, has emerged over the past decade. This workflow is built on open-source technologies and open standards for modeling and simulation (Wetter et al. 2025). The control industry has demonstrated growing interest in embracing control digitalization (Wetter et al. 2025), as they see risk and cost-

---

*These authors contributed equally to this work reduction benefits, as well as potential new markets (Pritoni et al. 2024). However, despite this momentum, there is still a pressing need for comprehensive case studies, reference implementations, and guidelines to facilitate broad adoption. Futher, while the existing literature predominantly focuses on new control system designs mostly tailored to new construction, the application of digitalized workflows to retrofits, especially for remains significantly underexplored (Granderson et al. 2025). Addressing this gap is critical, especially large commercial buildings, which account for a substantial portion of peak electricity demand and stress on distribution circuits. In this paper we introduce a new digitalized workflow for retrofits of existing buildings. We validate this approach by deploying a demand-flexible sequence on a commercial control platform.[*] For the control system to have the same interface as in a real building, the platform interfaces via BACnet (ASHRAE 2018) with a building emulator that generates physically realistic closed-loop responses. The BACnet interface enables a seamless transition from the emulator to a real building by simply remapping device addresses and I/O points.

Technical background

Control description language (CDL)

A fundamental building block of this workflow is the Control Description Language (CDL), a vendor-neutral language to represent control logic that allows simulation of the control logic coupled to building energy models. CDL is built upon a subset of Modelica, carefully selected to facilitate translation to various legacy control languages (Wetter, Grahovac, and Hu 2018). Control sequences for buildings have been implemented in CDL to form standardized sequence libraries in previous efforts. These standardized CDL sequence libraries are available through the Modelica Buildings Library (MBL) (Wetter et al. 2014)[*]. The standardized CDL sequence libraries, accompanied by Modelica templates for HVAC systems, forms the foundation for the digitalization of the control delivery (Wetter et al. 2022; Gautier et al. 2023; Wetter et al. 2025). CDL is currently undergoing standardization as ASHRAE Standard 231. The standard also defines the Control eXchange Format (CXF). CXF was developed to simplify the import and export of control logic across different tools and BASs. CXF files are serialized in JSON-LD (.jsonld) and define the same control logic as CDL, ensuring consistency and interoperability (Wetter et al. 2021). Standard 231 is set to become a key component of a comprehensive suite of complimentary ASHRAE standards and guidelines, including Standards 223 (semantic modeling) and 135 (BACnet communication protocol), and Guideline 36 (high-performance sequences). This suite collectively aim to facilitate the digitalization of controls and analytics (Pritoni et al. 2024). Several tools have been developed to support this workflow, including *ctrl-flow* (Wetter et al. 2025)[*], a tool to export standardized control specification documents from tested control models, *modelica-json*[*] a tool to export CDL sequences and related semantic models in json format (Wetter et al. 2021), and *funnel*[*], a tool to compare trajectories of a control sequence against its reference trajectories as part of verifying an implemented sequence against its digital specification.

Semantic models

Semantic models are building-specific representations of physical, logical, and virtual assets in buildings and their relationships, based on schemas and ontologies (i.e. dictionaries that contain definitions of these terms and relationships), such as the Brick schema and ASHRAE Standard 223 (Pritoni et al. 2021). Semantic models provide machine-readable and standardized descriptions of building concepts, enabling semi-automated configuration of applications to an installed system through a common language. Semantic models enable different systems and applications to understand and interpret data and information exchanged. Previous research has demonstrated their use to automate the setup of various building applications, including anomaly detection (Chiosa et al. 2024), digital twins for fault detection (Bjørnskov, Jradi, and Wetter 2025), model-predictive controls (Paul et al. 2025), and rule-base controls for DF (de Andrade Pereira et al. 2025).

To enable automation and interoperability, applications must be written using semantic concepts instead of hard-coding point names or IDs[*]. For instance, consider a VAV[*]/Zone control sequence that

---

[*] *Platform* refers to software that supports controls (e.g. building automation system or BAS), analytics (e.g. energy management and information system (EMIS), or other functions related to buildings; *application* denotes a single program running on a platform; and *code* refers to software written as part of an application.

[*] https://github.com/lbl-srg/modelica-buildings

[*] https://github.com/lbl-srg/ctrl-flow

[*] https://github.com/lbl-srg/modelica-json

[*] https://github.com/lbl-srg/funnel

[*] *point identifiers or point IDs* are unique identifiers assigned to each instance of a point. Typically these are not directly used in applications. *Point names* are descriptive names given to points for ease of identification

[*] *VAV*: variable air volume terminal unit

requires a zone air temperature sensor as a control input (Figure 1). Instead of hard-coding the setpoint name for each zone in the building, such as "EastZone", "WestZone", "NorthZone", and "SouthZone", the application can reference a point of class "Zone_Air_Cooling_Temperature_Setpoint" and its corresponding Zone of class "Zone". These two entities are linked by a relationship "hasPoint". When the application is instantiated, a query is run on the building's semantic model, identifying all instance of "Zones" that have a "Zone_Air_Cooling_Temperature_Setpoint". The query retrieves the actual unique point identifiers, network addresses, or API endpoints, for each VAV box, enabling the application to connect to the correct sensors as intended.

Ontologies such as Brick and ASHRAE Standard 223 represent semantic models as graphs. They utilize standards developed by the World Wide Web Consortium, such as the Resource Description Framework (RDF)[*]. These models are often serialized in the Terse RDF Triple Language (turtle) format[*], denoted as "text/turtle" (.ttl) files. The ecosystem surrounding these ontologies includes several key technologies that can be leveraged by the building industry. SHACL (Shapes Constraint Language))[*] is used to validate the presence of required semantic information in a building's semantic model, ensuring the successful deployment of control sequences. Additionally, SPARQL[*] is a query language designed to retrieve and modify information within semantic models, further enhancing the ability to automatically connect applications and to their inputs and outputs.

As building controls and analytics vendors increasingly adopt digital technologies, semantic models are becoming more readily available, even for existing buildings. In cases where these models do not exist, tools such as the Building Metadata OnTology Interoperability Framework *(BuildingMOTIF)* [*] can be used to create and validate them (Fierro et al. 2022).

Control deployment workflow for existing buildings

We contrast the traditional, predominantly manual control retrofit process with two novel digitalized workflows designed to improve scalability and reliability (Figure 2). While the first digitalized process addresses the replacement of an entire control system, the second introduces a workflow for retrofitting specific control sequences onto an existing platform.

1. Traditional control retrofit process

This represents a generalization of current industry practice, covering scenarios where either the control system or the control sequences are updated. Although there are some nuances between the two cases, such as the need to understand existing sequences in the latter, the steps are largely identical and predominantly manual in nature.

1.a. Manually collect and interpret building data. The first step in the workflow involves collecting and interpreting data on the existing building, its systems, and controls, including type of installed equipment, schematic diagrams of the installed systems, control sequences, and references to data and control points. This process is often hindered by incomplete, inaccurate, outdated, or non-digital documentation, making it a manual, error-prone, and time-intensive task for practitioners (Paliaga et al. 2020). A significant part of this effort involves deciphering the meaning of point names in existing control systems, drawings, and tables (Bergmann et al. 2020). Familiarity with the existing control sequence is also important, and it requires analysis of natural language descriptions or code implementations.

1.b. Specify updated control sequence in English. Based on the gathered information, the control designer selects a suitable sequence, typically relying on empirical evidence, personal experience, and knowledge of similar projects. Often, there is no systematic process to identify the optimal sequence for the specific project or to verify that the chosen sequence meets the requirements of the building owner (Wetter et al. 2022). The selected sequence is documented in a natural language document, which includes supporting charts, required control points, HVAC schematic diagrams, and configuration parameters (Paliaga et al. 2020).

1.c. Interpret and program control sequence for on-site hardware and software. The selected control implementer, in a typical design-bid-build process, receives the English document and develops the control logic code for the specific Building Automation System (BAS). Due to financial and time constraints, as well as a lack of clear specifications and processes for verification, common practice is to reuse code from previous projects with similar HVAC systems (Pritoni et al. 2024). The code is then deployed on-site, involving instantiation of the new control logic for each subsystem (e.g., each AHU controller).

---

[*] https://www.w3.org/RDF/

[*] https://www.w3.org/TR/turtle/

[*] https://www.w3.org/TR/shacl/

[*] https://www.w3.org/TR/sparql11-query/

[*] https://nrel.github.io/BuildingMOTIF/README.html

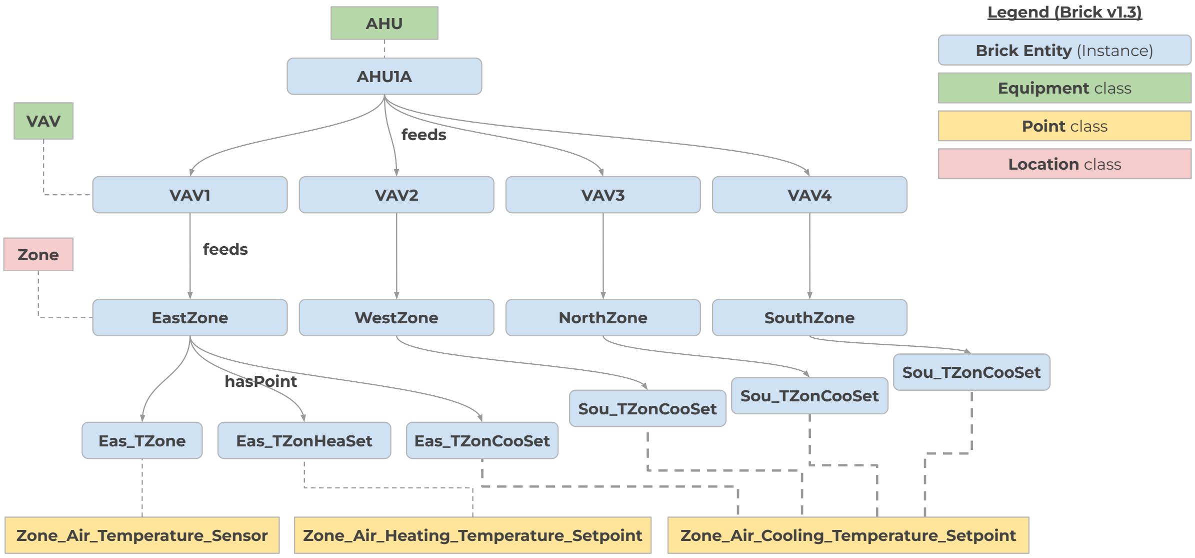


Figure 1: Example of semantic model (graph) based on the Brick ontology (v1.3) used in the case study. This model allows applications to query for abstract classes (e.g., "Zone_Air_Cooling_Setpoint") to dynamically resolve unique point identifiers for all the zones, thereby eliminating the need to hard-code specific point names.

1.d. Verify control sequence on site. Prior to full operation when sequences are updated or replaced, a commissioning agent typically conducts a series of tests. These tests include functional tests to verify that mechanical components, such as dampers, pumps, and fans, can be controlled and respond as expected (Kao 1992). In addition, tests are performed to ensure that the software is free of bugs, control response is stable and parameters, such as PID coefficients, are properly tuned. Due to the time-consuming nature and costs associated with conducting these field tests, their scope is often limited. As a result, the commissioning agent typically only tests the new control sequence under a restricted set of operational scenarios. Verifying control logic before field deployment is difficult because they require dynamic closed-loop simulators to stress test the sequences, a process that control providers are not set up to do efficiently.

2. Digitalized deployment process: replacement of control systems

The digitalized process described in previous work (Wetter et al. 2022) is particularly well-suited for new construction or major retrofits in which the control system is completely replaced. This allows the new control sequences to be defined during the design stage without constraints posed by the existing control software. The most important steps are summarized below and depicted in yellow in Figure 2.

2.a. Manually collect and interpret building data. Replacing the control system, like traditional workflows, requires gathering information about the building's existing physical assets, often from scratch, as the legacy system's removal typically erases previous connections to field sensors and actuators and their digital configurations.

2.b. Configure pre-verified control sequence.

The new workflow utilizes CDL sequences that have been pre-validated through simulation-based testing using Modelica models across a range of operational scenarios and building systems/configurations (Wetter et al. 2022). Currently, Modelica templates exist to semi-automatically create the CDL control sequences for different building systems and configurations (Gautier et al. 2023). Additionally, we have prototyped a web-based tool, *ctrl-flow*, which enables users to configure CDL sequences using intuitive drop-down menus. This tool is currently being extended to implement new features and sequences.

2.b. Export digital specification.

New features in *ctrl-flow* will allow to export English language control sequences, digital representation of these sequences in CDL, schematic diagrams, point types lists and semantic models. Even without this

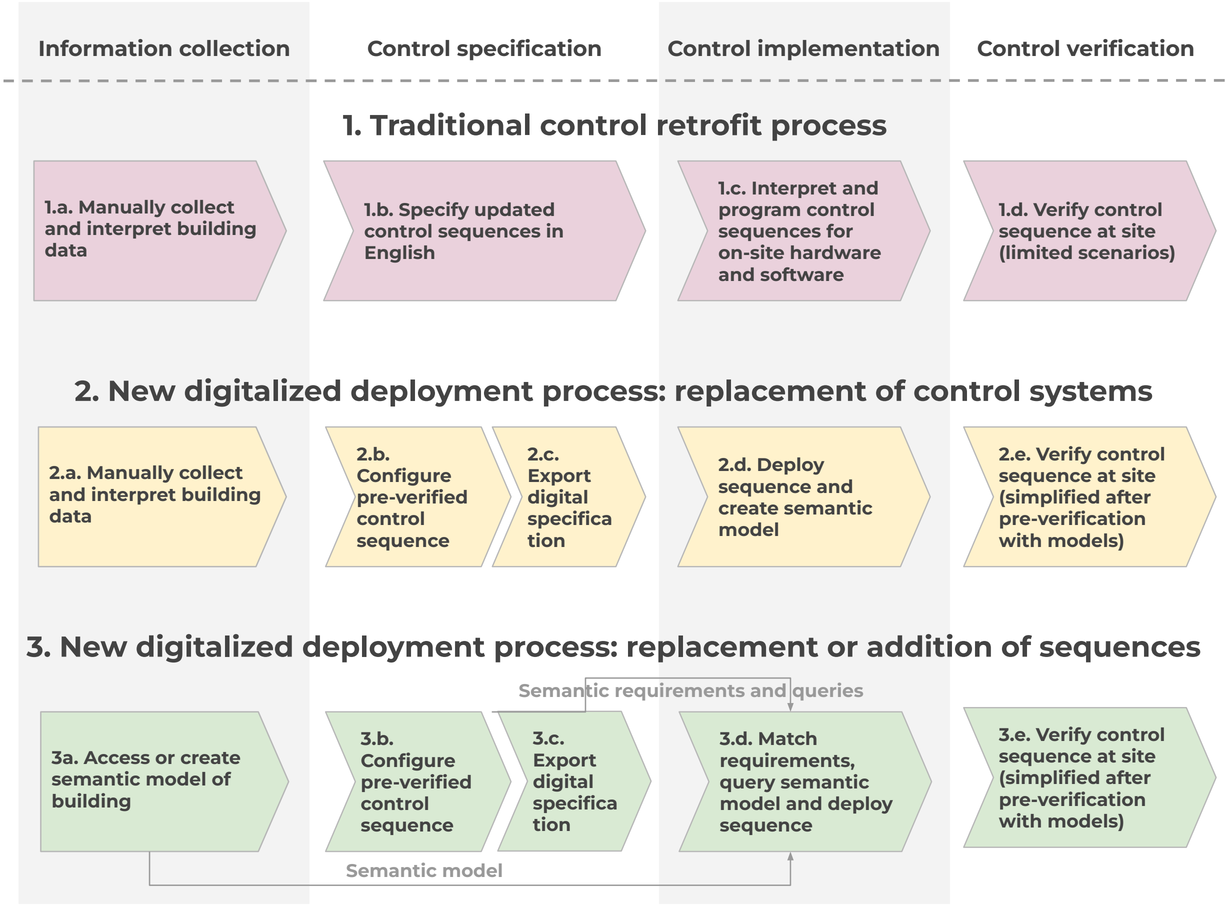


Figure 2: Comparison of traditional and two new digitalized control deployment processes. Process 2 (middle panel) illustrates the complete replacement of a control system; in this "clean slate" scenario, new point identifiers and semantic models are generated as part of the deployment (Step 2.c). In contrast, Process 3 (bottom panel) addresses the addition of sequences to an existing system. This requires a unique mapping process (Step 3.d) where the semantic model is queried to bind the new digital sequence to pre-existing physical points and network addresses.

tool, CDL supports the inclusion of semantic information within annotations in each sequence and provides a syntax for embedding and exporting semantic information [*]. Using the *modelica-json* software, the CDL and semantic information can be exported in CFX. *Modelica-json* can also export the semantic information embedded within the CDL sequence as "text/turtle" (.ttl) file.

2.c. Deploy sequence and create semantic model.

Next, the CDL sequence must be imported into the new control system. However, if the system does not natively support CDL/CXF and the semantic schema used, a translation is required to ensure compatibility with the native control software. Previous research has shown that such translations are feasible. For example, CXF representations of CDL sequences have been translated into Eikon (used in Automated Logic's WebCTRL framework (Wetter et al. 2022)), JavaScript (used by Normal (Normal Software 2025)), and IEC 61131 (used in programmable logic controllers (PLC) (Walther et al. 2025)).

After importing the sequences, they must be instantiated for each piece of equipment they control, such as all VAV boxes in a building as shown in Figure 1. During this phase, unique point IDs are generated for each input/output (I/O) of the sequences. Additionally, if the control platform supports it, semantic information derived from the CDL annotations can be assigned to the points just created or can be used to generate new points.

---

[*] https://obc.lbl.gov/specification/cdl.html#semantic-information

On certain platforms, this semantic information can be organized into a comprehensive semantic model that can be queried and updated using semantic web tools. These models can be used to configure other applications, including graphics, data visualization, and fault detection and diagnostics tools.

2.d. Verify control sequence at site. Similarly to the traditional workflow, testing is a crucial step before full operation. However, since the sequence's logic has already been validated through simulation, we anticipate that this testing phase will be shorter and pose less risk for the control implementer.

3. Digitalized deployment process: replacement or addition of sequences

When retrofitting control systems by replacing or adding control sequences, existing point names and unique identifiers (such as network addresses) often already exist, particularly for points linked to physical sensors and actuators previously installed. Therefore, during sequence instantiation, it is necessary to map the I/O points of the sequence to these existing points. This mapping process is a critical task that differentiate significantly the replacement or addition of sequence, depicted as the bottom workflow in Figure 2, from the full control system replacement, represented by the middle workflow in Figure 2.

3.a. Access or create semantic model of building.

If the control platform supports it, a semantic model can be used to store and organize information about point IDs, point names, and equipment instances in a building. Accessing and querying such model can support automation in the deployment of controls as described in step 3.d. If the model does not exist, it can be created with tools such as *BuildingMOTIF* (Fierro et al. 2022).

3.b. Configure pre-verified control sequence. As described in step 2.b, we can leverage pre-verified sequences that can be configured based on the type of equipment.

3.c. Export digital specification. The semantic information in the CDL annotations can also be leveraged to generate SHACL specifications and SPARQL queries.

3.d. Match requirements, query semantic model and deploy sequence. In this step, we match the semantic model obtained in step 3.a with the semantic information from the CDL sequence (step 3.c). Using SHACL shapes, we validate the presence of required semantic information in the semantic model and determine whether a given sequence can be successfully instantiated on a specific piece of equipment. Next, for each instance of the sequence, we use the SPARQL query language to retrieve the unique point IDs from the semantic model and bind them to the instantiated sequence.

3.e. Verify control sequence at site. Similarly to step 2.e, we expect a reduced time required for field testing.

Case study

To demonstrate the proposed digitalized deployment process for adding sequences to an existing control system (workflow at the bottom of Figure 2), we present a case study in which a supervisory DF control sequence is deployed on a commercial control platform: the Normal Framework [*]. Note that, due to space constraints, we do not provide an example of the new workflow #2, which involves replacing the entire control system, as depicted in the middle panel of Figure 2. However, this process partially overlaps with the one described for designing a new control system in previous publications. (Wetter et al. 2022).

The DF sequence tested is designed to reduce HVAC load between 4-9 PM. The control platform is connected through BACnet to a virtual HVAC and building emulator, utilizing the Building Optimization Performance Tests (BOPTEST) framework (Blum et al. 2021). Therefore, the control system interfaces with the emulator in the same way as it would interface with a real building.

BOPTEST offers a collection of pre-configured virtual building emulators representing various building types and HVAC systems, allowing for a closed-loop control response to the new sequence. For this case study, we utilize a modified version of the "Multizone Office Simple Air" BOPTEST testcase emulator. The modified testcase emulates one floor of an office building in Chicago, IL, USA with 4 perimeter zones, 1 core zone, and a single-duct multi-zone VAV system with terminal reheat and with one air handler. More details about this virtual building can be found in the BOPTEST documentation (BOPTEST 2025).

The Normal Framework enables the development of control and analytics applications that interact with real and emulated buildings. Furthermore, the platform provides a feature to classify and group points based on their point-type (similar to a Brick class, as shown in Figure 1) and associated equipment, assigning vendor-specific metadata tags to each point. These custom metadata tags can be leveraged by applications developed on the platform to query and write back to the corresponding points,

---

[*] Normal Framework: https://www.normal.dev/solutions.html

eliminating the need to reference each point individually using their point names or IDs. They can also be linked to semantic models using a standardized schema, such as Brick or ASHRAE Standard 223, enabling the application of standardized workflows for control sequence deployment. In the next paragraphs, we describe how the sequence was deployed, following the steps in the workflow at the bottom of Figure 2.

a. Create semantic model of building. We used the existing Brick semantic model embedded in the BOPTEST test case that represents the equipment, zones, and data points of the emulated building. We enhanced this model by adding references to the Normal Framework metadata tags as external references in the Brick model. This approach was preferred over directly utilizing the vendor-specific semantic model, as Brick is a widely-used schema across multiple industry vendors and has extensive documentation in the literature.

b. Configure pre-verified control sequence. For this case study, we developed the DF sequence in CDL shown in Figure 3. The sequence was coded following best-in-class open specifications for zone-level DF sequences (Granderson et al. 2025). The sequence runs within each zone controller, dynamically adjusting zone temperature setpoints in response to DF events. During the event, the sequence gradually increases the cooling setpoint or decreases the heating setpoint, effectively reducing energy consumption. Once the event concludes, the sequence smoothly returns the setpoints to their original values within a one-hour period, thereby preventing a large electricity rebound. We refer to this as the `zone_ratcheting` sequence.

To validate the sequence before applying it to the BOPTEST building, we used a Modelica model representing a single zone of a commercial office building. The validation was conducted using two summer days in Chicago, IL. The building was occupied from 7 AM to 8 PM. The HVAC system was modeled as an ideal on-off air conditioner with hysteresis, causing cycling (10 min), a nominal heating capacity of 3000 W at a coefficient of performance (COP) of 3.5, and a nominal cooling capacity of 5000 W at a COP of 2.5. The validation results are presented in Figure 4 . The results illustrate that the zone cooling temperature setpoint (dashed purple line in Figure 4 (a)), increases at the start of the DF period, effectively reducing electrical power consumption (dashed orange line in Figure 4 (b)) to zero during that time. As the building transitions to the unoccupied period when the DF period concludes, the cooling temperature setpoint is raised. Consequently, the typical behavior of gradually decreasing the cooling setpoint back to the original occupied setpoint is not triggered, allowing the building to maintain a more relaxed temperature setting during the unoccupied period.

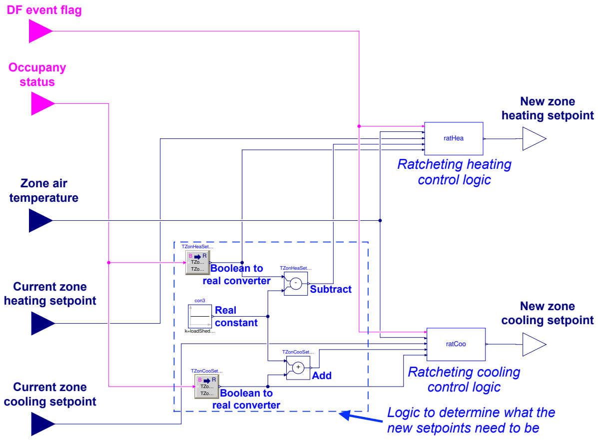


Figure 3: DF control sequence `zone_ratcheting` *implemented in CDL. Magenta connections are boolean-valued signals and blue connections are real-valued signals.*

c. Export digital specification. The CDL sequence created in the previous step includes semantic information embedded in the annotations. Figure 6 (middle left section) shows an excerpt of this semantic information exported by using the *modelica-json* tool in `.ttl` format. The content is organized in triples of `subject - predicate - object` and each statement describes how the `subject` is related to the `object` via a `predicate`. For example, the first line shows that the `zone_ratcheting` sequence should be used to `control` a `zone`. The second and third lines show what I/O points the sequence uses (e.g. `TZonHeaSetCur`, `TZonCooSetCur`, etc). The next sections show how these I/Os should bind to specific semantic information. For example, `TZonCooSetCur` should bind to a point described by the semantic definition `obc:heaset`. Objects using the `obc` prefix point to semantic definitions requiring how these concepts are represented in a semantic model, which are used to generate a SHACL requirements and SPARQL queries. Figure 6 (middle right section) also shows an example of one of these definitions for `obc:zone`. This definition prescribes that the zone must be an instance of `brick:Zone` and relate to several points using the relation `brick:hasPoint`.

d. Match requirements, query semantic model and

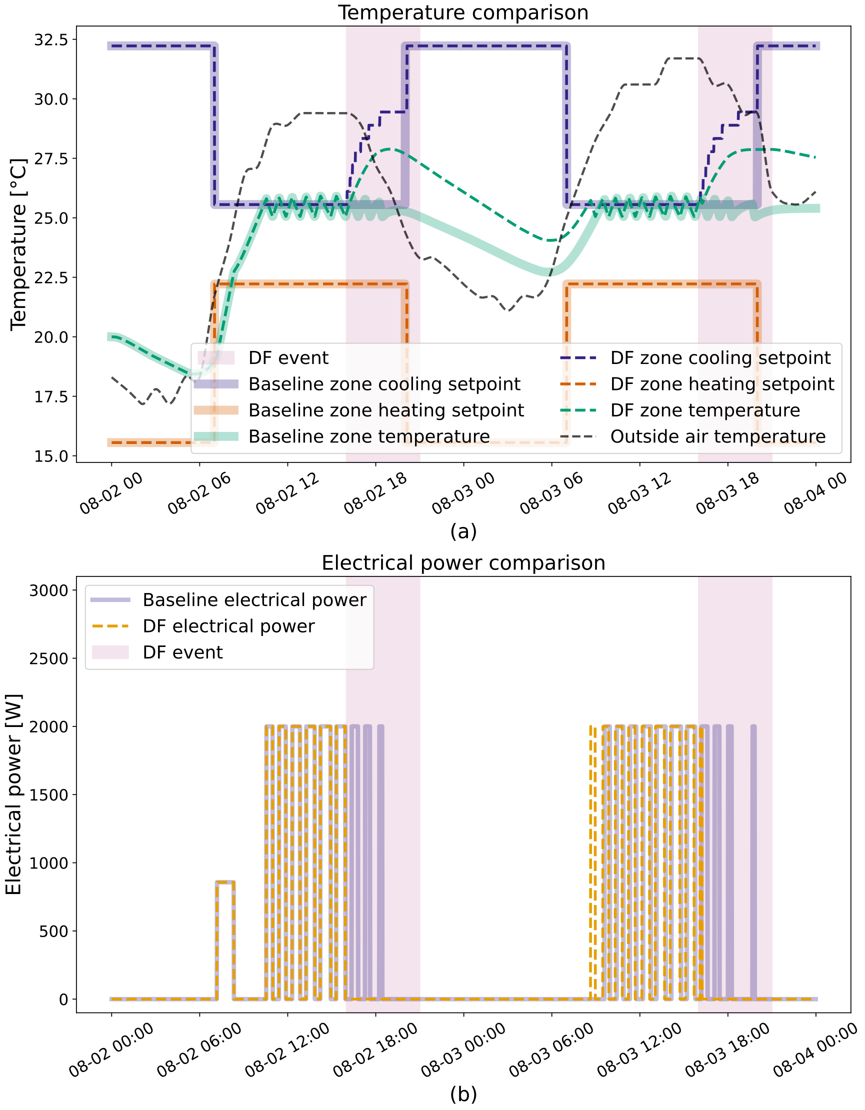


Figure 4: Results of the validation test of the `zone_ratcheting` *control sequence. Comparison of (a) zone temperatures and (b) zone electrical power measurements during the DF event and the baseline*

deploy sequence. The control vendor developed a translator[*] from CXF to JavaScript classes, which is utilized by the Normal Framework. This translator enables the conversion of CDL sequences, and it generates hooks, which are single JavaScript functions that run within the managed framework. The platform uses query blocks to retrieve the inputs to a hook, command blocks to write back the outputs from a hook and schedule blocks to determine how often the hook executes (Figure 6, top section). The combination of these query blocks, command blocks, schedule blocks, and hooks that run a control sequence in the control platform are referred to as an application.

The `zone_ratcheting` sequence was exported from CDL as CXF and imported into the Normal Framework. This sequence generated a hook called `zone_ratcheting_df`. To create the DF control application, this hook was selected, and positioned in the interface along with empty query and command blocks.

To instantiate the application, the semantic information exported from CDL and the Brick semantic model of

---

[*] Modelica Translator: https://github.com/normalframework/obc

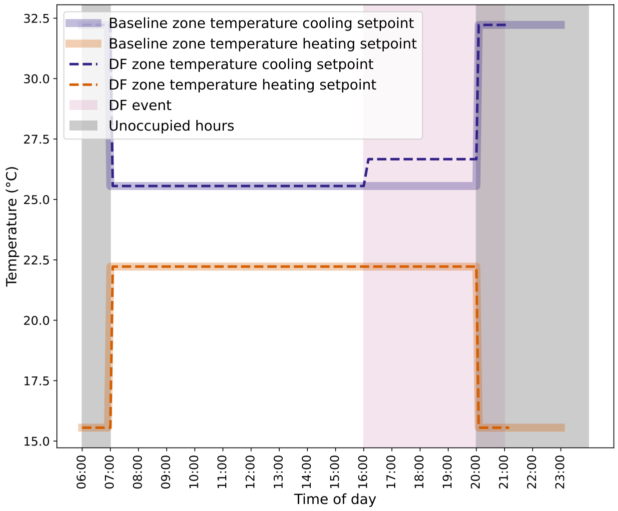


Figure 5: Results of the test of the translated DF sequence on the "East" zone on the, running on the commercial platform. The zone cooling setpoint is raised at the beginning of the event. Note that the setpoints changes at 8PM as the zone is no longer occupied)

the building were then used to update the query and command blocks, populating the query statements and matching them to the correct I/O of the hook. This was done externally to the control platform by downloading the application, running a script to automatically update the query and connect statements, and re-uploading the application to the platform. After manually configuring the hook to run every 5 minutes, the application was ready for execution. Upon running the application, the platform executes the hook for each zone. The bottom section of Figure 6 shows the configured application.

e. Verify control sequence at site. After instantiating the sequence on the vendor platform, we verified it by executing the sequence over a defined period and analyzed the corresponding input and output measurements. We then evaluated the measurements to confirm that the prescribed DF requirements were satisfied. As shown in existing 5, following the initiation of the demand response event, the zone cooling setpoint (purple) is increased in steps. This control action maintains the zone temperature within the allowable deadband, thereby reducing the need for mechanical cooling and achieving the intended load reduction during the demand-response event.

Discussion and future work

This case study demonstrated a digitalized control sequence deployment, highlighting its potential and advancing its further development and refinement. Future work will focus on these improvements:

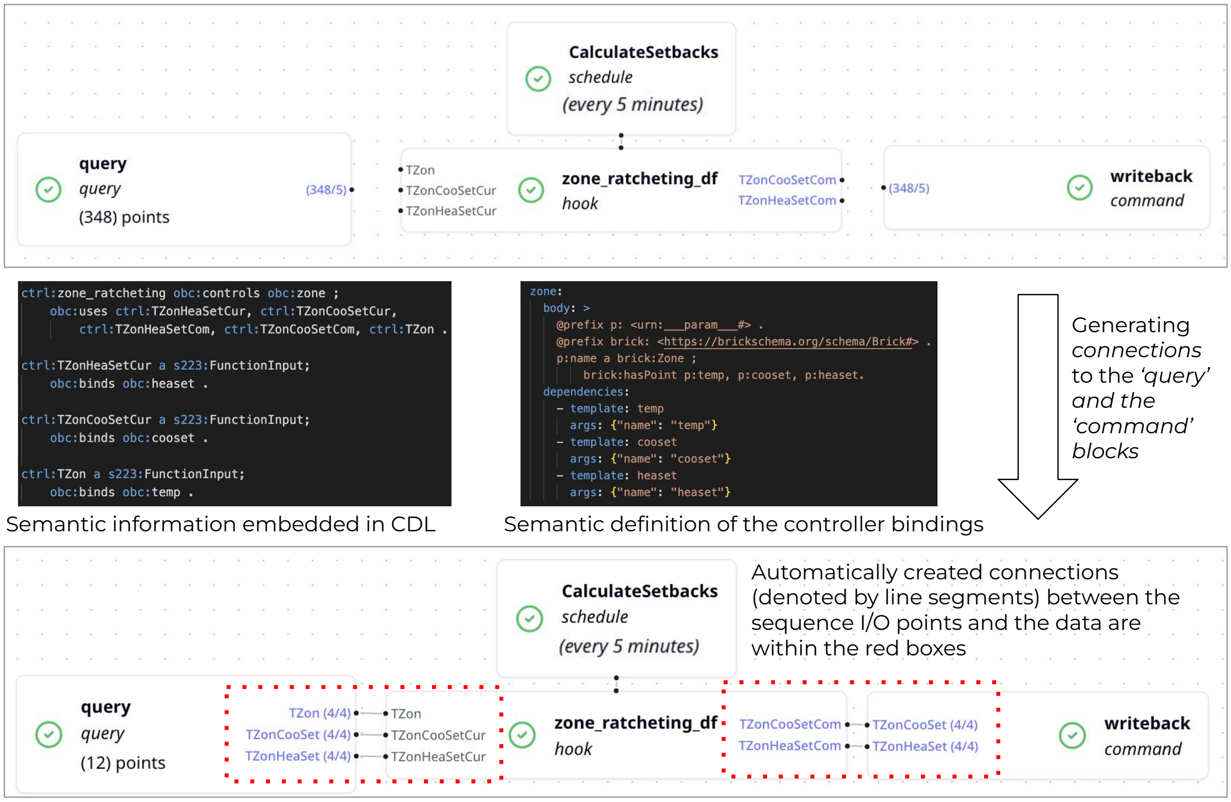
1. **Make the integration with semantic models more streamlined**. The case study highlighted a challenge in integrating the Brick semantic model, which was hosted externally, with the platform's internal metadata. The lack of integration between the two made the update process brittle. To improve this process, future efforts should focus on collaborating with vendors to internally support more standardized semantic schemas to enable a more reliable data exchange.

2. **Add DF control sequences with semantic information in the MBL**. In the case study, a new sequence was successfully developed and tested. To further enhance the framework, future efforts should focus on incorporating additional DF sequences into the MBL and conducting more thorough testing to ensure robust performance during all possible operating condition. Semantic models should also be incorporated with these models.

3. **Make the DF sequence available in ctrl-flow**. As new features of the *ctrl-flow* are rolled out, these validated DF sequences should be made available, allowing users to easily select, specify and deploy them, as outlined above.

4. **Test in real buildings**. Although the BOPTEST emulated building with its closed-loop response and realistic BACnet interface provided a robust test of the process, we plan to further validate the process in real-world buildings. This next step will allow us to gain valuable insight into practical challenges and complexities that may arise, ultimately refining our approach.

5. **Test with control practitioners**. Widespread industry adoption of these technologies and workflows hinges on its feasibility for the workforce. Consequently, future work must extend beyond technical validation to include field trials with practicing control integrators. These studies will assess the usability of emerging digital tools to determine if they effectively streamline the installation process. The goal is to quantify whether this digitalized



```
ctrl:zone_ratcheting obc:controls obc:zone ;
    obc:uses ctrl:TZonHeaSetCur, ctrl:TZonCooSetCur,
        ctrl:TZonHeaSetCom, ctrl:TZonCooSetCom, ctrl:TZon .

ctrl:TZonHeaSetCur a s223:FunctionInput;
    obc:binds obc:heaset .

ctrl:TZonCooSetCur a s223:FunctionInput;
    obc:binds obc:cooset .

ctrl:TZon a s223:FunctionInput;
    obc:binds obc:temp .
```

Semantic information embedded in CDL

```
zone:
  body: >
    @prefix p: <urn:___param___#> .
    @prefix brick: <https://brickschema.org/schema/Brick#> .
    p:name a brick:Zone ;
        brick:hasPoint p:temp, p:cooset, p:heaset.
  dependencies:
    - template: temp
      args: {"name": "temp"}
    - template: cooset
      args: {"name": "cooset"}
    - template: heaset
      args: {"name": "heaset"}
```

Semantic definition of the controller bindings

Figure 6: Translated DF sequence on the vendor platform. The top section displays the CDL logic translated into a platform-specific JavaScript "hook" called `zone_ratcheting_df`*. Using the semantic specifications in the middle section, a script programmatically populates the query and command blocks to bind the hook's I/O to specific building points, resulting in the executable application shown in the bottom section.*

approach reduces on-site configuration time and interpretation errors, thereby overcoming the scalability barriers inherent in the traditional retrofit workflow.

Conclusion

This paper presents two novel digitalized control deployment processes for existing buildings. The processes leverage recent advances in the standardized, digital control description language (CDL), semantic models (Brick), and a automated translation tool (Modelica-json) to streamline the deployment of control sequences through a digitalized workflow. A case study demonstrates the successful deployment of a demand-flexible "zone ratcheting" control sequence on a commercial control platform connected via BACnet to a virtual building using the BOPTEST emulation framework. Future work will focus on extending the process to support additional use cases (e.g. DF with electric system in the heating season), control sequences, and control platforms, as well as producing recommendations for industry stakeholders. In addition, field tests with practitioners should be conducted to uncover and mitigate practical challenges that may arise in real-world deployments.

AI Disclosure

During the preparation of this work, the authors used Anthropic Claude (Sonnet v4.5), Google Gemini (Gemini 3 Flash) and Meta Llama (Llama 4 Scout) for language editing, rephrasing and formatting. After using this tool/service, the author(s) reviewed the content rigorously, edited as needed, and take(s) full responsibility for the content of the publication.

Acknowledgments

This research was supported by the Assistant Secretary for Energy Efficiency and Renewable Energy, Office of Building Technologies of the U.S. Department of Energy, under Contract No. DE-AC02-05CH11231, and the California Energy Commission's Electric Program Investment Charge (EPIC) Program through Contract EPC-24-041.

References

ASHRAE. 2018. Standard 135: BACnet communication protocol.

Barwig, Floyd E, John M House, Curtis J Klaassen, Morteza M Ardehali, and Theodore F Smith. 2002. "The national building controls information program." *Proc. ACEEE summer study on energy efficiency in buildings*.

Becerik-Gerber, Burçin, Gale Lucas, Ashrant Aryal, Mohamad Awada, Mario Bergés, Sarah L Billington, Olga Boric-Lubecke, Ali Ghahramani, Arsalan Heydarian, Farrokh Jazizadeh, et al. 2022. "Ten questions concerning human-building interaction research for improving the quality of life." *Building and Environment* 226:109681.

Bergmann, Harry, Cory Mosiman, Avijit Saha, Selam Haile, William Livingood, Steve Bushby, Gabe Fierro, Joel Bender, Michael Poplawski, Jessica Granderson, et al. 2020. "Semantic interoperability to enable smart, grid-interactive efficient buildings." *Proc. ACEEE summer study on energy efficiency in buildings*.

Bjørnskov, Jakob, Muhyiddine Jradi, and Michael Wetter. 2025. "Automated Model Generation and Parameter Estimation of Building Energy Models Using an Ontology-based Framework." *Energy and Buildings*, jan, 115228.

Blum, David, Javier Arroyo, Sen Huang, Ján Drgoňa, Filip Jorissen, Harald Taxt Walnum, Yan Chen, Kyle Benne, Draguna Vrabie, Michael Wetter, and Lieve Helsen. 2021. "Building optimization testing framework (BOPTEST) for simulation-based benchmarking of control strategies in buildings." *Journal of Building Performance Simulation* 14 (5): 586–610.

BOPTEST. 2025. "Multizone Office Simple Air" test-case, BOPTEST. `https://ibpsa.github.io/project1-boptest/docs-testcases/multizone_office_simple_air/index.html`. Online documentation; last accessed 2025-11-07.

Chiosa, Roberto, Marco Savino Piscitelli, Marco Pritoni, and Alfonso Capozzoli. 2024. "A portable application framework for energy management and information systems (EMIS) solutions using Brick semantic schema." *Energy and Buildings* 323:114802.

Crowe, Eliot, Evan Mills, Tom Poeling, Claire Curtin, Diana Bjørnskov, Liz Fischer, and Jessica Granderson. 2020. "Building commissioning costs and savings across three decades and 1500 North American buildings." *Energy and Buildings* 227:110408.

de Andrade Pereira, Flavia, Kyriakos Katsigarakis, Dimitrios Rovas, Marco Pritoni, Conor Shaw, Lazlo Paul, Anand Prakash, Susana Martin-Toral, Donal Finn, and James O'Donnell. 2025. "A semantics-driven framework to enable demand flexibility control applications in real buildings." *Advanced Engineering Informatics* 64:103049.

Fierro, Gabe, Avijit Saha, Tobias Shapinsky, Matthew Steen, and Hannah Eslinger. 2022. "Application-driven creation of building metadata models with semantic sufficiency." *Proceedings of the 9th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*. 228–237.

Gautier, Antoine, Michael Wetter, Jianjun Hu, and Hubertus Tummescheit. 2023. "HVAC and Control Templates for the Modelica Buildings Library." oct, 217–228.

Granderson, Jessica, Armando Casillas, Marco Pritoni, and Weiping Huang. 2025. "An open control sequence specification to scale building demand flexibility via analytics software." *Energy and Buildings* 337:115616.

International Energy Agency (IEA). 2022. *Energy Efficiency Market Report 2022*. Paris: IEA.

Kao, James Y. 1992, 1992-01-01 05:01:00. HVAC functional inspection and testing guide:.

Klepeis, N. E., M. C. Nelson, Ott WR, R. Prill, W. Rabin, R. Scheu, N. Soczor, and A. Vette. 2001. "The National Human Activity Pattern Survey (NHAPS): a large-scale, cross-sectional survey of the United States." *Journal of Exposure Science and Environmental Epidemiology* 11 (1): 31–42.

Normal Software. 2025. Modelica Translator. `https://github.com/normalframework/obc`.

Paliaga, Gwelen, Rupam Singla, Coleman Snaith, Spencer Lipp, Dhananjay Mangalekar, Hwakong

Cheng, and Marco Pritoni. 2020. "Re-Envisioning RCx: Achieving Max Potential HVAC Controls Retrofits through Modernized BAS Hardware and Software." *Proc. ACEEE summer study on energy efficiency in buildings*.

Paul, Lazlo, Anand Krishnan Prakash, Armando Casillas, Tao Yang, Marco Pritoni, et al. 2025. "HP-FLEX: Field demonstration of the semantics-driven configuration of a Model Predictive Control system to make heat pumps flexible." Technical Report, Lawrence Berkeley National Laboratory (LBNL), Berkeley, CA (United States).

Pritoni, Marco, Drew Paine, Gabriel Fierro, Cory Mosiman, Michael Poplawski, Avijit Saha, Joel Bender, and Jessica Granderson. 2021. "Metadata schemas and ontologies for building energy applications: A critical review and use case analysis." *Energies* 14 (7): 2024.

Pritoni, Marco, Michael Wetter, Lazlo Paul, Anand Prakash, Weiping Huang, Steven Bushby, Parastoo Delgoshaei, Michael Poplawski, Avijit Saha, Gabe Fierro, et al. 2024. "Digital and Interoperable: The future of building automation is on the horizon. What's in it for me?" *Proc. ACEEE summer study on energy efficiency in buildings*.

Satchwell, Andrew, Mary Ann Piette, Aditya Khandekar, Jessica Granderson, Natalie Mims Frick, Ryan Hledik, Ahmad Faruqui, Long Lam, Stephanie Ross, Jesse Cohen, et al. 2021. *A national roadmap for grid-interactive efficient buildings*.

Schiller, Steven R, Jeff Deason, Greg Leventis, Lisa C Schwartz, and Kelly Parmenter. 2022. "The Future of Energy Efficiency for US Buildings-Drivers and Market Scenarios." *Proc. ACEEE summer study on energy efficiency in buildings*.

Sulzer, Matthias, Michael Wetter, Robin Mutschler, and Alberto Sangiovanni-Vincentelli. 2023. "Platform-based design for energy systems." *Applied Energy* 352:121955.

Torabi, Narges, H Burak Gunay, William O'Brien, and Toby Barton. 2022. "Common human errors in design, installation, and operation of VAV AHU control systems–A review and a practitioner interview." *Building and Environment* 221:109333.

Walther, Karl, Michael Wetter, Anand Prakash, and Jianjun Hu. 2025. "CDL-PLC translator: From Modelica HVAC control design to IEC 61131 PLC implementation." *Modelica Conferences*. 515–524.

Wetter, M., Y. Chen, K. Devaprasad, P. Ehrlich, A. Gautier, J. Hu, A. Prakash, and M. Pritoni. 2025. "Modelica Meets ASHRAE: Towards A Digital Standard for Building Control." *International Modelica Conference*.

Wetter, Michael, Paul Ehrlich, Antoine Gautier, Milica Grahovac, Philip Haves, Jianjun Hu, Anand Prakash, Dave Robin, and Kun Zhang. 2022. "OpenBuildingControl: Digitizing the control delivery from building energy modeling to specification, implementation and formal verification." *Energy* 238:121501.

Wetter, Michael, Antoine Gautier, Milica Grahovac, and Jianjun Hu. 2019. "Verification of control sequences within OpenBuildingControl." *Building Simulation 2019*, Volume 16. IBPSA, 885–892.

Wetter, Michael, Milica Grahovac, and Jianjun Hu. 2018, August. "Control Description Language." *1st American Modelica Conference*. Cambridge, MA, USA.

Wetter, Michael, Jianjun Hu, Anand Prakash, Paul Ehrlich, Gabe Fierro, Milica Grahovac, Marco Pritoni, Lisa Rivalin, and Dave Robin. 2021, September. "Modelica-json: Transforming energy models to digitize the control delivery process." *17-th IBPSA Conference*. Intern. Building Performance Simulation Assoc., 1–8.

Wetter, Michael, and Matthias Sulzer. 2024. "A call to action for building energy system modelling in the age of decarbonization." *Journal of Building Performance Simulation* 17 (3): 383–393.

Wetter, Michael, Wangda Zuo, Thierry S. Nouidui, and Xiufeng Pang. 2014. "Modelica Buildings library." *Journal of Building Performance Simulation* 7 (4): 253–270.