Notes Paper: Enabling Building Application Development with Simulated Digital Twins

Gabe Fierro
gtfierro@mines.edu
Colorado School of Mines
National Renewable Energy Laboratory
Golden, Colorado, U.S.A.

Anand K. Prakash
akprakash@lbl.gov
Lawrence Berkeley National Laboratory
Berkeley, California, U.S.A.

David Blum
dhblum@lbl.gov
Lawrence Berkeley National Laboratory
Berkeley, California, U.S.A.

Joel Bender
jjb5@cornell.edu
Cornell University
Ithaca, New York, U.S.A.

Erik Paulson
erik.paulson@jci.com
Johnson Controls International
Glendale, Wisconsin, U.S.A.

Michael Wetter
mwetter@lbl.gov
Lawrence Berkeley National Laboratory
Berkeley, California, U.S.A.

ABSTRACT

Despite the promise of "digital twins", prototyping building applications remains difficult due to the diverse structure and composition of building control systems and lack of standard descriptions and representative test buildings. Recent advances in building simulation software allow development of control sequences in a realistic and reproducible manner, but lack industry-standard interfaces and thus impede porting developed controls to real buildings. In this paper, we discuss the design and implementation of a simulated digital twin which uses open-source technologies to present simulated buildings as if they were "brick and mortar." We demonstrate how integrating building simulations software, control network virtualization and semantic metadata into a digital twin facilitates application prototyping and enables future novel applications.

CCS CONCEPTS

• **Software and its engineering → Software architectures**.

KEYWORDS

digital twin, metadata, Brick, BACnet, BOPtest, simulation

ACM Reference Format:
Gabe Fierro, Anand K. Prakash, David Blum, Joel Bender, Erik Paulson, and Michael Wetter. 2022. Notes Paper: Enabling Building Application Development with Simulated Digital Twins. In *The 9th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BuildSys '22), November 9–10, 2022, Boston, MA, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3563357.3564060

1 INTRODUCTION

With the increased penetration of digital interfaces to buildings comes the ability to construct advanced, software-driven analytics and controls. These applications can lower energy consumption, increase building performance and enhance occupant comfort; however, prototyping these applications remains difficult.

First, buildings are characterized by extreme heterogeneity in their architecture, the composition and topology of their various subsystems, and the digital interfaces used to interact with them. This variability (Challenge #1), coupled with a lack of standardized metadata for communicating that variability (Challenge #2), means most building software must be bespoke to each deployment site. It is also time consuming for software developers to develop the deep technical understanding of each building they interact with; buildings commonly contain unusual equipment, system configurations, or digital interfaces (Challenge #3). Finally, a lack of common test buildings makes it difficult to conduct "apples to apples" comparisons of software performance across buildings (Challenge #4).

Building simulation software such as EnergyPlus [4] and those based on the Modelica language [11] address Challenge #4 by enabling creation of reproducible simulations that facilitate measurement of different building behaviors. BOPTEST [3] builds on Modelica to address Challenges #3 and #4 by providing a REST API to implement controls against publicly available building emulators and calculate relevant KPIs. The REST abstraction is useful for prototyping, but does not help in porting applications to real buildings because those building management systems do not speak REST.

Recent developments to standardize digital descriptions of buildings in the form of semantic metadata help manage the heterogeneity of buildings (Challenges #1 and #2). Metadata efforts such as Brick [2] and Project Haystack [1] lift existing ad-hoc and unstructured descriptions into machine readable data models that facilitate software development. They allow software to easily identify and access available data as well as reason about the structure and topology of building subsystems which are often hidden in simulated settings. However, such metadata solutions do not grapple with the intricacies of physics, control processes and other essential elements of the buildings operation.

These features have yet to be synthesized into a single solution for prototyping building applications. We present the prototype of such a system which permits the development and reproducible

BuildSys '22, November 9–10, 2022, Boston, MA, USA

ACM ISBN 978-1-4503-9890-9/22/11.
https://doi.org/10.1145/3563357.3564060

evaluation of building applications in a "realistic" setting: a simulated building with a virtual industry-standard control network and a descriptive semantic metadata model. We call this assembly a simulated digital twin (SDT) because it combines the interfaces and abstractions of a "brick and mortar" building with the reproducibility and visbility of a simulation. Specifically, we construct our prototype using BOPTEST [3], BACnet, and Brick [2].

2 BACKGROUND

We first give a brief overview of the three primary components of the SDT: the simulation framework, the building communication protocol, and the semantic metadata model. For each, we illustrate the essential features and characteristics that enable the SDT.

Some technology choices are not fundamental to the design: for example, a SDT could easily be implemented using OPC-UA, LonTalk or other communication protocols. Other choices (e.g., BOPTEST) provide critical abstractions that are not easily replaced.

2.1 Building Operation TESTing (BOPTEST)

The BOPTEST Framework [3] is a set of software elements and services to emulate building HVAC systems such that control algorithms can be tested and benchmarked. The building emulation models in BOPTEST are developed using Modelica [11], which enables representing realistic physical dynamics, especially HVAC system pressure-flow dynamics, as well as explicit, dynamic control logic. The models include a baseline control strategy and allow the overwriting of the supervisory and local-loop (i.e. actuator) control signals. BOPTEST exposes the "control points" of these models using a standard web interface that allows control algorithms to interact with the models as if they are physical buildings. The framework also includes calculation of standardized key performance indicators (KPI) based on data generated during the simulation.

BOPTEST has three major components. First, it contains a run time environment that provides a rapidly accessible and repeatable environment to deploy the building emulators, select test scenarios, manage control signals and measurement outputs, simulate the responses of the emulators to external control signals, and calculate KPIs. Second, it contains a public repository of deeply vetted and well-documented test cases that provide all the necessary information required to emulate a particular building and calculate the defined KPIs, including the building model and exogenous inputs for KPI calculation and forecast generation such as electricity prices or carbon emission factors. BOPTEST also contains a KPI calculator that uses the outputs of the simulation to generate relevant metrics to evaluate the performance of a test controller.

2.2 BACnet

BACnet is a digital communications protocol developed by ASHRAE for building automation and control networks. BACnet provides a sophisticated data model for enabling interoperable interactions between devices, sensors, controllers and other computational elements; for the purpose of an initial SDT, we concentrate on *devices*, *objects* and *properties*. We chose to construct the SDT with BACnet because it is a widely-deployed international standard which is familiar to many building network operators.

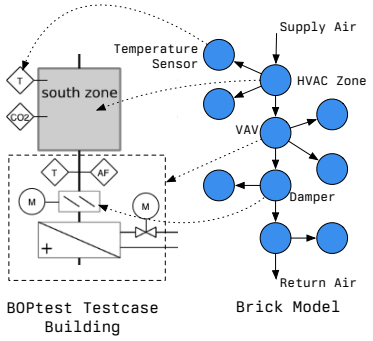

Figure 1: A Brick model captures the topology of the HVAC system. The system diagram is from BOPTEST's multizone office model based on the reference medium office building from DOE [5]

A BACnet network consists of one or more *devices*, which each host a set of addressable *objects*. BACnet defines a standard set of object types — e.g., analog input (sensor), analog output (control output), command (batch writes), and schedule — which organize the data in a BACnet network into sets of standard properties.

BACpypes[1] is an open-source project implementing the BACnet application and network layers. The modular design of BACpypes makes it possible to create a virtual BACnet network, in which the network, devices, objects and properties are implemented entirely in a single software process. This is a critical feature as it allows the use of BACnet as a familiarizing abstraction for building simulations.

2.3 Brick

Brick [2] is a graph-oriented metadata ontology for structuring information about the composition and topology of building subsystems and their related data. A *Brick model* is a digital representation of a particular building and its constituent data sources, assets, and other entities. Like other building metadata ontologies and schemas (REC [9], BOT [13], Haystack [1]), Brick abstracts away the construction-oriented details of buildings to concentrate on the contextual information important for applications. Brick has been demonstrated to decrease developer effort in implementing data-driven applications over 10s or 100s of buildings [7].

Brick's role in the SDT is to describe and thus enable discovery of the inputs and outputs of the simulation corresponding to sensors, setpoints, commands, and other points in a consistent manner, together with the topological information. The BOPTEST HTTP API describes these inputs and outputs with short names and descriptive labels that are non-standard and require knowledge of the underlying simulated building to correctly interpret, as is common for real buildings. Brick lifts the descriptions of data sources and the building into a common machine-readable representation that can be interpreted and consumed without human intervention. This representation is agnostic to whether the underlying building is real or simulated; this presents a common metadata interface that can be used to configure software in both settings. Brick also encodes

[1] https://github.com/JoelBender/bacpypes

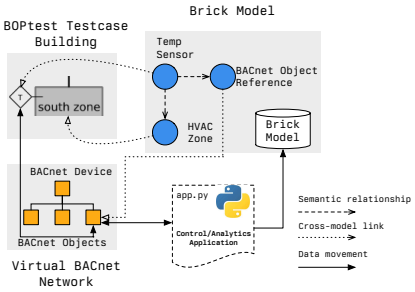

Figure 2: The movement of data and the references between the four major components of the SDT

the association between the data sources in the Brick model and other systems such as a BACnet network.

3 ARCHITECTURE

3.1 Components

The BOPTEST simulation is driven by an external client process (the BACnet shim) which proxies the inputs from the control network to BOPTEST via the HTTP API and mirrors updated values from the simulation into the control network. The BACnet shim implements a virtual BACnet network containing a single BACnet device which hosts a virtual BACnet object for each BOPTEST control input and data output (collectively: "I/O points") exposed by the HTTP API. The shim inspects the metadata associated with each I/O point to instantiate the most appropriate kind of BACnet object; for example temperature properties are exposed as Analog Input objects with the current temperature embedded in the Present Value property.

The Brick model relates the objects in the virtual BACnet network to the structure and composition of the underlying building and its systems. Figure 2 illustrates how the Brick model maintains references to the virtual BACnet objects as well as the elements in the BOPTEST model. Applications access the Brick model to discover live telemetry and control inputs for the BACnet network.

Applications developed for SDT do not know they are executing over a simulation, reducing future integration costs in bringing an application from a the SDT testing environment to production.

3.2 Data Flow

The BACnet shim is a standalone process and is responsible for the data flow in the SDT, and for periodically advancing the state of the BOPTEST simulation. When the shim starts up, it calls the BOPTEST HTTP API to initialize the simulation and obtain the initial values of the simulation state variables. The shim then creates the virtual BACnet device and objects, and sets the "present value" properties of the BACnet objects to be the initial values of the simulation. The shim then goes into an event loop, listening on the BACnet network for read and write requests from applications or controllers on the BACnet network.

BOPTEST only advances the state of the simulation when it receives an HTTP API request to do so, and that API request specifies how far into the future the simulation should advance. This simulation step is often very fast: BOPTEST can compute many minutes of simulation time in just a fraction of a second of wall-clock time. The

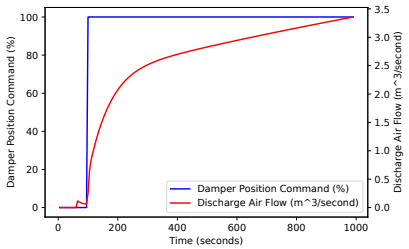

Figure 3: Result of a changing a Damper Position Command via BACnet to increase the air flow rate to a zone

shim periodically advances the simulation so that it is synchronized with wall clock time. When the BACnet shim advances the state of the simulation, it passes to BOPTEST the new values of any BACnet objects which have been changed by another BACnet device.

For example, a supervisory controller may want to change a setpoint for the "building" the SDT is emulating. To the controller, the SDT appears as if it were a real building and the controller will update a BACnet object to change the setpoint. After shim passes the updated setpoint to BOPTEST, the respective process variable will start to change in the BOPTEST simulation results, which the shim will make available in the present value property of a corresponding BACnet object. To the controller, the SDT responds the same way and at the same pace as a real building.

3.3 Bootstrapping the Brick Model

Here we explain how embedding semantic annotations in the simulation definition can assist in creating the Brick model for a BOPTEST test building; this is part of an ongoing effort to augment simulations with descriptive metadata. Software can extract these metadata annotations, providing the user with standard, machine-readable descriptions of all of the input and output points for a simulation.

We use the `modelica-json` [15] tool to convert the Modelica model embedded within the BOPTEST test case into a simplified form. The JSON output contains all the simulated elements and their corresponding semantic annotations. Leveraging the type information included in the semantic annotations and the object oriented design of the Modelica model, we extract relationships between the different elements. We borrow a technique from [6] to extract topological (`brick:feeds`), compositional (`brick:hasPart`) and spatial (`brick:hasLocation`) relationships from other types of objects and their properties. Finally, we annotate the BMS points in the Brick model with their BACnet attributes.

4 DEMONSTRATION OF USE

We have constructed a working prototype of the SDT using BOPTEST, BACnet and BRICK. To demonstrate the end-to-end usage, we implement a simple application. We start by first spinning up BOPTEST simulation for the "multizone-office-simple-air" test case and extracting a Brick model from the simulation description using the process in Section 3.3. We then instantiate the BACnet proxy, which discovers the HTTP endpoints presented by BOPTEST and creates virtual BACnet objects for each, using the inferred Brick model to determine the correct type of object. Finally, we enact our application by writing to a damper position command and observing the change in the sensed discharge air flow rate. Figure 3 shows

the change in damper position setpoint and the gradual ramping up of airflow rate that results from the control input.

5 COMPELLING APPLICATIONS

We now describe three "killer apps" for SDTs: facilitated controller development and testing, newly-enabled hardware-in-the-loop testing, and elastic co-simulation of large building populations.

Controller Development Over Standard Interfaces: The use of simulation in developing HVAC control sequences has been well-established in systems as early as HVACSIM+ [12] and SIMBAD [10], and as recent as the Modelica Buildings Library [16] and BOPTEST [3]. However, there is still a significant difference between interacting with simulations and interacting with production BMS and SCADA systems. As a result, it is non-trivial to port "lab bench" implementations of controls to function robustly over actual digital control systems.

The SDT approach has the potential to remove or at least reduce this porting cost. The virtualized building control system decouples the configuration of the simulation from the logic of the control process. Furthermore, BOPTEST's builtin KPIs eases the performance comparison of different controllers. This is reminiscent of similar work in the wireless sensor network literature [8].

Elastic Co-Simulations of Building Fleets: Existing software for simulating fleets of buildings are designed to give insight into their individual and collective energy performance under a variety of conditions. However, these solutions lack standard interfaces, which hampers their use in evaluating real-world controllers in these settings. Software like [17] simulates statistically representative samples of the U.S. building stock in parallel using high-fidelity energy simulations. Other software [14] leans on common interfaces for reinforcement learning agents. Neither of these support execution of unaltered control software on building fleets or simulating the behavior of individual buildings at sub-minute intervals. The SDT system abstracts the entire building simulation behind the familiar, standard interfaces of BACnet and Brick. We can extend the behavior of the SDT system with BOPTEST's capacity for parallel execution and existing container orchestration software to manage simulations of fleets of buildings.

Hardware-in-the-Loop Testing: Because our SDT speaks a standard BMS protocol, it can interoperate with real-world field controllers and existing BMS software. This makes it easier to implement hardware-in-the-loop simulations in reproducible and reconfigurable scenarios, without having to undertake the software integration effort to bridge an interface gap.

Finally, the SDT platform can be used for training, learning and experimentation. The combination of standard digital interfaces and high-fidelity simulations of building behavior provide a realistic "playground" that carries none of the safety concerns of working with an actual building.

6 CONCLUSION

We present the design and implementation of a simulated digital twin, an augmented building simulation with industry-standard interfaces that enables prototyping applications in a realistic and reproducible setting. Our SDT platform is open source and is available online at https://github.com/gtfierro/simulated-digital-twin.

ACKNOWLEDGEMENTS

This research was supported in part 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 DE-EE0008681.

REFERENCES

[1] 2021. *Project Haystack.* http://web.archive.org/web/20210111211811/https://project-haystack.org/

[2] Bharathan Balaji, Arka Bhattacharya, Gabriel Fierro, Jingkun Gao, Joshua Gluck, Dezhi Hong, Aslak Johansen, Jason Koh, Joern Ploennigs, Yuvraj Agarwal, Mario Berges, David Culler, Rajesh Gupta, Mikkel Baun Kjærgaard, Mani Srivastava, and Kamin Whitehouse. 2016. Brick: Towards a Unified Metadata Schema For Buildings. In *Proceedings of the 3rd ACM International Conference on Systems for Energy-Efficient Built Environments* (Palo Alto, CA, USA) *(BuildSys '16)*. Association for Computing Machinery, New York, NY, USA, 41–50. https://doi.org/10.1145/2993422.2993577

[3] David Blum, Javier Arroyo, Sen Huang, Ján Drgoňa, Filip Jorissen, Harald Taxt Walnum, Yan Chen, Kyle Benne, Draguna Vrabie, Michael 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 (2021), 586–610.

[4] Drury B Crawley, Linda K Lawrie, Frederick C Winkelmann, Walter F Buhl, Y Joe Huang, Curtis O Pedersen, Richard K Strand, Richard J Liesen, Daniel E Fisher, Michael J Witte, et al. 2001. EnergyPlus: creating a new-generation building energy simulation program. *Energy and buildings* 33, 4 (2001), 319–331.

[5] M Deru, K Field, D Studer, K Benne, B Griffith, P Torcellini, M Halverson, D Winiarski, B Liu, M Rosenberg, et al. 2010. DOE Commercial Reference Building Models for Energy Simulation–Technical Report. *National Renewable Energy Laboratory: Golden, CO, USA* (2010).

[6] Gabe Fierro, Anand Krishnan Prakash, Cory Mosiman, Marco Pritoni, Paul Raftery, Michael Wetter, and David 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* (Virtual Event, Japan) *(BuildSys '20)*. Association for Computing Machinery, New York, NY, USA, 70–79. https://doi.org/10.1145/3408308.3427627

[7] Gabe Fierro, Marco Pritoni, Moustafa AbdelBaky, Daniel Lengyel, John Leyden, Anand Prakash, Pranav Gupta, Paul Raftery, Therese Peffer, Greg Thomson, et al. 2019. Mortar: an open testbed for portable building analytics. *ACM Transactions on Sensor Networks (TOSN)* 16, 1 (2019), 1–31.

[8] Lewis Girod, Jeremy Elson, Alberto Cerpa, Thanos Stathopoulos, Nithya Ramanathan, and Deborah Estrin. 2004. EmStar: A Software Environment for Developing and Deploying Wireless Sensor Networks. In *2004 USENIX Annual Technical Conference (USENIX ATC 04)*. USENIX Association, Boston, MA. https://www.usenix.org/conference/2004-usenix-annual-technical-conference/emstar-software-environment-developing-and

[9] Karl Hammar, Erik Oskar Wallin, Per Karlberg, and David Hälleberg. 2019. The realestatecore ontology. In *International Semantic Web Conference*. Springer, 130–145.

[10] A Husaunndee, R Lahrech, H Vaezi-Nejad, and JC Visier. 1997. SIMBAD: A simulation toolbox for the design and test of HVAC control systems. In *Proceedings of the 5th international IBPSA conference*, Vol. 2. International Building Performance Simulation Association (IBPSA) Prague …, 269–276.

[11] Sven Erik Mattsson and Hilding Elmqvist. 1997. Modelica-An international effort to design the next generation modeling language. *IFAC Proceedings Volumes* 30, 4 (1997), 151–155.

[12] Cheol Park, Daniel R Clark, and George E Kelly. 1985. An overview of HVACSIM+, a dynamic building/HVAC/control systems simulation program. In *Proceedings of the 1st Annual Building Energy Simulation Conference, Seattle, WA*. 21–22.

[13] Rasmussen, Lefrançois, Schneider, and others. 2021. BOT: the building topology ontology of the W3C linked building data group. *Semant. Web* (2021).

[14] José R Vázquez-Canteli, Jérôme Kämpf, Gregor Henze, and Zoltan Nagy. 2019. CityLearn v1. 0: An OpenAI gym environment for demand response with deep reinforcement learning. In *Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*. 356–357.

[15] Michael Wetter, Jianjun Hu, Anand Krishnan Prakash, Paul Ehrlich, Gabriel Fierro, Milica Grahovac, Marco Pritoni, Lisa Rivalin, and dave Robin. 2022. Modelica-json: Transforming energy models to digitize the control delivery process. (2 2022). https://www.osti.gov/biblio/1844538

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

[17] Eric J Wilson. 2017. *ResStock-Targeting Energy and Cost Savings for US Homes.* Technical Report. National Renewable Energy Lab.(NREL), Golden, CO (United States).