Title

Open building operating system: a grid-responsive semantics-driven control platform for buildings

Journal

Science and Technology for the Built Environment, 31(3)

ISSN

2374-4731

Authors

Paul, Lazlo

De Andrade Pereira, Flavia

Prakash, Anand Krishnan

et al.

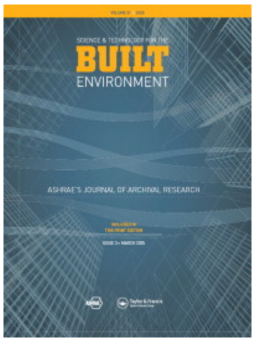
Publication Date

2025-03-16

DOI

10.1080/23744731.2024.2444819

Peer reviewed



Taylor & Francis
Taylor & Francis Group

Science and Technology for the Built Environment

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/uhvc21

Open building operating system: a grid-responsive semantics-driven control platform for buildings

Lazlo Paul, Flavia De Andrade Pereira, Anand Krishnan Prakash, Sang Woo Ham, Jingjuan Dove Feng, Rich Brown & Marco Pritoni

To cite this article: Lazlo Paul, Flavia De Andrade Pereira, Anand Krishnan Prakash, Sang Woo Ham, Jingjuan Dove Feng, Rich Brown & Marco Pritoni (2025) Open building operating system: a grid-responsive semantics-driven control platform for buildings, Science and Technology for the Built Environment, 31:3, 294-311, DOI: 10.1080/23744731.2024.2444819

To link to this article: https://doi.org/10.1080/23744731.2024.2444819

Science and Technology for the Built Environment, (2025) **31**, 294–311
This manuscript has been authored by employees of Lawrence Berkeley National Laboratory under Contract No. DE-AC02-05CH11231 with the U.S. Department of Energy. The U.S. Government retains, and the publisher, by accepting the article for publication, ackn. Published with license by Taylor & Francis Group, LLC.
ISSN: 2374-4731 print / 2374-474X online
DOI: 10.1080/23744731.2024.2444819

OPEN ACCESS Check for updates

Open building operating system: a grid-responsive semantics-driven control platform for buildings

LAZLO PAUL[1], FLAVIA DE ANDRADE PEREIRA[1,2,3], ANAND KRISHNAN PRAKASH[1], SANG WOO HAM[1], JINGJUAN DOVE FENG[4], RICH BROWN[1] and MARCO PRITONI[1]*

[1]*Building Technology and Urban Systems Division, Lawrence Berkeley National Laboratory, Berkeley, CA, USA*
[2]*School of Mechanical & Materials Engineering and UCD Energy Institute, University College Dublin, Dublin, Ireland*
[3]*CARTIF Technology Centre, Energy Division, Valladolid, Spain*
[4]*TRC Companies, Inc, Oakland, CA, USA*

Grid-interactive efficient buildings (GEBs) with flexible loads are a promising method to decarbonize buildings and support the grid. Despite the promising benefits of GEBs, automation systems managing flexible loads in response to grid signals are still uncommon in US commercial buildings. Recent literature showcases control solutions of this nature; however, they frequently depend on customized integrations that lack the essential tools and drivers required for scalability. To address these gaps, we created a workflow and software platform, called Open Building Operating System (OpenBOS). The software uses ASHRAE standard 223 P to improve scalability of GEB applications. It also provides access to a virtual testing environment, enabling the evaluation of control algorithms prior to their deployment in the field to mitigate the risk of system malfunctions or underperformance. Using the workflow and OpenBOS platform, two GEB applications were developed, deployed and tested in a simulated environment and a real building. Notably, the GEB applications significantly reduced energy costs by 28% and 23% respectively, compared to the baseline. Furthermore, amidst a shed event, demand savings amounted to 45% and 47%, while ensuring a minimal impact on comfort. The paper outlines insights gained and potential avenues for future research stemming from these novel tools.

1. Introduction and background

Grid-Interactive Efficient Buildings (GEBs) are buildings that combine a form of “smarter” control and communication technologies to optimize energy efficiency and actively provide grid services while offering a comfortable and productive environment for occupants (Neukomm, Nubbe, and Fares 2019). GEBs are pivotal in advancing global decarbonization efforts and, at scale, could reduce CO2 emissions by 80 million tons per year by 2030 (Satchwell et al. 2021). An essential facet of GEBs is demand flexibility, which is the ability to shift, shed or modulate loads within buildings. This active load management can be achieved using supervisory control schemes that operate building systems, such as lighting, heating, cooling and air conditioning (HVAC), in response to grid signals. Rule-based supervisory control schemes have been demonstrated to reduce 20% of commercial building peak load (Piette et al. 2007). More sophisticated control schemes, such as Model Predictive Control (MPC), have been applied to peak demand reduction according to time-of-use electricity rates, and have been capable of reducing demand costs by 30% in small commercial buildings (Kim and Braun 2022).

1.1. Key challenges in enabling demand flexibility for GEBs

In large commercial buildings, critical to transitioning to GEBs is the use of Building Automation Systems (BAS) to enable deployment of these “flexible” control strategies (Neukomm, Nubbe, and Fares 2019). Such systems can range in cost from tens to hundreds of thousands of dollars ($1,000-1,500 per data point) for a commercial building (Trenbath et al. 2022). The cost

Received January 27, 2024; accepted November 15, 2024
Lazlo Paul, BS, is an Associate Data Scientist. **Flavia de Andrade Pereira, MS,** Student Member ASHRAE, is a Research Associate and a PhD Candidate. **Anand Krishnan Prakash, MS,** is a Scientific Engineering Associate and a PhD Student. **Sang Woo Ham, PhD,** Associate Member ASHRAE, is a Postdoctoral Scholar. **Jingjuan Dove Feng, PhD,** Associate Member ASHRAE, is an Associate Director. **Rich Brown, MS,** is a Research Scientist. **Marco Pritoni, PhD,** Full Member ASHRAE, is a Research Scientist.
This paper is an extension of the ASHRAE conference paper Paul et al. (2023).
*Corresponding author e-mail: mpritoni@lbl.gov

of programming and testing the control sequences for a BAS can constitute up to 25% of that cost, and only increases for complex system configurations and control requirements such as demand flexibility. The factors contributing to these elevated costs include custom implementations carried out on a building-by-building basis, complexity of HVAC systems, and a shortage of qualified workforce for deploying and operating such control systems (Trenbath et al. 2022). This becomes especially challenging when operators have to deal with software from different vendors in different buildings they manage. Alternatively to modifying the existing BAS, both industry and researchers have demonstrated the effectiveness of GEB controls through modern Energy Management Information Systems, that are deployed as a software overlay on top of the BAS (Crowe, Kramer, and Granderson 2020). These platforms typically have lower capital and operating costs, with median base software costs ranging from $1,500 to $13,000 per building (Kramer et al. 2020) and reduced cost for implementation for more complicated control strategies, fostering greater scalability (Pritoni et al. 2022a, 2022b). Obstacles to their widespread adoption persist due to challenges in integrating with the underlying BAS, particularly concerning semantic interoperability (Bergmann et al. 2020). This difficulty arises from the diversity of systems, numerous configurations, and the absence of a standardized metadata schema that delineates common data points and facilitates a comprehensive understanding of the relationships among components and systems (de Andrade Pereira et al. 2024).

Similar to large buildings, the adoption of GEB solutions in small and medium commercial buildings faces obstacles due to the absence of semantic interoperability among systems. In addition they also face distinct and unique challenges. The lack of networked controls (i.e., only 13% have a BAS) and prevalence of simple but inefficient rooftop packaged HVAC units (EIA 2018), create a barrier to cost-effective upgrades for this category of buildings. This difficulty is heightened by the comparatively lower overall energy expenses and the elevated transaction costs to implement retrofits in relation to the potential savings (Katipamula et al. 2012). While more modern "light BAS" middleware platforms are now available on the market (LBNL 2023), their integration with specific hardware and software to coordinate independently controlled systems is still an obstacle to their widespread adoption. For instance, integration with different smart thermostats or networked lighting systems, still requires developing interfaces to vendor-specific Application Programming Interfaces (API), that do not share the same data model or authentication process, despite having similar functionalities (Gaidon and Poplawski 2017). Yet, given that small and medium commercial buildings constitute ∼50% of the floor space in commercial buildings and have shown considerable potential in delivering demand flexibility (Cai and Braun 2019), addressing these barriers is essential to fully harness their capabilities.

1.2. Semantic metadata models and standards

Semantic information, often called metadata, is contained in unstructured and nonstandard sources such as drawings, English language documents (e.g. point lists, API specifications), or embedded within the knowledge of personnel (Bergmann et al. 2020). This limits their discovery and interpretability, resulting in costly manual and ad-hoc point mapping processes when configuring new applications (Pritoni et al. 2021). To mitigate barriers, semantic standards for the building domain have been under investigation. Semantic models of buildings developed using these standards offer a structured approach to describing building systems and their data, representing them digitally in a consistent and standardized way (Duarte Roa et al. 2022). Semantic models are constructed according to semantic ontologies that formally define the concepts within them, enabling users to determine the meaning of available measurement or control points and their association with spaces and equipment using common descriptions. These models make information machine-readable, which allows for the seamless integration of data from various vendors and the definition of replicable, semi-automated application configuration processes (Roth et al. 2022).

Various semantic ontologies and data schemas have been developed to support the information management in buildings. Several focus on the architecture, engineering, and construction domain including IFC (ISO 2024) and BOT (Rasmussen et al. 2020). Ontologies such as FSO (Kukkonen et al. 2022) and TSO (Pauen et al. 2021) describe the topology and interconnection of building systems and can concisely represent the energy and mass flows between systems as well as the services they provide. However, these ontologies do not comprehensively represent all the information relevant to building automation system data in the operational phase, and would have to be used in conjunction with other ontologies for this use case. Other ontologies have been developed to support building operations, including SAREF (ETSI 2020), SSN/SOSA (Haller et al. 2017), Brick (Balaji et al. 2018), REC (Hammar et al. 2019), and BACS (Terkaj, Schneider, and Pauwels 2017). To date, numerous papers have explored their effectiveness across various use cases, including energy audits, Fault Detection and Diagnostics tools (FDD), and optimal control algorithms (Fierro et al. 2018; Pritoni et al. 2021; Santos et al. 2021; Delgoshaei, Heidarinejad, and Austin 2022; Mavrokapnidis et al. 2023). While broadly used in academia, these ontologies are not official standards and have not seen significant industry adoption. In the meanwhile, the industry has been embracing both proprietary and open-source metadata schemas, like Haystack (Project Haystack 2024), which have not undergone standardization through a public and widely recognized process. Over the past few years, the American Society of Heating, Refrigerating, and Air-Conditioning Engineers (ASHRAE) has been working on the development of a new standard named Standard 223 P: Semantic Data Model for Analytics and Automation Applications in Buildings[1]. Its objective is to formally define knowledge concepts and a methodology for building machine-readable and interoperable models that represent information from building systems through a Shapes

[1]https://docs.open223.info/intro.html.

Constraint Language (SHACL) (Knublauch and Kontokostas 2017) -based ontology. At the time of writing, this standard has been released for advisory public review and, although it holds the potential for widespread industry adoption, its utilization has not yet been explored in the academic literature due to the recency of its draft release.

1.3. Portable applications and control testing

Semantic models are designed to enable portable applications that can be applied across different buildings without extensive manual configuration or reprogramming during the deployment phase (Roth et al. 2022). Fierro et al. (2018) introduced MORTAR, an open-source platform featuring semantic models for more than 90 buildings. They showcased analytics applications capable of running seamlessly across multiple buildings requiring minimal changes to the code. Mavrokapnidis et al. (2023) proposed a programming model that abstracts the inputs used in application logic to enable their portability, and applies them to various FDD rules. Bennani et al. (2021) further explored how to automatically extend semantic queries to be able to adapt applications to similar building configurations. While MORTAR proves valuable for testing analytics applications, it lacks the capability to actively test controls due to the static nature of the data, which is prerecorded. Conducting simulations to test controls before deployment is a crucial stage in mitigating risk and enhancing the performance of these applications, as highlighted by Sulzer et al. (2023). Advancements in simulation tools, such as Building Optimization Testing Framework (BOPTEST) (Blum et al. 2021), have greatly simplified the process of testing control strategies before implementing them in actual buildings. BOPTEST does not natively support semantic models, but a prototype of a virtualized control network using BOPTEST and the Brick schema has been demonstrated by Fierro, Prakash, et al. (2022). Despite the potential, the tool was only tested with a simplified control strategy, controlling a single damper. Additionally, the testing environment did not support the emerging ASHRAE Standard 223 P.

1.4. Middleware architectures and control platforms

Recent academic work has established the importance of middleware and data platforms to scalably deploy smart building features (Alfalouji et al. 2022). Several different software architectures have been proposed for such platforms. Pauwels and Fierro (Pauwels and Fierro 2022) reviewed various data platforms and design patterns used for data-driven platforms utilizing semantic metadata and presented one possible representative architecture. This architecture used a classic three-layer stack model with applications, systems such as databases and storage, and a hardware abstraction layer. Each of these layers utilizes microservices to deliver features such as storage of the semantic models and time series data as well as FDD and control applications. This architecture however was not implemented and did not describe or demonstrate the services specifically needed to deliver portable controls. Additionally, it did not describe clear methods for the application of semantic metadata in the platform. Chamari, Petrova, and Pauwels (2023) proposed a similar service-oriented architecture for data-driven smart buildings. This work developed a microservice-based software architecture following the Zachman Framework and implemented it using open-source software. It integrated with semantic models using a GraphDB[2] database and demonstrated this architecture with several read-only use cases, such as integrating sensor data into a BIM model. However, this architecture was not applied to controls and lacks essential architectural elements for controls development and deployment, notably a testing module.

Research grade middleware platforms including XBOS (Prakash et al. 2020), Energon (He et al. 2021), and BRICKS (Santos et al. 2021) have incorporated semantic ontologies to increase the ease of querying data with support for graph-based self-configuration mechanisms. These have been employed in research projects to deploy advanced control schemes like MPC. VOLTTRON (Katipamula, Haack, et al. 2016), a middleware platform supported by US Department of Energy (DOE), incorporates tagging of points using Project-Haystack tags with plans to expand support to other semantic models. Many of these platforms share similar concepts, including the use of drivers/data connectors to integrate with heterogeneous data sources, use of a message bus to enable efficient communication between software agents, and APIs to enable applications to utilize data or execute controls. Such middleware platforms are essential for connecting the applications, data sources, and hardware devices that make up a building control solution, but none of them used the emerging ASHRAE standard.

1.5. Literature gaps, research questions and contributions

The literature in the preceding sections reveal various gaps. While there are multiple studies that provide research-grade platforms for building controls (Pauwels and Fierro 2022; Chamari, Petrova, and Pauwels 2023; Prakash et al. 2020; He et al. 2021; Santos et al. 2021), none of them support the emerging standard ASHRAE 223 P, and there is no evidence of their demonstrated effectiveness in control applications within real-world or simulation environments. Similarly, the BOPTEST control testing platform does not natively support semantic models. While Fierro, Prakash, et al. (2022) created a custom integration involving BOPTEST (Blum et al. 2021), BACnet (ASHRAE 2020), and the Brick Schema (Balaji et al. 2018), the authors only demonstrated control of a damper, as a simple application. A few researchers proposed workflows to develop portable applications (Fierro et al. 2018; Mavrokapnidis et al. 2023, Hviid et al. 2019, Hviid et al. 2022), but their main focus has been on analytics. Finally, very few field or simulation tests have been conducted with open-source, semantic-driven controls (Paul et al. 2023; de Andrade Pereira et al. 2024). However, neither of these studies demonstrate the use of the new semantic standard. Furthermore, de Andrade Pereira et al. (2024)

[2]https://graphdb.ontotext.com/documentation/10.5/.

did not provide a description of the software stack necessary for facilitating application portability within real buildings, and Paul et al. (2023), which is extended by this paper, did not discuss control testing services. Based on these gaps we formulated three research questions that we plan to explore with this work:

1. What is an effective workflow to develop reusable GEB applications? Do the applications developed for shedding loads exhibit the expected behavior?
2. What is the effort required to develop a control platform built on open-source software that supports ASHRAE 223P semantics? What open-source projects can we take advantage of?
3. What are the practical limitations of this approach?

This paper presents a vendor-independent software stack built using open-source software and a methodology for the deployment of portable control applications in real buildings. This software stack, called the Open Building Operating System (OpenBOS), innovates upon current building control technologies to provide a scalable method for the self-configuration, testing, and deployment of supervisory control applications. This paper expands upon the earlier work Paul et al. (2023) by incorporating support for ASHRAE Standard 223 P, developing a control testing module built on BOPTEST, and presenting a workflow for GEB application development and deployment. The paper demonstrates the robustness of the proposed approach through the testing and deployment of two portable applications across a simulated and real-building. Contributions of this paper include the following:

A. A workflow for defining and self-configuring portable applications in real buildings using the emerging ASHRAE Standard 223P
B. Identification of the software modules needed to deploy portable control applications at scale based on open-source software
C. Demonstration of the implemented control solution through testing the deployment in a simulated and in a real building.

The remainder sections of this paper are structured as follows: Section 2 describes the methodology used to address the research questions; Section 3 outlines the proposed scalable workflow; Section 4 illustrates the architecture and components of the OpenBOS platform; Section 5 details the testing and deployment of two portable control applications deployed using OpenBOS across simulated and real buildings, and presents their results; Section 6 discusses key findings in relation to the research questions; while Section 7 provides concluding remarks and outlines avenues for future work.

2. Method

To address the research questions highlighted in Section 1.5, we set out to design a workflow to develop and deploy portable GEB applications (Section 3) and a control platform that uses semantics to improve scalability of GEB applications (Section 4). The platform, referred to as OpenBOS, is based on the state-of-the-art open-source software and will integrate knowledge and software components from previous research. To evaluate the effectiveness of the workflow and platform, we implemented two GEB applications and deployed them in both a simulated building and real-world settings.

2.1. GEB applications to test

The two proposed control applications were designed to be portable and run effectively on HVAC units with zone-level thermostats.

2.1.1. Load shift application

The objective of the load shift application is to redistribute electrical load to support the grid during moments of stress. The rule-based algorithm incorporates both a load increase (take) and a load decrease (shed) strategy (Liu et al. 2022). Load take and shed periods may be identified using scheduled events or price signals, with high prices signaling a shed period. Before shed periods, the algorithm preheats or precools the building to its maximum or minimum comfort temperature. This take strategy increases the electrical load and stores thermal energy in the building mass. Then, the shed strategy adjusts zone temperature setpoints, reducing electrical loads in the building.

2.1.2. Stagger application

The objective of the second application is to stagger the operation of various HVAC units, preventing simultaneous demand peaks, similar to what is proposed by Winstead et al. (2020). The algorithm prioritizes each unit or system based on the difference between their monitored temperature and occupant-defined comfort bounds. This approach systematically rotates the operation of various units, seeking a balance between demand reduction and the delivery of HVAC service.

When integrating both applications, potential conflicts in setpoints are addressed by prioritizing the setpoints determined by the stagger application. The semantic requirements shared between the two applications, load shift and stagger, are summarized in Table 1. The Input and Output requirements describe the input and output data streams available for use by the application, while the Strategy requirements describe the required entities or system types. Lastly, the application requirements outline the sets of criteria used to validate the successful configuration of the application.

2.2 Simulated building

The modeled building is a single floor of a multizone office building located in Chicago with a total floor area of 17,900 ft^2 (1663 m^2), that uses a single-duct multi-zone VAV system, in which heating is provided by an air-to-water heat pump and cooling is provided using an air-cooled chiller[3]. It has five zones, each served by a VAV and with a

[3]https://ibpsa.github.io/project1-boptest/testcases/ibpsa/testcases_ibpsa_multizone_office_simple_air/.

Table 1. Summary of application semantic requirements.

Group	Name	Description
Input	Input-A1	A temperature measurement value must be present for the zone or equipment
	Input-B1	An occupied comfort upper limit property must be present for the zone or equipment
	Input-B2	An occupied comfort lower limit property must be present for the zone or equipment
	Input-C1	A heating setpoint must be present for the zone or equipment
	Input-C2	A cooling setpoint must be present for the zone or equipment
	Input-D1	A single setpoint must be present for the zone or equipment
Output	Output-A1	A temperature heating setpoint must be present for the zone or equipment
	Output-A2	A temperature cooling setpoint must be present for the zone or equipment
	Output-B1	A single a temperature setpoint must be present for the zone or equipment
	Output-B2	An enumerated mode property must be present for the zone or equipment
Strategy	Strategy-A1	There must be a zone with the above listed properties
	Strategy-A2	There must be an equipment with the above listed properties
Application	Stagger-A1	Input-A1, B1, and B2 must validate true. Input C1 and C2 or Input D1 must validate true. Either Output-A1 and A2 or Output-B1 and B2 must validate true. Either Strategy-A1 or A2 must validate true
	Shift-A1	Input-A1, B1, and B2 must validate true. Either Output-A1 and A2 or Output-B1 and B2 must validate true. Either Strategy-A1 or A2 must validate true

thermostat. It has a Time-of-Use (ToU) electricity rate based on an existing rate from ComEd, the utility serving the greater Chicago area, that can serve as a signal for load shifting.

2.2.1. Baseline control strategy

The baseline control of the model is based on the ASHRAE VAV 2A2-21232 control sequence (ASHRAE 2006) implemented in the Modelica Buildings Library for the VAVReheat - ASHRAE2006 model (Wetter et al. 2015). It uses Proportional Integral (PI) controllers to regulate the zone heating and cooling temperature setpoints, duct static pressure, and supply air temperature and outside air flow rate. The control scheme incorporates a schedule-based supervisory control scheme with six operating modes, determining the baseline setpoints and equipment enable/disable configurations. Occupancy is scheduled between 6:00 AM and 7:00 PM daily, with unoccupied periods outside of this timeframe.

2.2.2. Test scenarios

Four different scenarios are planned in the simulated building: a baseline scenario using a standard occupancy schedule and thermostat setback; the shift application, which adjusted the heating setpoint in response to a dynamic ToU electricity rate; the stagger application, which rotated the operation of each zone to reduce coincident demand; and a final scenario running both the shift and stagger applications. The lower thermal comfort boundary is set to 66.2 °F (19 °C) and the preheating period is set for 2-h before the shed event.

2.3. Real building

The chosen building for the demonstration is a small office building located in New York. It comprises a single zone covering 3,780 ft^2 (351.2 m^2) of office space, serviced by a dual-fuel heating system consisting of five ductless variable speed Heat Pumps (HPs) and one attic-mounted Gas Furnace (GF). The existing HPs did not have any internet connectivity and were controlled by IR remotes. To enable demand flexibility using OpenBOS, we had to install Wi-Fi enabled IR remotes. These remotes could still communicate with the HPs over IR, but also supported the Wi-Fi protocol for communicating with external software. These remotes were used to monitor the space and change setpoints of the HPs and a WiFi-enabled thermostat was used for the GF, with readings stored every five minutes. Electrical loads for each of the HPs were measured minutely using WiFi-enabled electricity meters. Testing took place over a four-day period during the heating season, during which the baseline control strategy and test control strategy were each run for two days. Detailed information of this site can be found in Paul et al. (2023).

2.3.1. Baseline control strategy

The baseline control scenario involves schedule-based setpoint operation, with the Gas Furnace (GF) being operated using “droop” control. This implies that the GF is assigned a lower heating setpoint than the HPs to avoid its operation when the HPs could adequately provide heating. During the occupied period (7:00 AM − 8:00 PM), the HPs had a heating setpoint of 70 °F (21 °C), and the GF had a heating setpoint of 68 °F (20 °C). During the unoccupied period, all systems operated with a setpoint of 60.8 °F (16 °C). The HPs were put into heat mode because this test occurred during the heating season. Information about the devices and baseline control strategy can also be found in Paul et al. (2023).

2.3.2. Test scenario

Given the limited time available for testing in the real building, a single scenario was tested, employing both the shift and stagger applications. During the preheating strategy, the

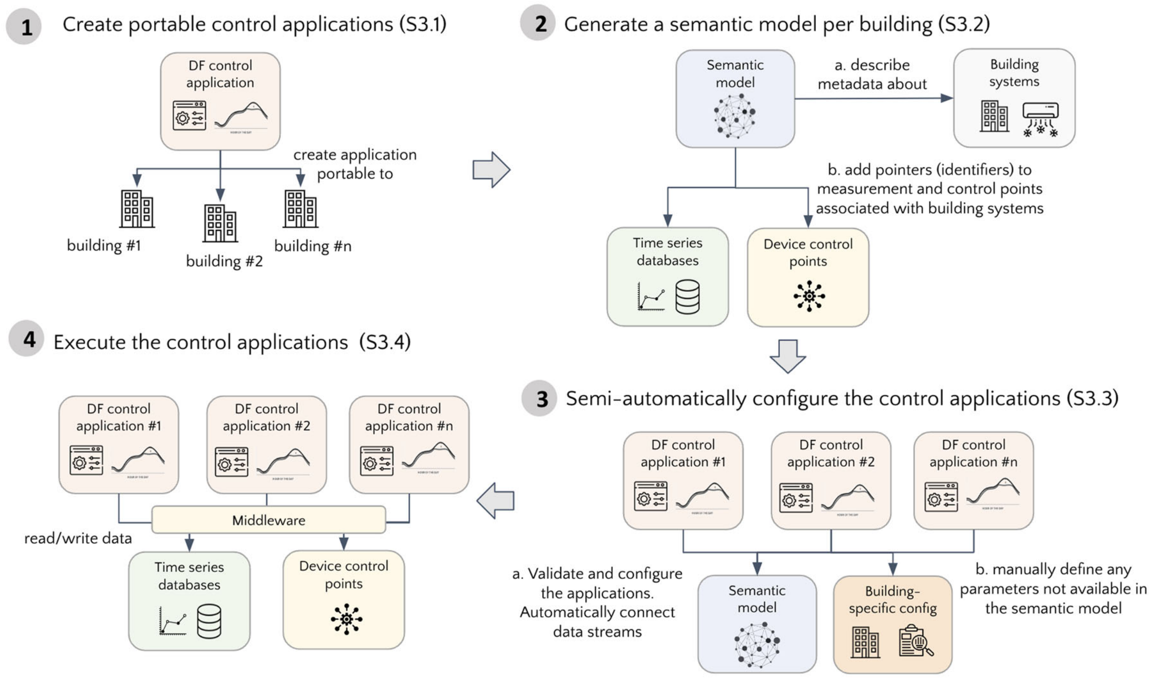


Fig. 1. An illustration of the proposed workflow to deploy portable GEB applications leveraging semantic models and open-source middleware adapted from Paul et al. (2023).

HP units are used in order to reduce furnace runtime and carbon emission. In order to maintain occupant comfort while reducing energy consumption, the control uses conservative lower thermal comfort boundary of 66.2 °F (19 °C), and a 2-h period to preheat before a shed event. The site is sent a shed signal from 7:00 to 9:00 AM, when winter heating loads in NY typically cause a demand peak (NationalGrid 2023). The GEB applications were configured using Brick as well as 223 P, resulting in identical configured applications.

2.4. Evaluation metrics

The two testbeds and GEB applications verified the workflow and the correct functioning of the OpenBOS platform. Beyond verifying these design objectives, the performance of the GEB applications was assessed in terms of their capability to shift site loads without generating new demand peaks. This evaluation utilized the following metrics:

- Energy during preheat (kWh)
- Energy during shed (kWh)
- Peak power during shed (kW)
- Energy cost ($/day)
- Furnace Runtime (Hrs/day) [field test]
- GHG Reduction (kgCO2e/day) [field test]

3. Design of the workflow

Figure 1 shows the proposed workflow to deploy portable GEB control applications through OpenBOS that consists of four steps. This procedure was initially introduced in Paul et al. (2023) and is comprehensively outlined in the current paper. First (Section 3.1), we design portable applications along with the necessary semantic requirements. The applications must be generalizable across various heterogeneous buildings, avoiding to hard-code details of a specific building, such as point names. Second (Section 3.2), we create or obtain a 223 P semantic model for a given building. The model should describe the metadata about the building systems and measurement/control points, as well as provide pointers to their respective sources, to read and write data (e.g., time series databases and control points via the middleware drivers). Third (Section 3.3), the portable application is semi-automatically configured by extracting required pointers and other metadata through automated semantic queries bundled with the application. Certain applications may require a limited amount of information that is not in the 223 P model, such as occupant comfort constraints or price signals and those will be included as separate configuration files. Finally (Section 3.4), the application is executed, utilizing the middleware for data reading and writing.

3.1. Create portable control applications

To create portable control applications that can be generalizable across heterogeneous buildings, this work follows the requirements proposed by de Andrade Pereira et al. (2024). We define that such controls need to be flexible to available data, adaptable to changing operating conditions, abstract to specific buildings, and built by modular functions. Moreover, for defining the controls data requirements, while enabling an unambiguous and consistent application validation and configuration process, this work

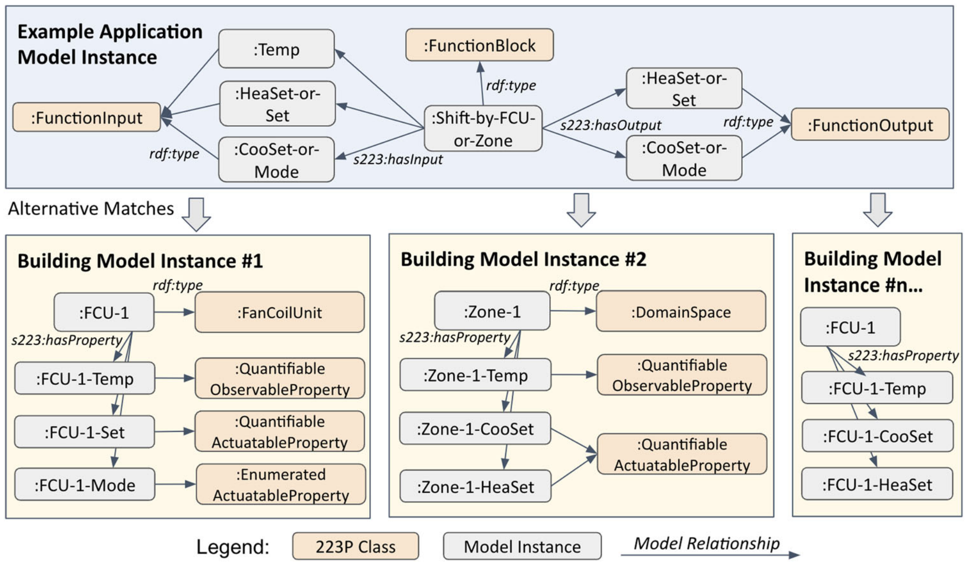


Fig. 2. Example of alternative building system configurations that can support a portable application modeled with 223 P. Specific properties and equipment are represented using classes from the 223 P ontology, which define general concepts: temperature setpoints are modeled using the 223 P class QuantifiableActuatableProperty, fan coil units using the 223 P class FanCoilUnit, Inputs to the shift application using the 223 P class FunctionInput, etc. Though it is not shown, the amount of Inputs and Outputs can vary to match building instances.

uses the standards 223 P and Shapes Constraint Language (SHACL) (Knublauch and Kontokostas 2017) which describe them as machine-readable semantic requirements. SHACL shapes specify the expected structure of semantic data, and thus can be used to constrain the possible building system configurations applicable to a given portable application. This is illustrated in Figure 2, which outlines two of the possible setpoint configurations: independent heating and cooling setpoints and single setpoint with a fixed deadband. It also displays equipment-level and zone-level relationships. This is significant because, for instance, while the load shifting application can be applied to both setpoint configuration, such as relaxing thermostat setpoints to decrease demand, the system configuration has an impact on how the application operates. In the first case, the temperature deadband can be expanded to decrease HVAC demand. However, in the second case, widening the deadband is not possible. Therefore, the setpoint must be adjusted either upward or downward, depending on the current HVAC mode, to achieve a similar outcome as widening the deadband.

Primarily, semantic requirements can be used to automatically identify if a building has an applicable system type and the necessary data points to run the application through the process of validation. If not, these requirements can provide detailed information about why the application can not be configured (Fierro, Saha, et al. 2022) which can reduce the debugging process when deploying an application. Subsequently, these requirements can be used to facilitate application configuration. This is achieved by using the 223 P concepts of function blocks. Function blocks are associated with the various building system configurations that the application may support. They specify building point configurations as potential input and output alternatives, and also map to modular functions that are used to interpret these alternatives in the portable application. The determination of the valid block inputs and outputs for a specific building, depends on the available data in the building model, and can be used to semi-automatically configure the application. This topic is further discussed in the following subsections.

3.2. Generate semantic model of the building

Several approaches have been investigated to automate the process of generating semantic models (Koh et al. 2018; Mishra et al. 2020), or to import and enrich them from the earlier phases of building design (Fierro et al. 2020). In this paper, we leverage BuildingMOTIF[4] to assist the generation of semantic models built upon the ASHRAE Standard 223 P. BuildingMOTIF is a toolset that helps to create semantic models following a well-defined workflow process that ensures they are constructed correctly and can support applications, utilizing concepts described by Fierro, Saha, et al. (2022). This process leverages SHACL representations of applications to drive model creation and ensure that semantic models satisfy the desired use cases. The created models include, for example, building thermal zones, HVAC systems and their measurement and control points, as well as the relationships between all of them. While the created models do not include telemetry (real-time or historical) data related to the operation of the building systems, they provide their meaning and context, as well as their addressing

[4]https://github.com/NREL/BuildingMOTIF.

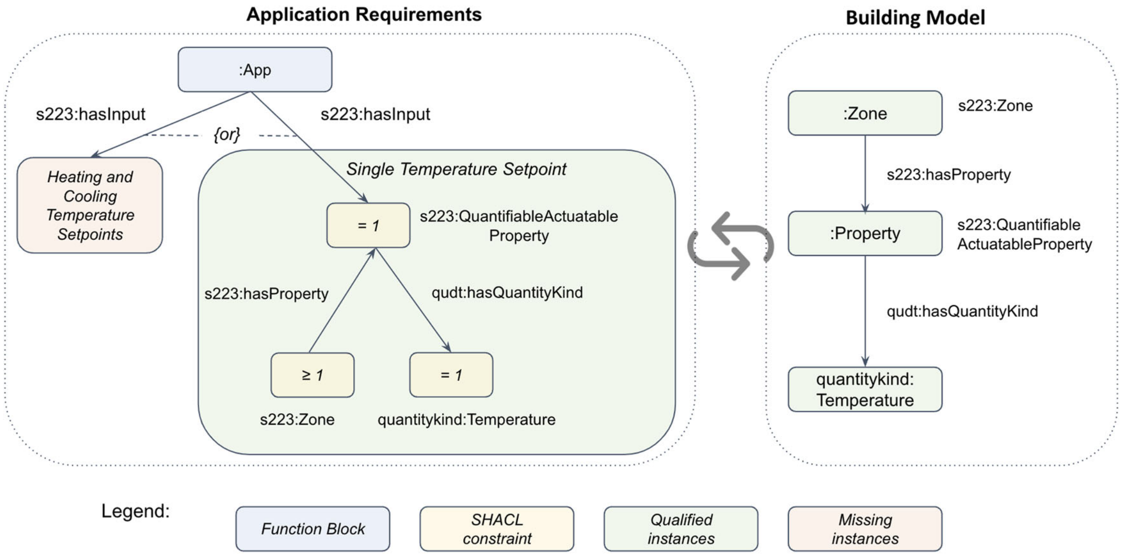


Fig. 3. Sample of alternative application requirements represented as SHACL constraints being validated against a compliant building model. The constraints describe the information that must be present in the model. For example, "> =1" means that there must be one or more s223:Zone related by s223:hasProperty to the s223:QuantifiableActuatableProperty.

information (i.e., pointers to where to find and query them via the OpenBOS middleware drivers and database). Once generated, the semantic models are used by OpenBOS to validate and to configure the applications.

3.3. Semi-automatically configure the control applications

As described in Section 3.1, applications are represented using function blocks with semantic requirements that refer to building configurations and also link to modular application functions. During validation, the semantic models from the buildings are checked against the semantic requirements of an application, which can be expressed using various alternatives, shown in Figure 3. For the requirements that are successfully validated, the building semantic model is enriched by inferring links between the data points that meet the applications' requirements within the building models to the respective instances in the applications' function block models. This means that if applications are successfully validated, their model's instances can be added to the building models. Once this is established, such links are used to query the necessary data points in each building and identify which application functions should be used to semi-automatically configure the application. This process is driven by machine-readable semantic requirements, simplifying and clarifying the application configuration process and providing a promising step toward automatic application deployment.

3.4. Execute the control application

The applications are executed through the OpenBOS platform connected to a virtual building in BOPTEST and a real building in New York State.

4. Design and development of the OpenBOS software platform

Figure 4 shows the required software modules and how they have been implemented in OpenBOS leveraging other open-source software. The top "application" level includes portable applications and user-oriented interfaces such as a data-dashboard for monitoring and analysis. The middle "data infrastructure" layer includes drivers that abstract the different communication pathways between hardware devices and external data sources. They harmonize the formats and data structures of the data retrieved from these sources to enable data aggregation, normalization, and controls. This layer also includes cyber-secure communication capabilities between the services and drivers and multiple databases (and interfaces) including a semantic model file-store for building semantic models and application semantic requirements, and a time-series database for historical data. The bottom layer contains the building measurement and control points (hardware) that are being read from and controlled respectively, and external data sources that are used by the applications. It also includes a module providing access to a simulated control testing environment to test controls before their deployment in real buildings.

4.1. Middleware and drivers

The middleware is the translation and communication layer that enables the integration between heterogeneous data sources and various services. These services may be portable applications designed for the platform, software provided from various vendors, or other control systems running in a building (e.g., a BAS). The middleware provides a uniform interface for applications to receive data from and control devices, and thus it is essential to the deployment of portable applications. While a semantic model decouples an

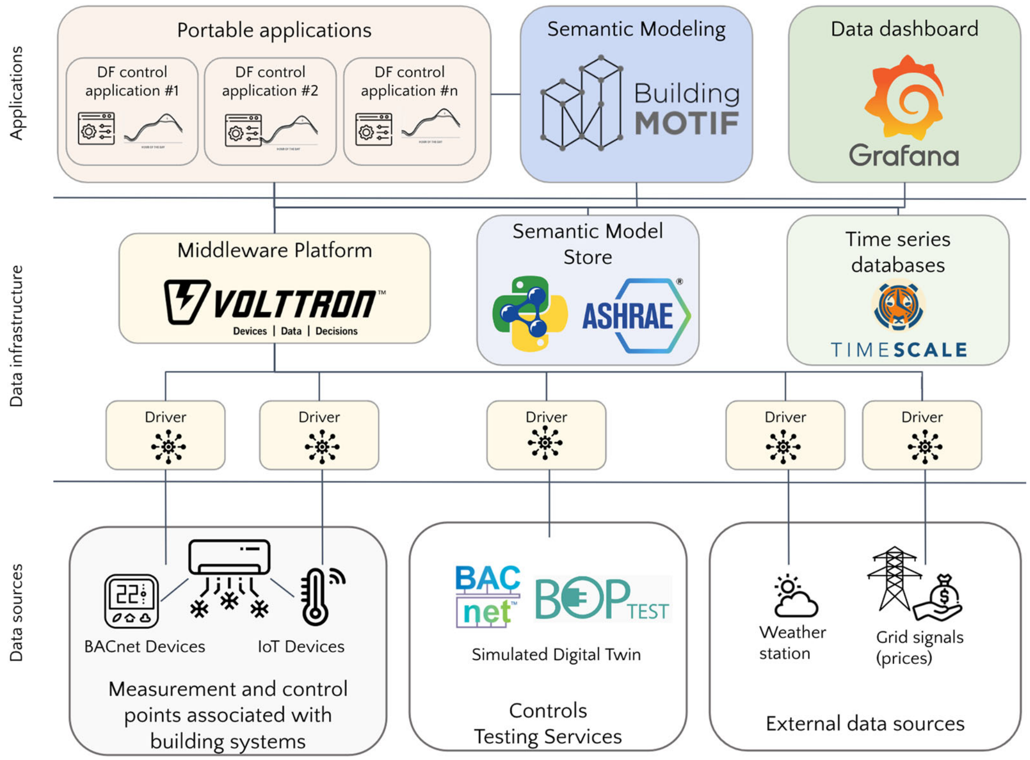


Fig. 4. Components of OpenBOS architecture and used services, adapted from Paul et al. (2023).

application from building specific representation of information, the middleware decouples applications from the diverse communication protocols, APIs, and data structures used in buildings. Cybersecurity is an essential feature of a middleware platform, because middleware handles communication and access control for each service and can also provide an interface to run custom programs.

OpenBOS uses VOLTTRON (Katipamula, Haack, et al. 2016) as the middleware platform. VOLTTRON handles data capture and retrieval using an agent-based approach that allows one to easily create new drivers to integrate diverse types of devices and external data sources. It is used by both research projects (Katipamula, Lutes, et al. 2016, Kim and Katipamula 2017; Winstead et al. 2020) and commercial products (Intellimation. Intellimation Innovation 2024; ACE IoT Solutions (ACE). About Eclipse VOLTTRON 2024). By integrating with a proven middleware, the portable applications presented in this paper may see wider deployment in existing implementations or by developers familiar with its use. It also has continual support, especially in the area of cybersecurity. Detailed threat profiles have been published for several releases including its current version, describing possible vulnerabilities and mitigation measures for deployments (Himes et al. 2021).

VOLTTRON has existing drivers for common protocols, such as BACnet and Modbus, enabling a simple setup procedure to start communicating with devices that support these protocols. It also supports integration with IoT devices that communicate via HTTP API (e.g.: thermostats, lighting control systems, environmental sensors) based on vendor-specific data models. It does so by providing a Python-based programming environment to develop new device drivers. Semantic models augment the capabilities of this approach, seamlessly integrating applications with diverse new devices with minimal manual effort.

4.2. Validation and inference engine

Application requirements expressed using semantic ontologies such as 223 P contain machine-readable SHACL rules that can be used to validate semantic models and add new information to them when certain conditions are met. The execution of these rules against a semantic model requires SHACL-based engines. Several open-source SHACL validation and inference engines are available including pySHACL[5], which is utilized by OpenBOS.

4.3. Data storage

For the deployment of portable control applications at least two types of databases are required. A time-series database (or equivalent) is needed to store the large amount of time-series historical data generated by data points in buildings. This data may be collected at irregular frequencies and may consist of enumerated or floating point values. Additionally, another database is required for the storage of the semantic model of the building. Triple-store databases (data is stored as a triple of subject-predicate-object) or file-stores are commonly employed to store and query these semantic models.

[5]https://github.com/RDFLib/pySHACL

Semantic models for buildings must also be updated for any changes within them.

OpenBOS currently uses a Timescale[6] database to handle the time-series data storage. Timescale is an open-source extension to PostgreSQL that is a relational database built to handle large amounts of time-series data. Due to the typically small size of semantic models for buildings, OpenBOS currently uses the Python library RDFLib[7] to manage semantic models and a file-store to store them. Yet, should the size and complexity of these semantic models increase, or if there is a requirement for more sophisticated querying capabilities, OpenBOS will have to shift toward utilizing a dedicated triple store.

4.4. Monitoring

To allow seamless monitoring of building data, OpenBOS leverages the open-source dashboard Grafana[8]. Grafana has flexible alerting capabilities and can be used to create interactive and dynamic dashboards by querying and transforming data from an attached database and then displaying it in a matrix of panels. Monitoring capabilities are an essential complement to control applications deployed in real buildings because they allow users to assess control application performance. This functionality provides the users of the control applications and the building operators real-time status information of the different building systems and ensures that the building is behaving as expected.

4.5. Controls testing service

OpenBOS provides support for model-based testing of the control algorithm before real-world deployment, by leveraging the SDT framework (Fierro, Prakash, et al. 2022). The goal of the SDT framework is to "integrate simulation software, control network virtualization and semantic metadata" to produce virtual buildings with BACnet endpoints that could be read from or written to, replicating a digital interface to a real "brick and mortar" building. SDT uses BOPTEST, as simulation engine (Blum et al. 2021). BOPTEST contains a collection of ready-to-use building HVAC system models (called "testcases") developed in Modelica[9]. These models capture realistic physical dynamics such as HVAC system pressure-flow dynamics. BOPTEST includes a baseline control strategy and allows the overwriting of the control signals to the components within the models. Alongside the BOPTEST web-based API, the SDT framework provides a real-time virtual BACnet network. This network functions as an interface that is more commonly recognized by professionals in the building industry. Additionally, it incorporates an overlaid Brick model to establish mappings between BOPTEST variables and BACnet objects.

OpenBOS integrates with the SDT framework to provide an environment for application developers to test their control applications, as well as to facilitate their portability from the testing phase to real-world deployment in buildings, shown in Figure 5. It has also extended the SDT framework by developing ASHRAE Standard 223 P models and interfaces to the BOPTEST testcases, thereby supporting ASHRAE Standard 223 P-based requirements for portable applications. This approach enables an evaluation of control applications across multiple virtual buildings, in an environment that closely resembles a real building. OpenBOS also supports modifying BOPTEST testcases' parameters to evaluate applications under different conditions such as multiple utility rates, weather conditions and GHG emission rates.

5. Deployment and test results

5.1. Test of the workflow and software stack

The deployment followed the workflow steps proposed in Section 3: 1) the applications were coded in Python and used SHACL (Section 3.1) to express the semantic requirements listed in Table 1; 2) the ASHRAE 223 P semantic models of the two buildings were developed using BuildingMOTIF; 3) the applications were configured and validated as described in Section 3.3 using pySHACL (Section 4.2). Occupant comfort boundaries as well as demand flexibility event details, specific to each building, were also added using configuration files; 4) the applications were initiated and operated in the two buildings without requiring reprogramming, effectively showcasing the capabilities of the platform.

5.1.1. Test of the OpenBOS GEB control application in the simulated building

The OpenBOS platform was connected to the BACnet interface of the SDT using a VOLTTRON BACnet driver, and the application was executed in real time. The demonstration was conducted during the heating season, so the applications were tested for 14 days of typical heating operation, during which the functionality of the applications was assessed based on how they affected the zone temperatures and the electrical load of the building.

5.1.2. Test of the OpenBOS GEB control application in the real building

The OpenBOS platform was connected to the five HP units and the GF installed at this site via VOLTTRON drivers.

5.2. Test results in the simulated building

Figure 6 shows the zone temperature setpoint adjustments made in each scenario described in Section 2.2.2, for an example zone. The shift application scenario used a ToU price signal to identify a shed period, and increased the setpoint before the shed period to preheat the building, then decreased it to reduce the electrical load. The stagger application scenario intermittently deactivated HVAC service for the zone by widening the temperature setpoint deadband to

[6]https://www.timescale.com/.
[7]https://github.com/RDFLib/rdflib
[8]https://grafana.com/.
[9]https://modelica.org/.

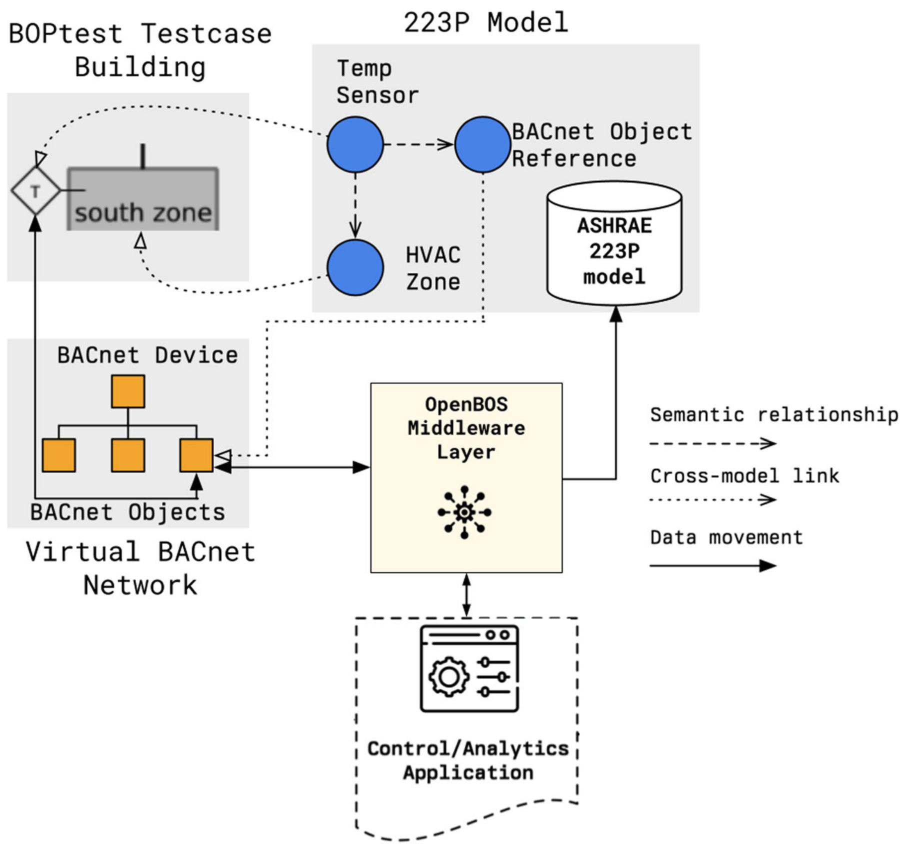


Fig. 5. The major components and relationships of the SDT testing service integrated with OpenBOS, adapted from Fierro, Prakash, et al. (2022).

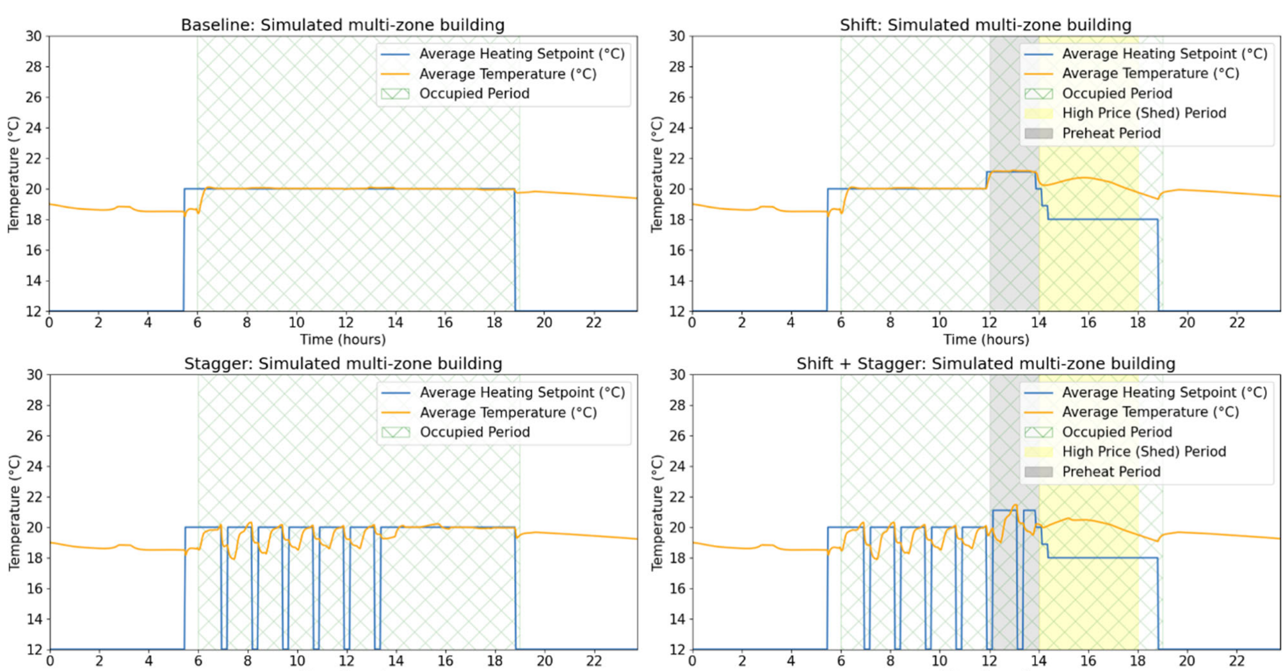


Fig. 6. Zone temperature setpoint and actual temperatures in each scenario for an example zone in the simulated building.

discourage simultaneous HVAC operation. The scenario with both applications running in the building showed similar behavior to the shift application, increasing the heating setpoint to preheat the building and decreasing it to reduce electrical load, with a couple periods where the deadband is widened to rotate the operation of HVAC units. Though

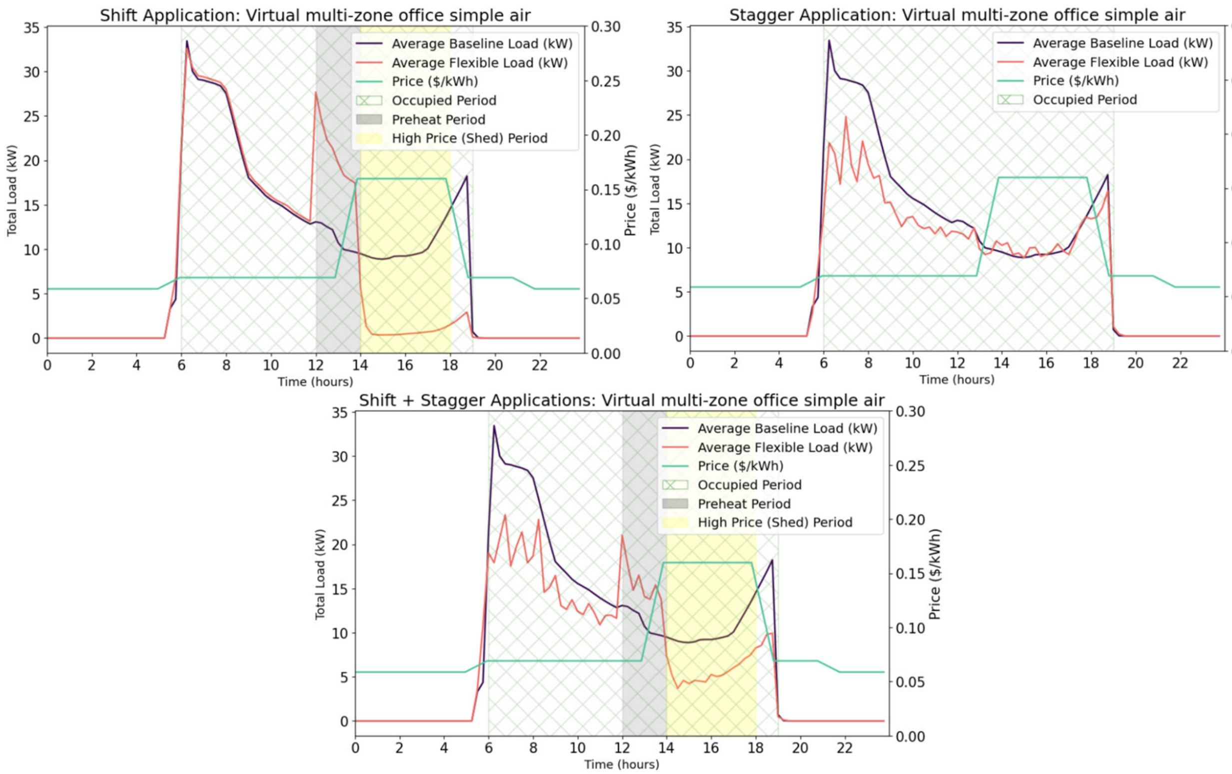


Fig. 7. Electric load profiles of the applications compared to the baseline for each scenario in the simulated building.

Table 2. Performance comparison between baseline and the semantics-driven GEB control application in the shift and stagger scenario in the simulated building.

Performance metrics	Baseline	GEB	Percent Change*
Energy during preheat (kWh) (12:00 - 14:00)	22.3	31.8	+39.8%
Energy during shed (kWh) (14:00 - 18:00)	39.3	21.6	−45.0%
Peak power during shed (kW) (14:00 - 18:00)	27.0	22.9	−15.5%
Energy cost ($/day)	19.3	13.9	−28.1%

*negative % represents a reduction, while positive % represent an increase compared to the baseline

setpoint changes fit expectations, the room temperature changes rapidly showcasing fast thermal dynamics likely caused by low thermal mass.

Figure 7 shows 15-minute average electric load profiles generated by the application compared to baseline load. The stagger application primarily reduces the period of high demand when morning warm up begins, when all the units call for heat in baseline operation. The shift application preheats the building before the peak price period, increasing loads, and significantly reducing them subsequently. The load staggering and load shifting application run at the same time show a mixture of the two results, lowering the intensity of both the morning and preheating event peaks.

The performance of the shift and stagger scenarios is presented in Table 2. The results are as expected, with an increase in energy consumption during the preheating event by 40% and a significant decrease during the shed event (45%). The peak power during the shed period is also reduced by 16% and the daily cost reduced by 28%.

5.3. Test results in the real building

Figure 8 illustrates the adjusted temperature setpoints following the flexible control compared to the baseline control and their resulting temperature, while Figure 9 shows the corresponding electric load profiles. Electric load is aggregated at 15 min intervals to align with the standard smart revenue meter aggregation period. Changes in the electric load align with expectations with a significant amount of load shifted from the load shed period to the preheating period preceding it. The results of this load shedding are also shown by the lower average room temperature, calculated by averaging the temperature readings from each WiFi-enabled device, in the flexible scenario when compared to the baseline during the shed period. After the load shedding period, electrical load remains slightly lower throughout the rest of the day, likely due to reduced overall HVAC operation from rotating the operation of each unit. This is corroborated by the room temperature measurements, which are slightly lower throughout the day when compared to baseline. Otherwise, temperature measurements in the

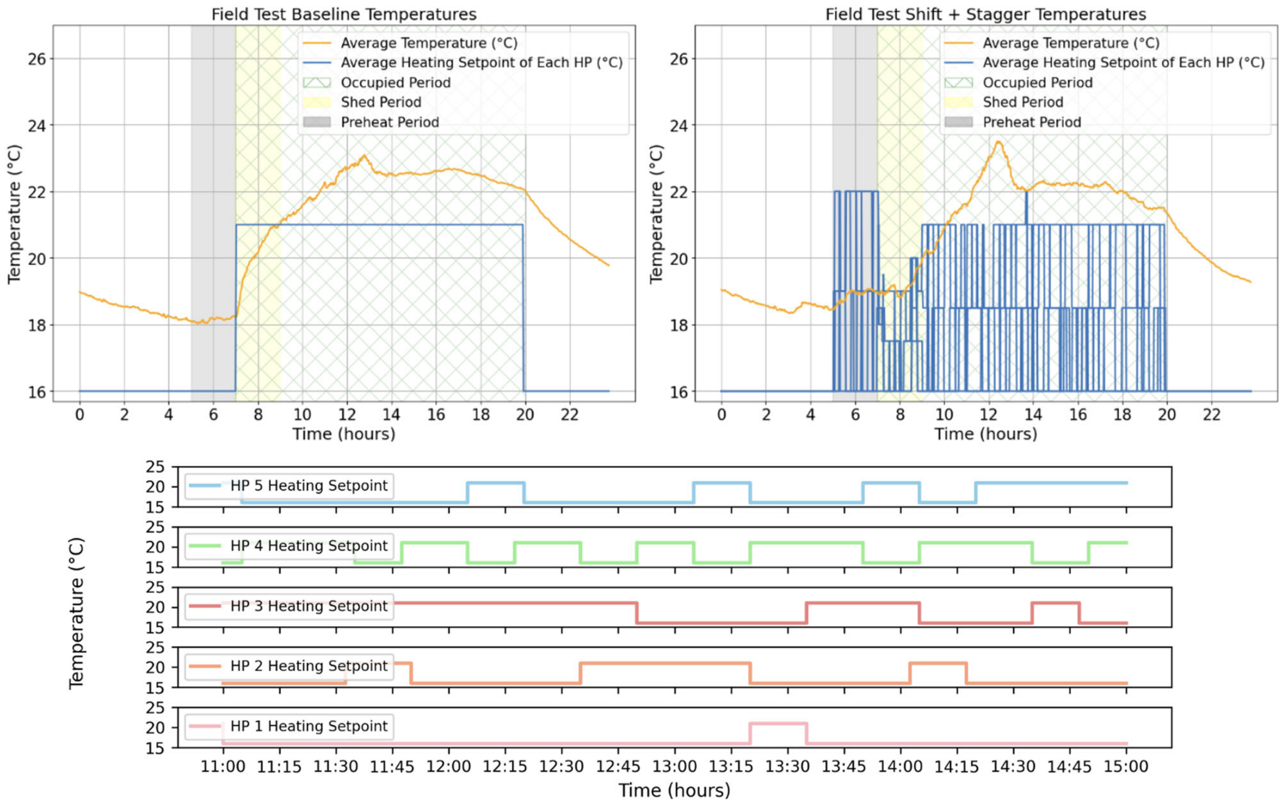


Fig. 8. Zone temperature setpoints for all HPs and actual temperatures for the baseline (left) and application (right) in the real building; magnified setpoint changes for each heat pump, showing the behavior of the stagger application changing setpoints every 15 min or more (bottom).

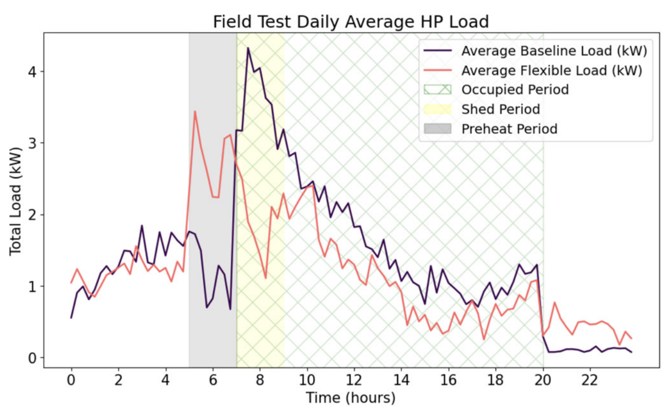


Fig. 9. Average electric load profiles of the applications compared to the baseline in the real building.

baseline and flexible scenarios are similar during occupancy. In the flexible scenario, thanks to some preheating by the HP units the room temperature remains around the lower comfort bound throughout the shed event, and reaches the normal lower comfort bound around 10:00. In the baseline scenario, the lower comfort bound is reached slightly earlier, around 9:00. The scenarios both maintain room temperatures within comfort bounds for roughly the same amount of time, 91.4% in the baseline scenario and 91.1% in the flexible scenario. Lower average room temperatures during the test period may also be caused by on average lower outdoor ambient temperatures during the test period. Based on National Weather Service Data[10], the average outdoor temperature was 41.7 °F (5.4 °C) during the baseline period and 39.9 °F (4.4 °C) during the test period. The measured room temperatures diverge from the setpoints setpoint because the temperatures were reported by IoT IR remotes located in occupant workspaces. These temperature measurements were different from the temperatures at the HP units, which used their own temperature sensors for control.

Table 3 presents the results of the test. The GEB application shifted the load from the shed period (47% reduction) to the preheating period (128% increase). During the shed period, the peak power was also reduced by 19%. Throughout the day the energy costs were reduced by 23% and the furnace runtime was reduced by 35%, resulting in a reduction in GHG emissions of 16%, based on a simple calculation using the EPA GHG equivalencies calculator (EPA 2024). To compare the outdoor temperature and heating conditions of the baseline and test periods, we use the Heating Degree Hours (HDH) method, which is a commonly used approach for approximating the amount of heating load for a building (Letherman and Al-Azawi 1986; Kajewska-Szkudlarek 2023). Using the standard HDH external temperature balance point of 65 °F (18.3 °C) and hourly temperature data for the location provided by the National Weather Service, the baseline period had 1146 °F (619 °C) HDH and the test period 1232 °F (667 °C) HDH,

[10]https://www.weather.gov/.

Table 3. Performance comparison between baseline and the semantics-driven GEB control application for the real building.

Performance metrics	Baseline	GEB	Percent Change*
Energy during preheat (kWh) (05:00 - 07:00)	2.40	5.48	+128%
Energy during shed (kWh) (07:00 - 09:00)	7.18	3.84	−46.5%
Peak power during shed (kW) (07:00 - 09:00)	4.47	3.61	−19.2%
Energy cost ($/day)	5.79	4.46	−23.0%
Furnace Runtime (Hrs/day)	0.53	0.34	−35.2%
GHG Reduction (kgCO2e/day)	27.1	22.7	−16.2%

*negative % represents a reduction, while positive % represent an increase compared to the baseline

showing that the baseline and test periods had similar conditions. Adjusting the energy cost and GHG emissions per HDH, we see that the application saved approximately 28.4% energy cost per HDH and 22.2% GHG emissions per HDH.

6. Discussion

In this paper, we have introduced a workflow and a software stack (OpenBOS) designed for deploying portable control applications in buildings. The successful demonstration of this software and methodology uncovered essential considerations, providing insights to address the research questions outlined in Section 1.5.

6.1. Workflow and development of portable applications (RQ1)

Our work demonstrates successful configuration and deployment of two rule-based control applications using OpenBOS and the proposed workflow. These applications were deployed with minimal manual reconfiguration effort in a simulation environment and a real building. In both scenarios, they performed as expected and showcased the ability to shift the load and reduce the cost in response to grid signals, with limited or no impact on comfort. While used as a proof of concept, these simple applications are broadly applicable to many system types, because they operate on thermostatic temperature setpoints common in many buildings. The availability of GEB applications that can be configured semi-automatically can support practitioners in scaling up deployment of GEB controls, by reducing labor and skills necessary for such tasks. The utilization of modular, clearly defined semantic requirements and functions for these applications maintains their understandability and customizability while enabling automation. This is critical, as it allows engineers applying them to buildings to understand and contribute to these applications while being supported by the developed tools and workflow.

6.2. Software architecture, effort required (RQ2), and lessons learned

OpenBOS was designed as a complete software solution taking advantage of several open-source software projects, including VOLTTRON (Katipamula, Haack, et al. 2016), Timescale[11], BACpypes[12], pySHACL[13], BOPTEST (Blum et al. 2021), BuildingMOTIF[14] and Grafana[15], as well as the emerging standard ASHRAE 223 P[16], as illustrated in Figure 4. The software architecture, described in Section 4, is in accordance with parallel research endeavors, as evidenced by works such as those by Pauwels and Fierro (2022) and Chamari, Petrova, and Pauwels (2023). The open-source approach may reduce long-term maintainability due its many dependencies. However, this is mitigated by the choice of widely used and well-maintained software and semantic web standards with multiple compliant substitutable software packages. OpenBOS demonstrates utilization of semantic metadata to promote its usage among other software platforms, which should be augmented by the open-source approach. The overall effort to develop this research-grade proof-of-concept software platform, two simple GEB applications and two 223 P semantic models was relatively contained, requiring a few months of work from the research team. We anticipate that stakeholders participating in various tasks will vary based on the business models of technology vendors and the organizational structure of building owners/operators. For example, software developers may handle the development of the platform and applications, enabling easier reuse across multiple buildings. On the other hand, energy consultants or building owners may be responsible for developing semantic models and instantiating applications, using the same semantic model for a specific building to streamline deployment and reduce costs. Alternatively, a single company may handle both application development and deployment. Though pinpointing the exact effort needed to develop a commercial product similar to this prototype is challenging, the availability of numerous open-source software packages implies that the undertaking may not be excessively demanding.

Several features of OpenBOS simplified deployment in a real building: 1) the middleware layer and semantics-driven application configuration process streamlined the process of integrating with unique devices, including smart IoT IR remotes; 2) the semi-automated configuration process and the abstraction and separation between control requirements and

[11]https://www.timescale.com/.
[12]https://github.com/JoelBender/bacpypes.
[13]https://github.com/RDFLib/pySHACL
[14]https://github.com/NREL/BuildingMOTIF.
[15]https://grafana.com/.
[16]https://docs.open223.info/intro.html.

implementation with specific devices, protocols and applications programming interfaces (APIs), allow deployment of the same application across multiple different buildings. This capability applies to both simulations and real-world buildings and gains greater significance as the number of similar buildings available for application deployment grows.

Some challenges in automating these processes still remain: 1) In the case-study tested, the IoT IR remote devices can set setpoints and modes for the operation of the HP mini-split units, but do not receive feedback from the units, and the temperature measurement reported is not the one used for control. The lack of interoperable communication with the HP mini-split units limits the ability to precisely control the units. While the applications tested do not require direct control or knowledge of heating and cooling operation state, more advanced control applications such as model predictive control (MPC) would require this information. 2) Capturing occupant preferences in an automated process is challenging. Communication and discussion with occupants to describe the changes they will expect in HVAC operation and to determine their preferences for building operation required significant time and attention. Occupant satisfaction with control applications is critical to their continued usage and subsequent impact, and future work should investigate how to enhance and streamline occupant interaction.

6.3. Limitations (RQ3), and future work

Several limitations emerged during this research. First, the self-configuration step in the workflow relies on the semantic model of a building, however some information necessary for the application is not found in the model. For example, the shift application above requires a parameter defining the length of the preheating period, but that is not contained in the new standard 223 P. Other more advanced control applications may require model parameters such as the thermal resistance of the envelope (Drgoňa et al. 2020). This aspect hinders complete automation of the process, calling for further research to tackle the issue. Some of the parameters to advanced control applications may be expressed using other existing ontologies such as BOT and IFC, which can effectively represent architectural concepts in buildings, FSO, which can describe topology in addition to energy flows, or TSO, which can describe the interconnection of building service systems and link them to spatial structures.

Second, the control applications demonstrated are supervisory in nature, that is they determine setpoints of lower-level algorithms that are directly responsible for the control of the HVAC components (de Andrade Pereira et al. 2024). This underlying logic has a significant impact on the actual behavior of the control system, but it is not included in the semantic model. For example, the presented stagger application cycles the operation of different HVAC units, relying on lower-level logic to modulate the compressors, reduce short cycling and protect the equipment. This logic is often embedded in the equipment, but if it is not, then a staggered control application should account for this in its configuration. Fortunately, there is a proposed complementary standard, ASHRAE Standard 231 P, aimed at making control logic descriptions interoperable and machine-readable (ASHRAE 2024). Future work should investigate the integration of 223 P and 231 P and its use in the proposed workflow and control software platform, and should compare this approach to other ontologies representing building control information, such as BACS.

Thirdly, this study focuses solely on two simple applications and two semantic models of buildings. As more applications are tested and deployed on additional buildings, there may arise a need to expand these applications to accommodate optional features and alternative configurations. The exploration of the iterative process of updating applications and building models is a subject deserving more attention in future research.

Future work should also explore the development of standard and open-source control libraries for GEB applications, including more complex algorithms applicable to more system types and shared with the research community. Additionally, the portability of the proposed applications in this paper should be further validated across different buildings of different vintages, and the practical challenges of porting applications from simulations to real buildings should be addressed. A commissioning process should also be investigated to ensure that building systems function correctly prior to the deployment of portable applications. This is essential to mitigate the challenges in translating an application from an idealized simulation to a real building, especially for older buildings that are more likely to be affected by faults and in need of retro-commissioning. Furthermore, the performance assessment of these applications should extend to field evaluations spanning different climate zones and building types to provide results that will be more relevant to the diverse characteristics of buildings that may benefit from implementing OpenBOS.

Further efforts are also required to enhance the robustness and broad applicability of these software platforms and applications. It is crucial to create tools that empower non-coders to effortlessly choose and set up applications, as well as enabling them to craft semantic models. Creating more test cases in BOPTEST, which reflect real building types, can also support more thorough testing of applications before deployment, reducing the risk of system malfunction or underperformance. More research is also needed to quantify the time and labor required by this approach, compared to a more traditional workflow and to identify practical implementation challenges that need to be addressed.

7. Conclusion

This work outlines the creation of a workflow designed for deploying portable GEB applications in buildings. The workflow is composed of four sequential steps: 1) creation of portable applications, 2) generation of the semantic model for the building, 3) semi-automated configuration of the control application, and 4) execution of the control application.

We have also developed and tested a vendor-neutral software stack, named OpenBOS, leveraging semantics to enhance the scalability of GEB applications. The software is built on state-of-the-art open-source software and supports the upcoming ASHRAE semantic standard 223 P to facilitate interoperability of applications. Furthermore, it facilitates the virtual testing of applications through a virtual testbed, known as BOPTEST, integrated with a BACnet network and a semantic model of the building. Utilizing this workflow and platform, two GEB applications were successfully developed, deployed and tested in both a simulated and a real building. The instantiation of applications for each building was achieved with minimal manual effort for mapping them to specific data and control points. Remarkably, the GEB applications demonstrated substantial reductions in energy costs by 28% and 23% in the respective buildings. Moreover, during a shed event, demand was reduced by 45% and 47%, all while maintaining a negligible impact on comfort. This successful deployment of OpenBOS and the portable control applications highlights the potential for significant cost and effort savings in the deployment and operation of GEB controls across multiple buildings.

Acknowledgements

The authors would like to acknowledge the leadership of Gwelen Paliaga, support from the LBNL team, Armando Casillas, Weiping Huang, Peter Grant and David Blum, from Gabe Fierro at Colorado School of Mines and efforts of the ASHRAE 223P committee and strike teams.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This work was supported by the Assistant Secretary for Energy Efficiency and Renewable Energy, Building Technologies Office, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231, by the New York State Energy Research & Development Authority (NYSERDA) through the NextGen HVAC Innovation Challenge program, by California Energy Commission through grant EPC-19-013, and by the CBIM-ETN funded by the European Union Horizon 2020 research and innovation program under the Marie Skodowska-Curie grant agreement No 860555. This manuscript has been authored by employees of Lawrence Berkeley National Laboratory under Contract No. DE-AC02-05CH11231 with the U.S. Department of Energy. The U.S. Government retains, and the publisher, by accepting the article for publication, acknowledges, that the U.S. Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for U.S. Government purposes.

ORCID

Lazlo Paul http://orcid.org/0000-0002-8819-5863
Flavia De Andrade Pereira http://orcid.org/0000-0002-2019-7571
Anand Krishnan Prakash http://orcid.org/0000-0002-3694-3225
Sang Woo Ham http://orcid.org/0000-0003-1776-2610
Marco Pritoni http://orcid.org/0000-0003-4200-6905

References

ACE IoT Solutions (ACE). About Eclipse VOLTTRON. 2024. Accessed January 25, 2024. https://aceiotsolutions.com/volttron/

Alfalouji, Q., T. Schranz, A. Kümpel, M. Schraven, T. Storek, S. Gross, A. Monti, D. Müller, and G. Schweiger. 2022. IoT middleware platforms for smart energy systems: An empirical expert survey. *Buildings* 12 (5):526. 10.3390/buildings12050526.

ASHRAE. 2006. *Sequences of operation for common HVAC systems*. Atlanta, GA: ASHRAE.

ASHRAE. 2020. Standard 135-2020 – BACnet—A data communication protocol for building automation and control networks (ANSI Approved). Accessed January 25, 2024. https://www.techstreet.com/standards/ashrae-135-2020?product_id=2191852

ASHRAE. Titles, purposes, and Scopes. TPS - titles, purposes, and scopes. Accessed January 25, 2024. https://www.ashrae.org/technical-resources/standards-and-guidelines/titles-purposes-and-scopes#231p

Balaji, B., A. Bhattacharya, G. Fierro, J. Gao, J. Gluck, D. Hong, A. Johansen, J. Koh, J. Ploennigs, Y. Agarwal, et al. 2018. Brick: Metadata schema for portable smart building applications. *Applied Energy* 226:1273–92. 10.1016/j.apenergy.2018.02.091.

Bennani, I. L., A. K. Prakash, M. Zafiris, L. Paul, C. Duarte Roa, P. Raftery, M. Pritoni, and G. Fierro. 2021. Query relaxation for portable brick-based applications. Proceedings of the 8th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation, November 17, 2021. 10.1145/3486611.3486671

Bergmann, H., C. Mosiman, A. Saha, S. Haile, W. Livingood, S. Bushby, G. Fierro, et al. 2020. Semantic interoperability to enable smart, grid-interactive efficient buildings. *ACEEE Summer Study on Energy Efficiency in Buildings*, Held Virtually (United States), August 17–20, 2020. https://www.osti.gov/biblio/1735554

Blum, D., J. Arroyo, S. Huang, J. Drgoňa, F. Jorissen, H. T. Walnum, Y. Chen, K. Benne, D. Vrabie, M. Wetter, et al. 2021. Building optimization testing framework (BOPTEST) for simulation-based benchmarking of control strategies in buildings. *Journal of Building Performance Simulation* 14 (5):586–610. 10.1080/19401493.2021.1986574.

Cai, J., and J. E. Braun. 2019. Assessments of demand response potential in small commercial buildings across the United States. *Science and Technology for the Built Environment* 25 (10):1437–55. DOI: 10.1080/23744731.2019.1629245.

Chamari, L., E. Petrova, and P. Pauwels. 2023. An end-to-end implementation of a service-oriented architecture for data-driven smart buildings. *IEEE Access*. 11:117261–81. 10.1109/access.2023.3325767.

Crowe, E., H. Kramer, and J. Granderson. 2020. *EMIS applications showcase: Highlighting applications of energy management and* *information systems*. Berkeley, CA: Lawrence Berkeley National Laboratory, . https://smartenergyanalytics.org/assets/EMIS%20Showcase.pdf

de Andrade Pereira, F., L. Paul, A. Casillas, A. Prakash, W. Huang, M. Pritoni, C. Shaw, S. Martin-Toral, D. Finn, and J. O'Donnell. 2024. Enabling portable demand flexibility control applications in virtual and real buildings. *Journal of Building Engineering* 86: 108645. 10.1016/j.jobe.2024.108645

Delgoshaei, P., M. Heidarinejad, and M. A. Austin. 2022. A semantic approach for building system operations: Knowledge representation and reasoning. *Sustainability* 14 (10):5810. 10.3390/su14105810

Drgoňa, J., J. Arroyo, I. Cupeiro Figueroa, D. Blum, K. Arendt, D. Kim, E. P. Ollé, J. Oravec, M. Wetter, D. L. Vrabie, et al. 2020. All you need to know about model predictive control for buildings. *Annual Reviews in Control* 50:190–232. 10.1016/j.arcontrol.2020.09.001.

Duarte Roa, C., P. Raftery, R. Sun, L. Paul, A. Prakash, M. Pritoni, G. Fierro, and T. Peffer. 2022. Towards a stronger foundation: Digitizing commercial buildings with brick to enable portable advanced applications. Berkeley, CA: UC Berkeley. http://dx.doi.org/10.20357/B7ZG6R

Energy Information Administration (EIA). 2018. Commercial buildings energy consumption survey (CBECS) 2018. Accessed January 25, 2024. https://www.eia.gov/consumption/

Environmental Protection Agency (EPA). n.d. Greenhouse gasses equivalencies calculator—calculations and references. Accessed January 25, 2024. https://www.epa.gov/energy/greenhouse-gases-equivalencies-calculator-calculations-and-references

European Telecommunications Standards Institute (ETSI). 2020. Smart applications reference ontology and onem2m mapping, ETSI TS 103 264 V3.1.1 (2020-02). https://www.etsi.org/deliver/etsi_ts/103200_103299/103264/03.01.01_60/ts_103264v030101p.pdf

Fierro, G., A. K. Prakash, D. Blum, J. Bender, E. Paulson, and M. Wetter. 2022. Notes paper: Enabling building application development with simulated digital twins. In *Proceedings of the 9th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BuildSys '22)*, 250–53. New York, NY: Association for Computing Machinery. 10.1145/3563357.3564060.

Fierro, G., A. Prakash, C. Mosiman, M. Pritoni, P. Raftery, M. Wetter, and D. E. Culler. 2020. Shepherding metadata through the building lifecycle. In *Proceedings of the 7th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, 70–79. BuildSys '20. New York, NY, USA: Association for Computing Machinery. 10.1145/3408308.3427627

Fierro, G., A. Saha, T. Shapinsky, M. Steen, and H. Eslinger. 2022. Application-driven creation of building metadata models with semantic sufficiency. In *Proceedings of the 9th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (*BuildSys '22*)*, 228–37. New York, NY, USA: Association for Computing Machinery. 10.1145/3563357.3564083

Fierro, G., M. Pritoni, M. AbdelBaky, P. Raftery, T. Peffer, G. Thomson, and D. E. Culler. 2018. Mortar: An open testbed for portable building analytics. In *Proceedings of the 5th Conference on Systems for Built Environments, 172–81. BuildSys '18*. New York, NY, USA: Association for Computing Machinery.

Gaidon, C., and M. Poplawski. 2017. Connected lighting system interoperability study part 1: Application programming interfaces. *United States*. 10.2172/1414814.

Haller, A., K. Janowicz, S. Cox, D. L. Phuoc, K. Taylor, and M. Lefrancois. 2017. Semantic sensor network ontology (SSN)—W3C Recommendation. *Open Geospatial Consortium (OGC) & World Wide Web Consortium (W3C).* Accessed January 25, 2024. https://www.w3.org/TR/vocab-ssn.

Hammar, K., E. O. Wallin, P. Karlberg, and D. Hälleberg. 2019. The realestatecore ontology. In *International Semantic Web Conference*, 130–145. Springer.

He, F., Y. Deng, Y. Xu, C. Xu, D. Hong, and D. Wang. 2021. Energon: A data acquisition system for portable building analytics. In *Proceedings of the Twelfth ACM International Conference on Future Energy Systems (e-Energy '21)*, 15–26. New York, NY, USA: Association for Computing Machinery. 10.1145/3447555.3464850

Himes, C., P. O'Connell, G. Seppala, T. Simmons, A. Steinmetz, and C. Younkin. 2021. *VOLTTRON Threat Profile for VOLTTRON Version 8.0*. Richland, WA: Pacific Northwest National Laboratory (PNNL).

Hviid, J., A. Johansen, F. C. Sangogboye, and M. B. Kjærgaard. 2019. Enabling auto-configuring building services: The road to affordable portable applications for smart grid integration. In *Proceedings of the 10th ACM International Conference on Future Energy Systems*, 68–77. Association for Computing Machinery. 10.1145/3307772.3328288

Hviid, J., A. Johansen, G. Fierro, and M. B. Kjærgaard. 2022. Service portability and information discovery in building operating systems using semantic modeling. In *Proceedings of the 11th International Conference on the Internet of Things (IoT '21)*, 110–117. New York, NY, USA: Association for Computing Machinery. 10.1145/3494322.3494337

Intellimation. Intellimation Innovation. 2024. Accessed January 25, 2024. https://intellimation.io/innovation/

International Standards Organization (ISO). 2024. ISO 16739-1:2024 2024. *Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries*. International Standards Organization

Kajewska-Szkudlarek, J. 2023. Predictive modeling of heating and cooling degree hour indexes for residential buildings based on outdoor air temperature variability. *Science Reports* 13:17411. 10.1038/s41598-023-44380-4.

Katipamula, S., J. Haack, G. Hernandez, B. Akyol, and J. Hagerman. 2016. VOLTTRON: An open-source software platform of the future. *IEEE Electrification Magazine* 4 (4):15–22. 10.1109/MELE.2016.2614178

Katipamula, S., R. Lutes, G. Hernandez, J. Haack, and B. Akyol. 2016. Transactional network: Improving efficiency and enabling grid services for buildings. *Science and Technology for the Built Environment* 22 (6):643–54. DOI: 10.1080/23744731.2016.1171628.

Katipamula, S., R. M. Underhill, J. K. Goddard, D. J. Taasevigen, M. A. Piette, J. Granderson, R. E. Brown, S. M. Lanzisera, and T. Kuruganti. 2012. *Small- and medium-sized commercial building monitoring and controls needs: A scoping study*. Richland, WA, USA: Pacific Northwest National Lab (PNNL).

Kim, D., and J. E. Braun. 2022. MPC solution for optimal load shifting for buildings with ON/OFF staged packaged units: Experimental demonstration, and lessons learned. *Energy and Buildings* 266: 112118. 10.1016/j.enbuild.2022.112118

Kim, W., and S. Katipamula. 2017. Development and validation of an intelligent load control algorithm. *Energy and Buildings* 135 (2017):62–73. 10.1016/j.enbuild.2016.11.040

Knublauch, H., and D. Kontokostas. 2017. Shapes constraint language (SHACL). W3C. Accessed January 25, 2024. https://www.w3.org/TR/shacl/

Koh, J., D. Hong, R. Gupta, K. Whitehouse, H. Wang, and Y. Agarwal. 2018. Plaster: An integration, benchmark, and development framework for metadata normalization methods. In *Proceedings of the 5th Conference on Systems for Built Environments (BuildSys '18)*, 1–10. New York, NY, USA: Association for Computing Machinery. 10.1145/3276774.3276794

Kramer, H., G. Lin, C. Curtin, E. Crowe, and J. Granderson. 2020. Building analytics and monitoring-based commissioning: industry practice, costs, and savings. *Energy Efficiency* 13 (3):537–49. 10.1007/s12053-019-09790-2.

Kukkonen, V., A. Kücükavci, M. Seidenschnur, M. H. Rasmussen, K. M. Smith, and C. A. Hviid. 2022. An ontology to support flow system descriptions from design to operation of buildings. *Automation in Construction* 134:104067. 104067. 10.1016/j.autcon.2021.104067.

Lawrence Berkeley National Laboratory (LBNL). 2023. Smarter small buildings campaign: Categorized listing of packaged RTU controls. Accessed January 25, 2024. https://smartersmallbuildings.lbl.gov/sites/default/files/2023-10/Categorized_Listing_of_Packaged_Rooftop_HVAC_Unit_RTU_Controls.pdf

Letherman, K. M., and M. M. J. Al-Azawi. 1986. Predictions of the heating and cooling energy requirements in buildings using the degree hours method. *Building and Environment* 21 (3-4):171–6. 1986ISSN 0360-1323, 10.1016/0360-1323(86)90026-0.

Liu, J., R. Yin, L. Yu, M. A. Piette, M. Pritoni, A. Casillas, J. Xie, T. Hong, M. Neukomm, and P. Schwartz. 2022. Defining and applying an electricity demand flexibility benchmarking metrics framework for grid-interactive efficient commercial buildings. *Advances in Applied Energy* 8 (December 2022):100107. 10.1016/j.adapen.2022.100107.

Mavrokapnidis, D., G. Fierro, M. Husmann, I. Korolija, and D. Rovas. 2023. SeeQ: A programming model for portable data-driven building applications. In *Proceedings of the 10th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BuildSys '23)*, 159–168. New York, NY, USA: Association for Computing Machinery. 10.1145/3600100.3623744

Mishra, S., A. Glaws, D. Cutler, S. Frank, M. Azam, F. Mohammadi, and J. S. Venne. 2020. Unified architecture for data-driven metadata tagging of building automation systems. *Automation in Construction* 120:103411. 10.1016/j.autcon.2020.103411

NationalGrid. 2023. *Demand Response Program*. Accessed March 29, 2023. https://www.nationalgridus.com/NY-Business/EnergySaving-Programs/Demand-Response.

Neukomm, M., V. Nubbe, and R. Fares. 2019. Grid-interactive efficient buildings technical report series: Overview of research challenges and gaps. United States. 10.2172/1577966.

Pauen, N., D. Schlütter, J. Frisch, and C. van Treeck. 2021. TUBES system ontology: Digitalization of building service systems. *CEUR Workshop Proceeding* 3081:43–54.

Paul, L., F. de Andrade Pereira, S. Ham, M. Pritoni, R. Brown, and J. Feng. 2023. Open building operating system: An open-source grid responsive control platform for buildings. ASHRAE Annual Conference 2023. Tampa, FL.

Pauwels, P., and G. Fierro. 2022. A reference architecture for data-driven smart buildings using brick and LBD ontologies. In *Proceedings of the REHVA 14th HVAC World Congress (CLIMA 2022).*

Piette, M. A., D. Watson, N. Motegi, and S. Kiliccote. 2007. Automated critical peak pricing field tests: 2006 pilot program description and results. Berkeley, CA: Lawrence Berkeley National Laboratory (LBNL). LBNL-59351. https://drrc.lbl.gov/publications/automated-critical-peak-pricing-field.

Prakash, A., K. Zhang, P. Gupta, D. Blum, M. Marshall, G. Fierro, P. Alstone, J. Zoellick, R. Brown, and M. Pritoni. 2020. Solar + optimizer: A model predictive control optimization platform for grid responsive building microgrids. *Energies* 13 (12):3093. 10.3390/en13123093.

Pritoni, M., D. Paine, G. Fierro, C. Mosiman, M. Poplawski, A. Saha, J. Bender, and J. Granderson. 2021. Metadata schemas and ontologies for building energy applications: A critical review and use case analysis. *Energies* 14 (7):2024. 10.3390/en14072024.

Pritoni, M., G. Lin, Y. Chen, R. Vitti, C. Weyandt, and J. Granderson. 2022a. From fault-detection to automated fault correction: A field study. *Building and Environment* 214:108900. 10.1016/j.buildenv.2022.108900.

Pritoni, Marco, G. Lin, Y. Chen, J. House, E. Crowe, and J. Granderson. 2022b. Market barriers and drivers for the next generation Fault Detection and Diagnostic Tools. *ACEEE Summer Study on Energy Efficiency in Buildings* 2022 10.20357/B7801T.

Project Haystack. n.d.. Project Haystack Website (Haystack 3 Schema). Accessed January 25, 2024. https://www.project-haystack.org

Rasmussen, M. H., M. Lefrançois, G. F. Schneider, and P. Pauwels. 2020. BOT: The building topology ontology of the W3C linked building data group. *Semantic Web* 12 (1):143–61. 10.3233/SW-200385.

Roth, A., M. Wetter, K. Benne, D. Blum, Y. Chen, G. Fierro, M. Pritoni, A. Saha, and D. Vrabie. 2022. Towards digital and performance-based supervisory HVAC control delivery. *ACEEE Summer Study on Energy Efficiency in Buildings* 2022. http://dx.doi.org/10.20357/B70G62.

Santos, G., T. Pinto, Z. Vale, R. Carvalho, B. Teixeira, and C. Ramos. 2021. Upgrading BRICKS—The context-aware semantic rule-based system for intelligent building energy and security management. *Energies* 14 (15):4541. 10.3390/en14154541.

Satchwell, A., M. A. Piette, A. Khandekar, J. Granderson, N. Frick, R. Hledik, A. Faruqui, et al. 2021. A national roadmap for grid-interactive efficient buildings. United States https://www.osti.gov/servlets/purl/1784302.

Sulzer, M., M. Wetter, R. Mutschler, and A. Sangiovanni-Vincentelli. 2023. Platform-based design for energy systems. *Applied Energy* 352 (December 2023):121955. 10.1016/j.apenergy.2023.121955.

Terkaj, W., G. F. Schneider, and P. Pauwels. 2017. Reusing domain ontologies in linked building data: The case of building automation and control. In *JOWO 2017 The Joint Ontology Workshops: Proceedings of the Joint Ontology Workshops 2017 Episode 3: The Tyrolean Autumn of Ontology*, ed. S. Borgo, O. Kutz, F. Loebe, & F. Neuhaus. Bolzano, Italy: CEUR-WS.org. http://hdl.handle.net/1854/LU-8578705.

Trenbath, K., R. Meyer, K. Woldekidan, K. Maisha, and M. Harris. 2022. Commercial building sensors and controls systems - Barriers, drivers, and costs. United States. https://www.osti.gov/servlets/purl/1880546.

Wetter, M., W. Zuo, T. Nouidui, and X. Pang. 2015. Modelica buildings library. Report #: LBNL-1002944. Berkeley, CA: Lawrence Berkeley National Laboratory (LBNL). https://escholarship.org/uc/item/14d271qz

Winstead, C., M. Bhandari, J. Nutaro, and T. Kuruganti. 2020. Peak load reduction and load shaping in HVAC and Refrigeration Systems in Commercial Buildings by Using a Novel Lightweight Dynamic Priority-Based Control Strategy. *Applied Energy* 277(November 2020):115543. 10.1016/j.apenergy.2020.115543.