Building Technologies & Urban Systems Division
Energy Technologies Area
Lawrence Berkeley National Laboratory

Towards a Stronger Foundation: Digitizing Commercial Buildings with Brick to Enable Portable Advanced Applications

Carlos Duarte Roa[1], Paul Raftery[1], Ruiji Sun[1], Lazlo Paul[2], Anand Krishnan Prakash[2], Marco Pritoni[2], Gabe Fierro[3,4], Therese Peffer[5]

[1]Center for the Built Environment, University of California
[2]Lawrence Berkeley National Laboratory
[3]Colorado School of Mines
[4]National Renewable Energy Laboratory
[5]California Institute for Energy and Environment

Energy Technologies Area
August 2022

https://doi.org/10.20357/B7ZG6R

This work was supported by the Assistant Secretary for Energy Efficiency and Renewable Energy, Building Technologies Office, of the US Department of Energy under Contract No. DE-AC02-05CH11231.

Towards a Stronger Foundation: Digitizing Commercial Buildings with Brick to Enable Portable Advanced Applications

Carlos Duarte Roa[1], Paul Raftery[1], Ruiji Sun[1], Lazlo Paul[2], Anand Krishnan[2], Marco Pritoni[2], Gabe Fierro[3,4], Therese Peffer[5]

[1]Center for the Built Environment, University of California; [2]Lawrence Berkeley National Laboratory; [3]Colorado School of Mines; [4]National Renewable Energy Laboratory; [5]California Institute for Energy and Environment

KEYWORDS

Commercial Buildings, Digital twin, Data model, Interoperability, Advanced applications, Building management system

ABSTRACT

Most large commercial buildings have digital controls for their heating, ventilation, and air-conditioning (HVAC) and lighting systems with the potential to implement advanced control strategies and data analytics. However, advanced control strategies and data analytics are rarely deployed at scale due to non-standard naming conventions and heterogenous building configurations. Semantic metadata standards, like Brick, show promise to proliferate these applications across many buildings, but they have not been widely adopted by industry due to barriers such as perceived risk and unfamiliarity with the technology. This paper describes the workflow we established and evaluated while using it to develop over ten Brick models of existing buildings. Through this process, we observed that digitizing existing commercial buildings is a cost and labor-intensive effort in which understanding the buildings' data streams is the major bottleneck. Yet, we conclude this investment is worthwhile since various use case applications such as fault detection and diagnostics, thermal comfort analysis, and HVAC control optimization can utilize the same Brick model. The paper also explores the challenges and lessons learned we encountered while creating these data models, such as: 1) difficulties in finding metadata descriptions and relationships for existing buildings; 2) handling missing concepts in the schema needed to model a building; 3) lack of guidance on how to structure the data model or how much detail to include; 4) unfamiliarity with technologies, which makes the learning curve steep for applications developers. Finally, we also describe future directions for semantic metadata research and development to make such transformative technologies more accessible to practitioners.

Introduction

Building management systems (BMS) are an increasingly common feature in buildings that control heating, ventilation, and air-conditioning (HVAC) and lighting systems. This is especially true in large commercial buildings (EIA 2022). The network of sensors provided by the BMS allows opportunities for energy consumption, cost, and control optimizations, as well as predictive maintenance, grid-interaction, visualization and reporting, and fault detection and diagnostics. Unfortunately, control strategies in the BMS are typically programmed with simple sequences that have limited potential to deliver on these opportunities, which may cost a building 5%-40% in energy savings (Ahmed et al. 2010; Lin, Kramer, and Granderson 2020). However, there has been significant growth in the adoption of energy management and information systems in the past decade, enabling the opportunities mentioned above (Kramer et al. 2020). Still, barriers exist that hinder the widespread use of advanced controls and analytics. These barriers include proprietary equipment and BMS, the unique naming convention of building assets and data points, and the inherent uniqueness of buildings and their systems, contributing to the lack of interoperability and portability of software tools at scale (Ahmed et al. 2010; Fierro 2021). For example, the lack of interoperability within a specific building prevents HVAC control sequences from using occupancy sensors installed for the lighting system that can also be used by the HVAC to enable occupancy-based controls. Another example is that various third-party vendors with new control and analytic platforms may enter the building at any stage during its occupied lifecycle to implement their latest algorithms. One of the first steps for these vendors is the discovery of existing data streams and other building resources and mapping them to their individual applications or platforms, but data point mapping might be a project in and of itself (Ploennigs et al. 2014; Wang et al. 2018). Naming conventions of data streams for new BMS deployments is currently not a standardized practice and the completeness of relevant information in the initial naming process directly influences the effort that future third-party vendors will undertake to onboard their algorithms (Butler and Veelenturf 2010; A. A. Bhattacharya et al. 2015; Wang et al. 2018). Moreover, each vendor will often need to undertake its own point mapping process since the organization or structure of the data collected by one vendor may be inadequate for another (Bergmann et al. 2020). Application onboarding, which includes the point mapping process, can cost several hundreds of dollars per point (Granderson and Lin 2016). Any repetitive effort by control vendors is inefficient and derails resources from implementing other energy saving applications or improvements in the performance of the applications.

Semantic metadata schemas offer a solution to the high costs of configuring and deploying analytics and controls by organizing building information into a single unified representation that can serve as the foundation for applications for many third-party vendors. Recent research identified 40 public semantic metadata schemas for various lifecycle stages of the building, but many of them are designed for building services, such as HVAC, lighting, domestic hot water, life and fire safety (Pritoni et al. 2021). Semantic metadata schemas for building services standardize the meaning of data communicated over a building's extensive network of sensors that make up the BMS. The high number of schemas available within building services gives a couple of insights. First, the building industry sees the value added in defining the transmitted data in a more complete and consistent way. Second, the industry is still in the early stages of determining the scope of schemas to benefit existing use cases while having the flexibility to adapt to future ones. The building industry will eventually coalesce into a smaller number of complete, extensible, expressive, and usable schemas that satisfy the needs of the majority of building stakeholders (A. Bhattacharya, Ploennigs, and Culler 2015).

Brick is an example of a metadata schema that shows promise to aid the industry in proliferating existing and future advanced control strategies and analytics (Balaji et al. 2016). It is an ontology that organizes its formal entities, or objects, into a class hierarchy using three root nodes: *Equipment*, *Point*, and *Location*. *Equipment* generally refers to a physical device that provides some service to part or all of the building. *Point* generally refers to a source of data or a control input made available over a building's control network. Points are how applications interact with the building's control network; for example, a *Sensor* in Brick does not refer to the physical device or transducer but rather the digital source of those measurements on the network.

Nameplate characteristics such as design capacities, air flows, and voltage ratings of an equipment are defined as properties of the equipment. *Location* generally refers to either physical or logical spaces that can be grouped based on a common characteristic. The definition of subclasses from these root classes becomes more specific further down the hierarchy. As illustrated in Figure 1, the *Equipment* class is a parent class with subclasses that include *HVAC*, *Lighting*, and *Electrical* while these subclasses are parent classes to more specific subclasses such as *Air Handling Unit* and *Chiller* for *HVAC*.

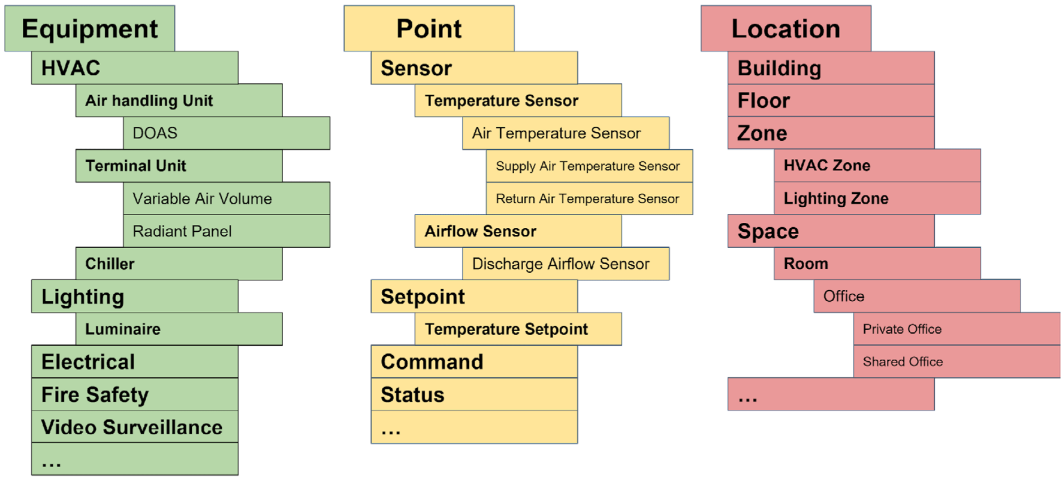

Figure 1: Excerpt of the Brick schema class hierarchy with three root nodes: *Equipment*, *Point*, and *Location*. Each box represents an example of a Brick class entity with a formal definition and set of properties. Two entities, or classes, can be connected through a relationship (Fierro 2021).

Eventually, we can identify a class in the hierarchy that is specific enough to assign to a building's asset and associate properties to it. Then we can take two classes and associate them by assigning a formal Brick relationship. Brick relationships add the functional and spatial relationships for equipment, points, and building spaces or thermal zones. Figure 2 shows (top) a schematic and (bottom) the corresponding Brick data model of a multi-zone variable air volume (VAV) air-handling unit (AHU) in a typical built-up HVAC system. The figure shows how we used the *hasPart* relationship to indicate the main and subcomponents of a built-up AHU unit. Next, we used the *feeds* relationship to indicate the order of the subcomponents as air transverses through the system. Many analytics need information on the order of components to do the proper analysis, such as the order of the heating and cooling coils or if it is a blow- or draw- fan design system. Lastly, the *hasPoint* relationship indicates the data streams associated with each component.

The Brick ontology[1] is currently under active development; thus, new features are regularly added. Nevertheless, the latest iteration shows great promise to capture the requirements of three fundamental use cases identified by Pritoni et al. (2021). The three use cases are 1) building energy audits, 2) automated fault detection and diagnostics, and 3) optimal control of building services. Brick has been gaining traction among academia and industry stakeholders (Brick Consortium 2022). However, there is a lack of documentation that illustrates the workflow to create Brick data models for buildings. Therefore, we present the process we undertook to develop Brick data models for more than ten existing buildings using the currently

[1] brickschema.org/

available tools. There were no specific selection criteria for the development of data models for these ten plus buildings except that we had data available or access to the building to develop the Brick models. Then, we present lessons learned and gathered from undertaking these efforts and existing challenges that need to be addressed in future iterations of the Brick ontology. We will finish by discussing future directions we envision that will help increase the widespread adoption of building schemas, focusing on the Brick ontology.

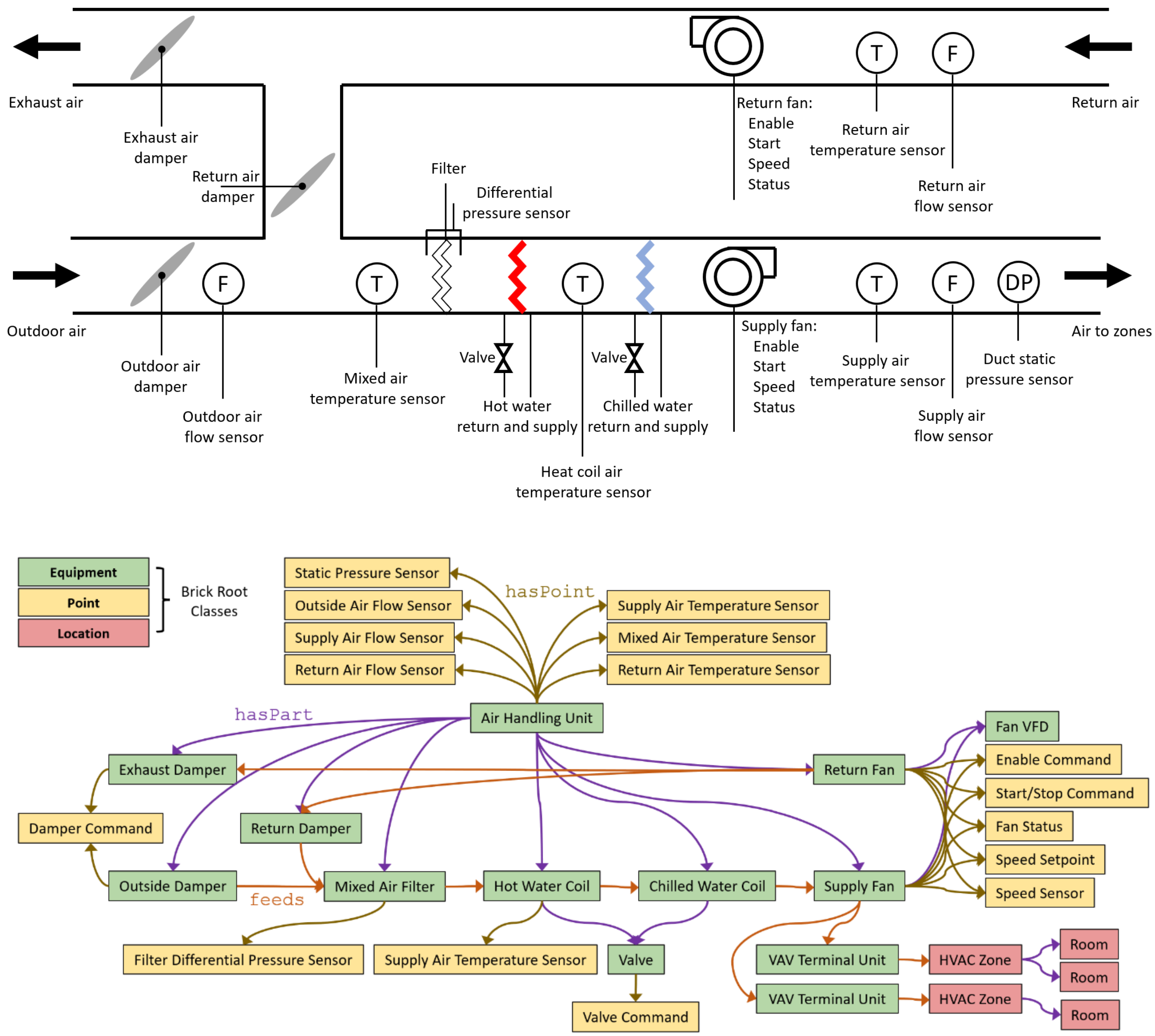

Figure 2: (Top) schematic of a multi-zone variable air volume (VAV) air handling unit (AHU).(Bottom) the corresponding Brick data model. The boxes in this figure represent *instances* of the class name given by the text inside.

Brick Model

Development Workflow

This section presents the workflow we used to develop Brick data models for over ten existing buildings. The buildings in the list include buildings with conventional air-based HVAC systems and hydronic-based systems. It also includes various BMS. Figure 3 shows the five-step workflow we employed to develop the buildings' Brick models: 1) collect siloed metadata, 2) organize metadata, 3) transform metadata, 4) apply inference and reasoning to Brick model, and 5) validate Brick model.

The first step is to collect metadata that is siloed in various document sources and formats. Since our interest is in researching the potential of Brick models to facilitate analytics and controls, we centered our process around the time series data available through the BMS. Thus, we initiate the workflow process by scanning the BMS communication network to retrieve a list of points available. The scan feature may be available through the BMS interface. In other cases where it is not, we performed a BACnet network scan and point list retrieval using open-source network discovery utilities and other related software such as Nmap and BACpypes (Nmap 2022; Bender 2018). At a minimum, we obtain point names and additional information that may include point type, data units, short description, and BACnet object properties. Point names can contain a significant amount of information (Butler and Veelenturf, 2010). Therefore, we used our domain expertise to extract the embedded information. An example of the type of embedded metadata is illustrated in Figure 4. We performed this extraction of information manually or semi-automatically if the naming conventions were consistent. In most cases, we used a mix of the two methods since it was rare that consistency was found among all point names within one building. For example, all air temperature points followed one pattern while air flow points followed another, or the delimiter was exchanged from an underscore to a hyphen. Then we referred to mechanical and architectural drawings when available to complete the functional and spatial relationships among the various sensors and equipment.

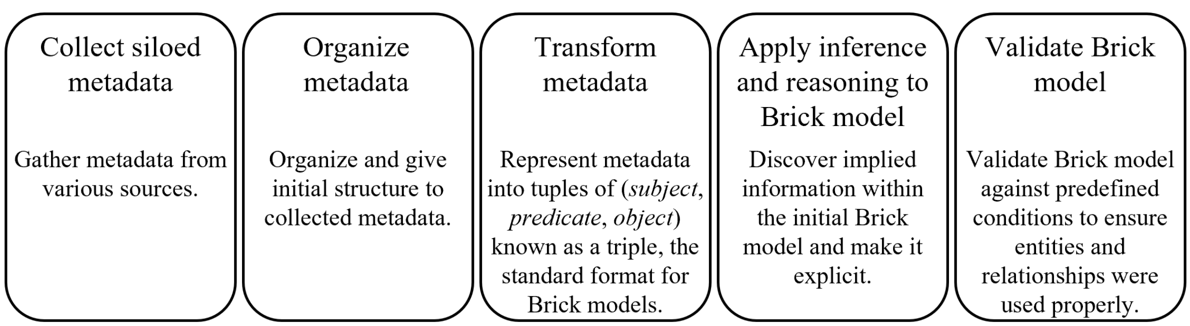

Figure 3: The five-step workflow process to develop a Brick model for a building.

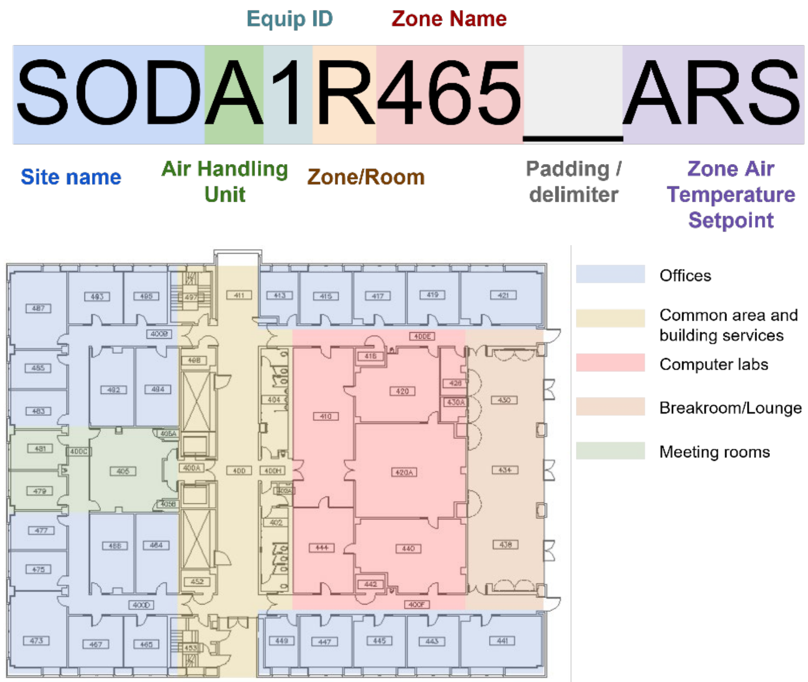

Figure 4: Example of metadata embedded in a point name (Fierro 2021) and metadata embedded in a floor plan drawing. These metadata may tell us site name, equipment information, sensor type and location, room/space type, number, and location, and occupancy.

The second step in the Brick building workflow is to organize and implement an initial structure to the collected metadata. In this step, we found spreadsheets to work well. We initiate the process by organizing the point names in one column and the Brick classes in another. It may also be helpful to group points by plant, distribution, and zone level. In this way, we can add columns with information pertinent to the specific group. For instance, we can group zone terminal equipment that can be part of the distribution level. For this group, we can define columns that indicate which AHU serves each terminal equipment, the available sensors on each terminal unit, the zone(s) or room(s) that each terminal unit serves, type of zone (such as office, conference, breakroom, etc), the area of the zone, the floor number that the equipment/zone is on, and any design parameters of the terminal units, (e.g., minimum and maximum air flow rate).

The third step is to transform the metadata into machine-readable text files capable of accepting programmatic queries. It is easier to use currently available tools on the semi-structured metadata we implemented in step two to transform it into a Brick data model. A Brick data model represents the information we gathered by identifying and characterizing the "things" inside a building called *entities* and the relationships between them; this is a graph structure. The RDF[2] data model, which Brick is built on, describes graphs as a set of statements called *triples*. *Triples* have three parts: a *subject* node or *entity*, an object node or *entity*, and an edge which describes the relationship from the *subject* to the *object*. The relationship is also known as the *predicate* and has a direction. In Figure 2, the various colored nodes represent instances of classes defined by Brick. The set of triples describing a building makes up the Brick model for that building (Balaji et al. 2016); this is a machine-readable representation which can be queried by applications. One example of a triple from Figure 2 is (Air_Handling_Unit (*subject*) hasPart

[2] Resource Description Framework (www.w3.org/RDF/)

(*predicate*) Hot_Water_Coil (*object*)). We used the open-source tool *brick-builder* available on Github to transform our spreadsheet-organized metadata to Brick data models (Fierro 2022).

The fourth step is to apply inference to a Brick model to discover implied information and make it explicit. Recall that Brick is an ontology, a formal logic-based representation of the knowledge in some domain. The Brick ontology leverages this representation to define rules and axioms that outline what additional information can be derived from the statements in a building's Brick model. This includes "inherited" information such as the substance and quantity measured by a sensor and what equipment is further upstream or downstream of some entity (Fierro et al. 2020a). The process of applying inference will add this derived information to the building model so users of the model can access it. The inference process alleviates the process of producing a Brick model because it allows helpful annotations and statements to be produced automatically rather than manually. Brick defines inference rules using modern web technologies to formalize knowledge representation such as the OWL[3] 2 RL and SHACL[4] W3C[5] standards (Motik et al. 2012; Knublauch and Kontokostas 2017). Adopting these standards lets users choose from a wide array of open-source and commercial products to perform the inference on Brick models.

The fifth step is to validate a Brick model against predefined conditions and constraints to ensure that entities and relationships were used properly. These are defined using the SHACL W3C standard. Constraints enforce the logical and semantic consistency of the model. Logical constraints ensure that the Brick ontology is being used correctly. For example, a sensor class entity cannot also be a location, points must be related to equipment via the "*isPointOf*" relationship rather than "*isPartOf*" or other relationships. Semantic constraints encode domain knowledge in a way that can be automatically verified: buildings contain rooms and not the other way around; sensors should have associated units that are appropriate to what the sensor is measuring. Constraints can also be used to express invariants that must hold true for a particular kind of equipment or a particular building. For example, certain models of VAV terminal units might be required to have a certain array of associated points, or a specific building might have 20 HVAC zones. These constraints can ensure that the building model has the expected metadata.

Use cases and applications

Brick data models offer great flexibility for various use cases. We have developed applications that span fault detection and diagnostics, thermal comfort analysis, and HVAC control optimization, energy predictions, and dashboards. We used the Brick data models we created, in addition to Brick models created by others, as part of the Mortar[6] database (Fierro et al. 2020a) to run various applications. The applications include one that retrieves relevant VAV terminal unit data streams to categorize if the hot water reheat valve in the device lets hot water through when it is supposed to be closed (passing valves) (Duarte Roa et al. 2022). This application analyzed 1,335 VAV terminal units in 20 buildings. In another example, we retrieved temperature data streams from 1,953 zones in 25 buildings to assess their ability to maintain long-term thermal comfort (Sun, Duarte Roa, and Raftery 2022). Finally, we are in the process of

[3] Web Ontology Language (www.w3.org/TR/owl2-profiles/)
[4] Shapes Constraint Language (www.w3.org/TR/shacl/)
[5] World Wide Web Consortium (www.w3.org/)
[6] mortardata.org/

demonstrating real-time control of the hot water plant in an existing building. We embedded BACnet object information within the building's Brick data model to retrieve real-time measurements and information from the BMS and sent new commands to it to adjust the hot water supply temperature. In theory, this application can be ported over to another building with a hot water plant and Brick data model with little or no modification, similar to the first two applications where multiple buildings were analyzed with only one instance of the application.

Lessons learned

Digitizing existing commercial buildings is currently a labor-intensive effort. We observed that digitizing buildings with Brick by using the five-step workflow process outlined in Figure 3 is no exception. Gathering building information and understanding data streams from existing buildings are major bottlenecks; these correspond to the first two steps of the workflow process we presented. Accurate, complete, and readable building drawings are harder to come by for older buildings. Older buildings also have the increased likelihood of multiple retrofits which means multiple independent documents detailing those retrofits which need to be tracked down and reconciled with all other building data. In many cases, the documents are also a mix of paper and digital formats. Efforts to digitize a full or partial building with Brick as well as another schema is redundant and not good use of time. If third party vendors adapted their platforms or algorithms to utilize a single semantic metadata schema, this would save time and money. Even though developing a Brick data model for an existing building is a significant effort, we believe the investment is worthwhile. A Brick data model offers an interoperable digital representation of a building with the capacity to store information and be continually updated to assist a variety of use cases within the scope of building services provided by different vendors. Recent work (Fierro et al. 2020b) proposes algorithms for incrementally editing a Brick model over time as different digital representations evolve. The development costs start being recouped after the Brick data model enables the second vendor to reduce onboarding process time and costs, but further studies are needed to calculate a data model's return on investment. The use of open-source technologies within the design of the Brick ontology ensures that various stakeholders with many software tools that support these technologies can take advantage of a Brick data model.

Challenges encountered

Though there are clear advantages in using Brick to merge siloed metadata into one machine-readable and interoperable format, we encountered some challenges throughout the workflow. The four main challenges are the following:

1. Difficulties in finding metadata descriptions and relationships for existing buildings.
2. Missing concepts in the Brick ontology.
3. Lack of guidance on how to structure the Brick data model or how much detail to include.
4. Unfamiliarity with technologies, which makes the learning curve steep for applications developers.

The Brick ontology's design supports use cases in building services, thus BMS point names are a good place to start the metadata extraction process. We took advantage of some naming conventions, as illustrated in Figure 4, but those are not always present , and the convention was not maintained throughout a single building let alone across multiple buildings. We attempted to develop simple rule-based programs to semi-automate the metadata extraction, but it proved difficult. We found ourselves implementing many logical conditions to address the inconsistencies, such as an underscore switched for a hyphen, the order of the expected information was reversed, or the abbreviations used for a single piece of information varied. After extracting the available information from point names, we continued with building drawings. Unfortunately, though, information in drawings is sometimes not well recorded. For example, it may be difficult to tell if the drawings supplied by the building manager are the as-built drawings of the building or if they include any retrofit information. On one occasion, we carried out a walkthrough of the building to identify information about the return air streams of the HVAC system to properly document it in the Brick data model. These examples stress the importance of keeping metadata updated and why Brick data models may be useful as a repository of information that can be easily updated with clear versioning.

Buildings are highly unique with customized built-up systems that reflect their designed function, performance, and environmental conditions they must withstand. The Brick ontology may not have all the concepts needed to model a building. In particular, the initial concepts for Brick were developed for a typical all-air system, which made it challenging to build Brick models for two of our buildings whose HVAC systems are mainly composed of hydronic water loops. The two buildings incorporated radiant heating and cooling systems in their HVAC systems. Therefore, the zone equipment is no longer a VAV terminal unit where the air is the controlled temperature but a concrete slab or metal panel whose core or surface temperature is controlled using various types of water valves (e.g., two-way valves or modulating valves). One of the buildings also contained reversible water-to-air heat pumps that were not included in the Brick ontology version we used to create the data model for that building. We also need a way to represent the data stream that informs if the heat pump is extracting (heating mode) or adding (cooling mode) energy from/in the water loop. The modeling gets more complex if modeling the lower-level components of a heat pump system is essential. There is no support for differentiating between equipment with dynamic functions such as a heat exchanger that can act as both a condenser and an evaporator in a heat pump system. Similar issues may arise when encountering buildings with variable refrigerant flow (VRF) HVAC systems. The Brick development team is currently developing heat pump and VRF equipment entities and related points for future release that address some of the issues mentioned above.

The Brick ontology offers tremendous flexibility to develop Brick data models. However, there is a lack of guidance on how to structure the data model or how much detail to include. One option is to be very detailed, as we attempted to illustrate in Figure 2. Still, we believe that adding some Brick entities such as *Entering* and *Leaving Air Temperature Sensor*, along with similar entities for other fluids and sensor types, would allow for a better representation of a detailed Brick data model. The addition of these suggested entities would have allowed us to attach a *Leaving Air Temperature Sensor* to the heating coil and the supply fan. This explicitly identifies the state of a medium right before or after undergoing some processing through a piece of equipment. Similarly, we can increase the detail and be more explicit in the model by attaching *Entering* or *Leaving Air Flow Sensor*s to the components of the AHU instead of connecting them to the high-level equipment (AHU).

The other option is to reduce the complexity of the data model and keep points attached to the high-level equipment and avoid the detailed *feeds* relationships between the internal components of the AHU as depicted in Figure 2. Instead, we would directly connect the *feeds* from the AHU to the VAV terminal units. Much of the information will be implied in this type of structure, and some details may not be modeled. For example, there are no *Air Temperature Sensor* subclasses to model extra sensors that might be in the AHU after internal components such as the air temperature sensor after the heating coil as depicted in Figure 2. This loss of information for interim processes might be okay for the modeler and the Brick data model purposes. Each structure has its advantages and disadvantages. A less detailed model takes less time to build, and extracting information may be straightforward. However, a simpler model might introduce ambiguity in the capabilities of the building.

On the other hand, a detailed model can take full advantage of sophisticated current and future algorithms that depend on the interactions and processes of the low-level equipment. Furthermore, since it is presumably more complete, more application developers will have the ability to bootstrap their platforms and applications to a Brick data model. For example, our passing valve application mentioned above can also apply to hot and chilled water valves found in the AHU, but we could not analyze them because the majority of developers of the Mortar database Brick models we used in the analysis chose to model the AHU at a high level. Thus, we did not have the internal air temperature sensors before and after the hot and chilled water coils to do a proper analysis.

Finally, the Semantic Web technologies used in designing the Brick ontology and its supporting tools are generally unfamiliar to the building industry. Although, many companies outside the building industry, (e.g., technology, search, and social media companies), have used these technologies to standardize the exchange of information on the modern-day internet. Nevertheless, there are perceived risks and the time and effort to learn, adopt, and embrace these technologies can be steep for the building industry. New Brick users need knowledge in these Semantic Web technologies, in addition to their domain expertise, to develop complete and beneficial Brick data models. Furthermore, all Brick modelers need to foresee the Brick model's use cases and think about how application developers might create programmatic queries to extract information contained within it and avoid any ambiguity. As discussed above, there are multiple approaches to model a building with Brick, making it difficult to construct a programmatic query that encompasses the various modeling approaches which result in fundamentally different graph structures. For example, one model may have an *Air Temperature Sensor* related to the room while another model has a *Zone Air Temperature Sensor* related to a VAV terminal unit which then *feeds* to an HVAC zone which in turn *hasPart* part a room. A single programmatic query to extract the room's air temperature will return successful results for one but not the other. This issue can sometimes be addressed with generalized queries (Bennani et al. 2021) but in other times, too generic queries can derail from the analysis intent of the application, such as analyzing thermal comfort by using the VAV terminal unit's supply air temperature instead of the zone's air temperature. Thus, having a consistent structure expected within Brick data models is paramount to creating effective portable queries.

Recommendations for the future

The Brick ontology is showing great promise to proliferate applications that are written once and run across multiple buildings. We have some suggestions to further improve upon this goal. The Brick project team needs to open new pathways to increase the involvement of the building industry. This could be done by increasing the available documentation for Brick to show the benefits and advantages to digitizing buildings with Brick. This may involve creating Brick modeling examples of simple and/or typical systems and examples of programmatic queries to retrieve the available information within them. The examples must establish a preferred modeling approach and final structure of a Brick graph so the community of Brick modelers can follow suit. The collaboration with the building industry community will also help identify modeling gaps and missing concepts and entities. In these early stages of Brick, we envision the use cases and the development of the Brick ontology to co-evolve and this is most likely a good approach since it is directly taking into account the user of the Brick ontology.

Conclusion

This paper documents the five-step workflow process we used to develop Brick data models for more than ten existing buildings. The five steps are to 1) collect siloed metadata from various sources and formats; 2) organize the collected metadata and implement a semi-structured format; 3) transform the semi-structured format into a Brick data model which is a machine-readable and interoperable text file; 4) apply inference and reasoning to the initial Brick model to discover implied information and make it explicit; and 5) validate the newly created Brick data model to ensure that entities and relationships were correctly used. The five-step process, along with other point mapping techniques performed by third-party vendors, is time-consuming, and therefore, the process should only be performed once but allow multiple building stakeholders to make use of the knowledge gathered from the process.

The Brick data model is moving in the right direction to serve as the singular repository containing the knowledge gathered to serve multiple needs. A Brick data model can undergo continual updates to adapt as different sensor measurements and other information are needed for incoming third-party vendors with new algorithms or platforms.

However, we encountered challenges that may hinder the widespread adoption of the Brick ontology. Buildings are highly unique and Brick is well equipped to model the various systems pertaining to building services (e.g., HVAC, lighting, domestic hot water, and life and fire safety), but not all the concepts needed to model the variation found in buildings currently exist within the Brick ontology. Brick allows a building to be modeled in many different ways but there is a lack of guidance on a preferred approach to structure the Brick data model and how much to include.

The Brick project must continue to be involved with the community of Brick modelers and Brick application developers to understand and receive feedback on the missing gaps and concepts to increase the coverage of the Brick ontology. There is a need for increased documentation and examples on how to model common systems with the preferred modeling approach. This documentation and examples would also include how data should be extracted and help the building industry get familiar with Web Semantic technologies. Documentation and examples could also show the possibilities of using these technologies within the building industry and reduce the perceived risks with learning, adopting, and embracing these technologies.

References

Ahmed, A., J. Ploennigs, K. Menzel, and B. Cahill. 2010. “Multi-Dimensional Building Performance Data Management for Continuous Commissioning.” Advanced Engineering Informatics 24 (4): 466–75. doi.org/10.1016/j.aei.2010.06.007.

Balaji, B., A. Bhattacharya, G. Fierro, J. Gao, J. Gluck, D. Hong, A. Johansen, et al. 2016. “Brick: Towards a Unified Metadata Schema For Buildings.” In Proceedings of the 3rd ACM International Conference on Systems for Energy-Efficient Built Environments, 41–50. Palo Alto CA USA: ACM. doi.org/10.1145/2993422.2993577.

Bender, J. 2018. “BACpypes Documentation.” media.readthedocs.org/pdf/bacpypes/latest/bacpypes.pdf.

Bennani, I. L., A. K. Prakash, M. Zafiris, L. Paul, C. D. Duarte Roa, P. Raftery, M. Pritoni, and G. Fierro. 2021. “Query Relaxation for Portable Brick-Based Applications,” 10.

Bergmann, H., C. Mosiman, A. Saha, S. Haile, and W. Livingood. 2020. “Semantic Interoperability to Enable Smart, Grid-Interactive Efficient Buildings.” In 2020 Summer Study on Energy Efficiency in Buildings, 18. ACEEE. www.aceee.org/2020-buildings-summer-study-virtual.

Bhattacharya, A. A., D. Hong, D. Culler, J. Ortiz, K. Whitehouse, and E. Wu. 2015. “Automated Metadata Construction to Support Portable Building Applications.” In Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments, 3–12. BuildSys ’15. New York, NY, USA: Association for Computing Machinery. doi.org/10.1145/2821650.2821667.

Bhattacharya, A., J. Ploennigs, and D. Culler. 2015. “Short Paper: Analyzing Metadata Schemas for Buildings: The Good, the Bad, and the Ugly.” In Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments, 33–34. BuildSys ’15. New York, NY, USA: Association for Computing Machinery. doi.org/10.1145/2821650.2821669.

Brick Consortium. 2022. “Brick Consortium - BrickSchema.” 2022. brickschema.org/consortium/.

Butler, J. F., and R. Veelenturf. 2010. “Point Naming Standards.” ASHRAE Journal. http://128.84.159.37/Bibliography/BACnet-Today-10/Butler_2010.pdf.

Duarte Roa, C., P. Raftery, R. Singla, and M. Pritoni. 2022. “Detecting Passing Valves at Scale Across Different Buildings and Systems: A Brick Enabled and Mortar Tested Application.” In *ACEEE Summer Study on Energy Efficiency in Buildings*. Pacific Grove, CA: ACEEE.

EIA. 2022. “Commercial Buildings Energy Consumption Survey (CBECS) Data.” Energy Information Administration, U.S. Department of Energy. www.eia.gov/consumption/commercial/data/2018/index.php?view=characteristics.

Fierro, G., M. Pritoni, M. Abdelbaky, D. Lengyel, J. Leyden, A. Prakash, P. Gupta, et al. 2020. “Mortar: An Open Testbed for Portable Building Analytics.” ACM Transactions on Sensor Networks 16 (1): 1–31. doi.org/10.1145/3366375.

Fierro, G. 2021. “Self-Adapting Software for Cyberphysical Systems.” Doctoral Dissertation, Berkeley, CA: University of California, Berkeley. www2.eecs.berkeley.edu/Pubs/TechRpts/2021/EECS-2021-159.pdf.

———. 2022. CSV-to-Brick. Python. github.com/gtfierro/brick-builder.

Fierro, G., J. Koh, S. Nagare, X. Zang, Y. Agarwal, R. K. Gupta, and D. E. Culler. 2020. “Formalizing Tag-Based Metadata With the Brick Ontology.” Frontiers in Built Environment 6 (September): 558034. doi.org/10.3389/fbuil.2020.558034.

Fierro, G., A.K. 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. doi.org/10.1145/3408308.3427627.

Granderson, J., and G. Lin. 2016. “Building Energy Information Systems: Synthesis of Costs, Savings, and Best-Practice Uses.” Energy Efficiency 9 (6): 1369–84. doi.org/10.1007/s12053-016-9428-9.

Knublauch, H., and D. Kontokostas. 2017. “Shapes Constraint Language (SHACL).” W3.Org. 2017. www.w3.org/TR/shacl/.

Kramer, H., G. Lin, C. Curtin, E. Crowe, and J. Granderson. 2020. “Proving the Business Case for Building Analytics.” None, 1695762, ark:/13030/qt5m66941j. Berkeley, CA: Lawrence Berkeley National Laboratory. doi.org/10.2172/1695762.

Lin, G., H. Kramer, and J. Granderson. 2020. “Building Fault Detection and Diagnostics: Achieved Savings, and Methods to Evaluate Algorithm Performance.” Building and Environment 168 (January): 106505. doi.org/10.1016/j.buildenv.2019.106505.

Motik, B., B. Cuenca Grau, I. Horrocks, Z. Wu, A. Fokoue, and C. Lutz. 2012. “OWL 2 Web Ontology Language Profiles (Second Edition).” W3.Org. 2012. www.w3.org/TR/owl2-profiles/.

Nmap. 2022. Nmap: The Network Mapper - Free Security Scanner. Nmap Project. nmap.org/.

Ploennigs, J., B. Gorman, N. Brady, and A. Schumann. 2014. “BEAD - Building Energy Asset Discovery Tool for Automating Smart Building Analytics: Demo Abstract.” In Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings, 194–95. BuildSys ’14. New York, NY, USA: Association for Computing Machinery. doi.org/10.1145/2674061.2675030.

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. doi.org/10.3390/en14072024.

Sun, R., C. Duarte Roa, and P. Raftery. 2022. “Enabling Portable and Reproducible Long-Term Thermal Comfort Evaluation with Brick Schema and Mortar Testbed.” In 2022 ASHRAE Annual Conference. Toronto, ON, Canada.

Wang, W., M. R. Brambley, W. Kim, S. Somasundaram, and A. J. Stevens. 2018. “Automated Point Mapping for Building Control Systems: Recent Advances and Future Research Needs.” Automation in Construction 85 (January): 107–23. doi.org/10.1016/j.autcon.2017.09.013.