Metadata Schema Generation for Data-driven Smart Buildings

Lasitha Chamari[1], Joep van der Weijden[2], Lolke Boonstra[2], Stefan Hoekstra[2], Ekaterina Petrova[1] and Pieter Pauwels[1]

[1]*Eindhoven University of Technology, The Netherlands*
[2]*Delft University of Technology, The Netherlands*

Abstract

A smart building is a combination of advanced information systems originating from different domains. Domains such as design and construction, maintenance, energy management, automation & control have complex yet important relationships, and ensuring their connectivity is crucial for building operations. Semantic web technologies can be used to model and link these domains and their relationships using domain ontologies. To that end, there are a number of smart building ontologies that are available in each domain. However, the process of generating a metadata schema by using those ontologies for a given building is not investigated adequately. Further, such tools that generate those metadata schemas are rare. Therefore, this study presents a semi-automatic metadata schema generator using an ontology database and a text search engine. The proposed approach is applied to a campus building. Building Automation System metadata was used in the metadata schema generator. Finally, this study shows how the generated metadata scheme can be used to efficiently query and visualize time-series data for developing data-driven smart building applications.

Keywords

Metadata Schemes, Building Management Systems, Linked Data, Semantic Web, Brick ontology

1. Introduction

Centralized Building Automation Systems (BAS) are widely used in large buildings to regulate indoor climate, ventilation, lighting schedules, and other essential building functions. These systems are integrated with various sensors and actuators, such as Air Handler Units (AHUs), boilers, exhaust fan systems, chillers, fire safety systems, and more. The BAS collects data from these sensors and provides built-in functions for data collection, trend analysis, and visualization[1]. BAS vendors like Johnson Controls, Siemens, or Priva often provide these proprietary functionalities. As a result, integration with other data analytics tooling becomes a challenge. Another problem is using vendor-specific conventions and data models for describing sensor points[2]. This leads to customized translators and data models, making it harder to use available data for any party interested in developing data-driven building controllers[2]. The naming of points is often called a "tag" and these tags encapsulate data about the type of point, its relationship with other equipment and its location, which are essential in developing applications. These complex tags are created to be understood by field engineers and, therefore not intended to make them machine-readable[3]. Therefore, extensive research has been conducted to see how these BAS data points can be recognizable and unified across various vendors and buildings and how to make machine-readable.

Proceedings LDAC2023 – 11th Linked Data in Architecture and Construction, June 15–16, 2023, Matera, Italy
* Corresponding author
EMAIL : l.c.rathnayaka.mudiyanselage@tue.nl (Lasitha Chamari)
ORCID : 0000-0003-2726-1393 (Lasitha Chamari) 0000-0002-8651-0671 (Ekaterina Petrova) 0000-0001-8020-4609 (Pieter Pauwels)

Recent advancements in ontologies in the BAS domain, such as the Brick ontology[4] and standards such as Haystack[2], have contributed positively to standardizing the semantics of various metadata associated with BAS. These standards help establish semantic models that describe physical and/or logical points in the BAS, such as equipment, their locations, associated points, and the type of those points, whether they are an alarm, setpoint, or command, for example. In essence, a metadata schema for a building is a Resource Description Framework (RDF)[3] graph that uses classifications available in BAS domain ontologies such as the Brick ontology, and this type of metadata schema is the focus of this research. These metadata schemes facilitate deploying analytics applications without making previous knowledge about the building a prerequisite[3]. Despite advancements in the field, creating metadata schemas for BAS remains a challenge. The process of transforming the existing legacy identifiers in BAS to ontological models is non-trivial. This paper presents a five-step methodology that can be employed to generate metadata schemas for a given BAS based on the Brick ontology.

1. Set the objective clearly, why the metadata scheme is needed and how it will be used.
2. Sort the relevant metadata required for schema generation and classify and group metadata into understandable and logical categories.
3. Use a text search engine populated with the classes and definitions extracted from the chosen ontology.
4. Run an automatic text search and fine-tune the results with expert human input.
5. Integrate the schema with the application to achieve the objectives set in step 1.

This methodology is demonstrated through a case study in a Dutch building, a living lab at the Delft University of Technology. This living lab is a testing ground for data-driven control methods developed in the Brains for Buildings[4] project, including Fault Detection and Diagnosis (FDD) and other data-driven applications.

2. Related Work

This section discusses methods for standardizing the concepts around BAS systems and methods for generating a metadata schema based on those standards.

Ontologies and tagging systems have been used prominently for standardizing BAS metadata. Haystack is a tag-based method that allows describing BAS points using a library of predefined vocabulary. Because of the absence of formal rules on how to use these tags, the buildings that use Haystack tags usually tend to have an ad-hoc collection of tags. Charpenay et al. [5] also identified Haystack's textual document approach to describe semantics and its limited accessibility via web standards, such as a RESTful Application Programming Interface (API), as barriers to implementing Haystack on a large scale. To that end, they proposed the Haystack Tagging Ontology (HTO), which supports semantic web technologies (RDF, OWL, SPARQL) to address this gap in Haystack [14]. The Brick ontology uses classes and subclasses to describe points hierarchically. The main Classes are Collection, Equipment, Location, Measurable, and Point.

Further, a Relationship Class defines relationships between the subclasses such as *hasLocation, hasPoint, isFedBy,* etc. The Semantic Sensor Networks (SSN)[5] [6] and Sensor, Observation, Sample, and Actuator (SOSA) [7] ontologies describe the sensors and actuator domains in general (not specific to BAS) in great detail. Terkaj et al. [8] elaborated on how multiple ontologies (BOT, SSN, SOSA, etc.) can be reused by integrating them to describe BAS. Their proposed BACS ontology also tried to describe control logic in addition to BAS points. The RealEstateCore[6] (REC) ontology [9] is another addition to the smart building domain, which aims at integrating the concepts described by the above ontologies according to real estate needs.

[2] https://project-haystack.org/
[3] https://www.w3.org/TR/rdf11-concepts/#section-rdf-graph
[4] https://brains4buildings.org/
[5] https://www.w3.org/TR/vocab-ssn/
[6] https://www.realestatecore.io/

As mentioned, ontologies define standard semantics for the BAS and beyond to great depth. The next challenge is using them to translate legacy BAS identifiers available in buildings usually made without standardized semantics. Previous studies have employed three methods to derive standard semantics of BAS points. They are i) tag-based [3] [10], ii) time-series data based [11] [12] [13] and iii) a combination of both using tags and time-series data. They often aimed at inferring mostly the type of point [14]. Other inferences include the location and relationships among points.

Tag-based methods used identification techniques such as regular expressions [3], syntactic clustering of text identifiers[3], and linguistics. Bhattacharya [3] showcased the use of regular expressions to identify typical patterns in metadata descriptors with human expertise as input. Their technique transforms metadata into Haystack standard. In [5], a system has been proposed for the automated classification, naming, and management of sensors through active learning from sensor metadata. This method was based on matching unique point descriptions. Their approach focuses on pool-based active learning algorithms, which leverage scenarios with a small set of labelled data and a vast pool of unlabeled data and showed 28% fewer training examples when compared to a regular expression-based method. However, the algorithm's performance under different equipment, vendors, and facilities management sets is not investigated.

The above text-based methods are effective when the naming convention is in place. When a proper naming convention is unavailable, time-series data can infer the metadata of points. Fürst [15] developed a crowdsourcing approach for maintaining BAS metadata. The authors propose a web application that allows users to suggest and vote on the mappings between BAS data points and a common ontology. Their solution is based on their hypothesis that much of the physical state of a building can be observed by humans and that building metadata maintenance should be based on human input. However, this is heavily dependent on end-users' participation, who may not be motivated or have the necessary expertise to accurately match and label sensor points. Also, the method may not be effective in environments with complex or specialized equipment, as users may lack the necessary domain expertise to correctly match and label sensor points. This method also requires system control, which may only be feasible for some buildings and during a specific period. Another method used is perturbing the operation of equipment such as AHUs [16] to discover the functional relationships between AHUs and VAV boxes and studying the responses in VAVs. This approach is suitable when the point names are unavailable and the relations are not encoded in the point naming convention.

The study by Gao [14] involved implementing and evaluating six distinct metadata inference techniques based on time-series data analysis. These approaches were tested on sensors from 614 AHUs installed in 35 building sites, encompassing over 400 buildings in the United States of America. The study focused on inferring 12 types of sensors and actuators in AHUs necessary for a rule-based FDD application and subsequently mapped to the Brick ontology. However, since the research was conducted solely on AHUs, the findings may not be generalizable to other equipment types. It is also important to note that the time-series data-based methods employed in the study necessitated the availability of historical sensor data collected from the buildings, which may not be readily accessible in all cases. Another method of generating inferences by time-series data is demonstrated by [17] by relying on the fact that sensors and equipment in the same physical environment are affected by the same real-world events, thereby making correlated changes in the time-series data. All the approaches require some level of human input, and no fully automated method is currently available for metadata standardization.

After reviewing the existing literature on metadata mapping, we have identified several gaps that need to be addressed.

1. Many of these studies have only been able to map metadata to a predefined ontology and, therefore, have no freedom to choose another ontology.
2. Often the need for such a metadata schema is subjective in terms of the end goal, and therefore not all buildings need to or have the required resources to execute existing methods to generate its schema.
3. Tag-based mapping relies solely on predicting the standard semantics based on the point descriptions. This method needs a lot of training data and human input. Point descriptions can also be limited regarding the information that can be extracted from them.

4. Additionally, previous work did not demonstrate the integration of generated schemas with the buildings and the practical application of such schemas for data-driven applications making it challenging to evaluate their usefulness.

3. Methodology

This section outlines the suggested approach to generating metadata schema from the legacy BAS points. Section 3.1 and 3.2 discusses the objectives of a metadata schema and types of metadata sources in buildings that can be used in the metadata schema generation process. Section 3.3 describes the method for classifying a list of BAS object identifiers into logical groups and how to use them to interpolate to a given full list of object identifiers extracted from a BAS, thereby reducing the number of total points for mapping. Section 3.4 describes a text search engine method to map a given identifier with the most suitable classes from the chosen ontology. We also discuss how much human interaction is needed to fully map all the identifiers to the classifications of the chosen ontology.

The proposed method can be summarized in five steps.

1. Set the objective clearly, why the metadata scheme is needed and how it will be used.
2. Sort out the relevant metadata required for schema generation and classify and group metadata into understandable and logical categories.
3. Use a text search engine populated with the classes and definitions extracted from the chosen ontology.
4. Run an automatic search and fine-tune the results with human input. Generate the metadata schema in RDF syntax.
5. Integrate the schema with the application to achieve the objectives set in Step 1.

Our proposed metadata schema generation method generally applies to mapping metadata to a chosen ontology. We demonstrate this methodology with the Brick ontology for our use case building.

3.1. The objective of the metadata schema generation

Buildings have many metadata sources, which are usually difficult to comprehend or access. First, it is important to decide on the objective of a metadata schema. This helps to narrow down both; the metadata source to be used and the pool of available ontologies and their classifications that can be used for creating a metadata schema. It also helps to create semantic graphs with semantic sufficiency[18] to execute an intended application. One objective could be to link the time-series data to the standard semantics of the sensors and equipment they are linked to for efficiently querying the data. Another objective can be using the semantic graph for reasoning over it for FDD applications[19], linking time-series data with BIM models[20] and so on. Depending on the requirement, one or many metadata sources can be included in the process.

3.2. Data collection

There are multiple sources of metadata in a building. Four such sources and the metadata available from them are illustrated in **Figure 1**. They include a list of BAS object identifiers, time series data from systems, Process and Instrumentation Diagrams (P&ID), and BIM models. As discussed above, if the objective is to enrich time-series data with standard semantics, source 1 and 2 would be sufficient. If the objective is to link time-series data with BIM models, sources 1, 2 and 4 are required. For an FDD application where process and instrumentation details are crucial, sources 1, 2 and 3 are required.

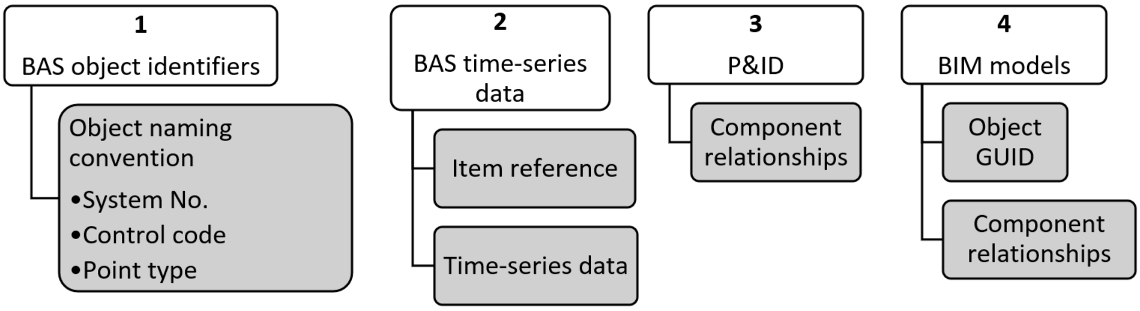

Figure 1: Metadata sources available in a building

3.3. Classifying BAS metadata

Although many sources are described above, in this section, we will focus on BAS metadata. BAS usually contains a lot of information about the device type, control commands, room numbers, etc. A partial list of BAS object identifiers extracted from a BAS of a building is shown in Table 1. This metadata table includes properties specified by the BACnet standard and those defined by the vendor. In the given example, metadata associated with each point in the BAS system includes properties such as Item Reference, Object ID, Object Type, and Point Name. The Item Reference is also the reference used in time-series data storage. Object ID and Type denote the BACnet Object identifier and type, respectively. The Point Name column assigns a unique name for each point. The description column provides text describing any additional information.

Table 1
Metadata extraction from BAS containing time-series reference (Item Reference) and descriptions.

	Item Reference	Object ID	Object Type	Point Name	Description (NL)	Description (EN)
1	XXX.FEC005.CLG-O	CLG-O	AO Mapper	(33) 201.CV-02V--	Regelafsluiter koeler	Cooler control valve
2	XXX.FEC006.CLG-O	CLG-O	AO Mapper	(33) 202.CV-02V--	Regelafsluiter koeler	Cooler control valve
3	XXX.SHWP1-FAULT	SHWP1-FAULT	BI Mapper	(33) 001.TP-01A--	Transportpomp 1 storing	Transport pump 1 malfunction

However, different vendors use different naming conventions, so each building has its unique naming. An example is shown in **Figure 2**. This name comprises four parts. The first part refers to the building number; the next part identifies the system type (such as 201 for AHU); the third position represents a part of the previous equipment (such as a valve belonging to AHU), and the last part indicates whether the point refers to a Measure, Report, or Control code.

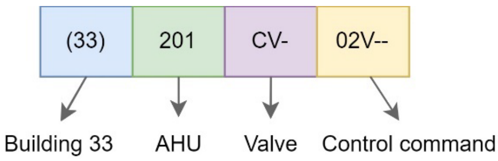

Figure 2: Point naming convention example

At this stage, points can be grouped into groups according to the naming convention. In the above example, since there are three parts in the naming convention (since the building no. does not change) that have distinct values, looking at each of these groups separately and mapping them to standard semantics is more efficient than, for example trying to find relevant classes for a list of thousands of identifiers.

3.4. Mapping to ontology using a search engine.

In this step, it is required to determine which ontology will be used for the metadata schema. Regarding BAS systems, two major standards are now being used, namely Brick and Haystack. A study by [21] compared the Brick and Haystack ontologies and showed that due to their excessive flexibility, Haystack often leads to unexpected representations. Brick being an ontology, provides more structured representations. Further, the Brick ontology maintains a wealth of documentation and is, therefore, straightforward for a developer. Therefore, we use the Brick ontology in this demonstration. Of course, several other ontologies, such as SSN or RealEstateCore, can also be used, depending on the use case.

The idea is to find the relevant classes from the Brick ontology that closely match the BAS point identifier classification system. As described above, the naming convention can be broken down into four parts, and each part can individually be mapped to the Brick ontology's classes. Further, any relationships derived by the point names can also be used to find the Brick relationships such as (*haspart, hasLocation* etc.).

A text search engine can be used to automate this mapping up to a certain level. This process is illustrated in **Figure 3**. Classes and definitions from the ontology can be extracted by running a SPARQL[7] query on the Brick ontology's RDF representation. The retrieved data can then be indexed using the Meilisearch[8] text search engine: the first index includes only class names, while the second index contains both class names and definitions.

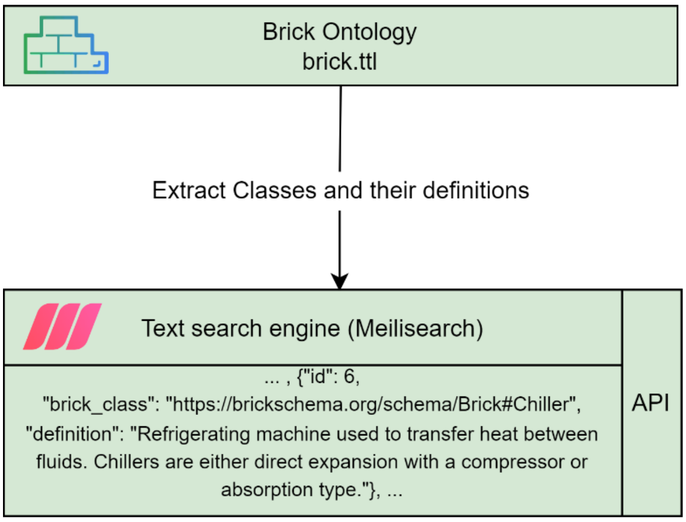

Figure 3 : Extracting class names and definitions from the ontology and populating the search engine.

Then, the BAS point classifications from the above step can be matched against the search engine. These matching results can then be reviewed with the assistance of a human expert to override any mismatching choices made by the text search engine. Mapping for the points shown partially in Table 1 is illustrated in **Figure 4**. Here, three parts of the naming convention were individually mapped to the Brick ontology. Results shown in green are the result of the text search engine, and the results shown in grey are the inputs from an expert human.

The final step is the generation of RDF-based schema out of the class mappings. Since the metadata is now available in CSV, it is possible to use a tool that generates the RDF graph from the CSV file by

[7] https://www.w3.org/TR/sparql11-query/
[8] https://www.meilisearch.com/

specifying the target relationships between columns. The Brick ontology group provides three such software tools[9]. The results will be the metadata schema in RDF syntax.

System No.

Identifier	Description (NL)	Description (EN)	Brick Class
001	Ketel 1	Boiler 1	Boiler
001	Ketel 1	Boiler 1	Boiler
101	Koelmachine 1	Chiller 1	Chiller
201	LBK 1	AHU 1	AHU

brick:Equipment

Control Code

Identifier	Description (NL)	Description (EN)	Brick Class
CV	Regelafsluiter	Regulating valve	Valve
MC	Vochtregeling	Humidity control	Humidity_Setpoint
CR	Cardreader	Card reader	Access_Reader
HT	Enthalpie Transmitter	Enthalpy Transmitter	Enthalpy_Sensor

brick:Equipment

Point type

Identifier	Description (NL)	Description (EN)	Brick Class
A--	Alarm	Alarm	Alarm
B--	Bedrijfsmelding	Operation status	Status
V--	Sturing	Control	Valve_Command

brick:Point

Figure 4 : Three parts of the naming individually mapped to the Brick ontology

4. Results

This section presents the results of the metadata schema generation method applied to a campus building located in Delft, Netherlands. This building is a living lab used in the Brains for Buildings project. One of the primary objectives of the living lab is to provide historical and real-time access to BAS sensor data. Unfortunately, the sensor data and metadata are not understood properly by users and, therefore, difficult to use for data-driven applications. To ensure that users understand the sensor data, contextual data must be added. This contextual data will enable developers to filter the sensor data of the equipment, sensors, and points of interest to them. Therefore, we aim to develop a tool that can be easily deployed in a building, providing the ability to query time-series data efficiently [20]. This includes the ability to query data by equipment or point types and gain initial insights about the data through data visualization. Therefore, our objective of the metadata scheme is to standardize the semantics of the time series data available from the BAS and provide them to end users who need to develop data-driven applications. With such an objective, the metadata schema generation becomes straightforward and requires less data than the literature. Our method categorises point types based on their equipment and point types, and we do not rely on ambiguous point descriptions.

4.1. Generating the metadata schema

To generate the metadata schema, we utilized a list of object identifiers extracted from the BAS, which is the primary source of metadata in the building (Table 1). We used recorded time series data from the BAS over one year, which contains references to the list of metadata and serves as the link between the two datasets. Additionally, we examined the Process and Instrumentation Diagrams (P&ID) of the systems; however, due to their format in PDF, we did not incorporate them into the

[9] https://docs.brickschema.org/lifecycle/creation.html#from-structured-tabular-sources

process. The building's Building Information Model (BIM) contained a significant amount of Mechanical, Electrical, and Plumbing (MEP) equipment and their relationships, but its usage was restricted due to inconsistent naming conventions with the BAS. Ultimately, we used the metadata extraction list and time-series data to generate the metadata schema.

We classified the naming convention into three parts (system number, control code, and point type). This results in another three mapping tables for each part of the naming convention, available in the vendor catalogues. However, our main objective is to link the BAS points with their time-series data and expose the time-series data to the user with its standardized metadata. Our applications of interest do require normalizing the metadata of all sensors in the building, but only specific sensors with time-series data. Therefore, the selection of identifiers can be narrowed down to those with associated time-series records. This can be done by matching the identifiers containing the time-series data for a selected period for all BAS points. In our case building, we reduced the full list of 2338 points to 948 points based on the availability of time-series records. Again, these points were filtered down to the points that followed the naming convention using a simple regular expression, further narrowing it down to 763 points.

Table 2 presents the number of classes the text search engine identified for the three different groups of identifiers in the naming convention. A maximum of three search matches were used to map the BAS point labels against both indexes. However, manual refinement was necessary with the assistance of human experts, as the text search engine could not map all point labels successfully. Some identifiers did not have a matching Brick Class, such as room control units (devices used to control room temperature, ventilation, and light levels), sprinklers, hydrophore installations (water pressurization systems used in areas with insufficient water pressure), and manual switches. The Brick ontology development group has acknowledged these omissions and intends to include them in future updates. Then, the results were interpolated to all 763 points of interest, each with three corresponding Brick classes according to its naming convention.

Table 2

Summary of matching Brick Classes

Part of the naming convention	No. of identifiers	No. of identifiers based on time-series data availability	No. of Identifiers mapped to Brick Classes
System number	39	16	11
Control code	117	40	28
Point type	34	19	12

In order to find a logical pattern between the three types of Brick classes we obtained from mapping, we examined a sample of the matching classes, as shown in **Table 3**.

Table 3

Summary of matching Brick Classes for the three parts of the point naming

Systems	Control Code	Point type
Boiler	Water_Pump	Open_Close_Status
Hot_Water_System	Gas_Valve	Alarm
Chilled_Water_System	Pump	Reset_Command
AHU	Valve	Power_Loss_Alarm
Exhaust_Fan	Damper	Status
Energy_Storage	Energy_Sensor	Setpoint
Fire_Safety_System	Flow_Sensor	Sensor
Breaker_Panel	Switch	Parameter

Our analysis revealed a logical pattern in the point labels in terms of the Brick ontology's arrangement of points, which can be used to effectively map the BAS points to the Brick ontology.

These relationships are shown in **Figure 5**. Specifically, the first column represents a *brick:Equipment*, while the second column denotes a part of the equipment mentioned in the first column. The relationship between these columns can be established using the Brick ontology's *brick:hasPart* property. Finally, the last column specifies the point's characteristics, such as whether it is an alarm, setpoint, sensor, or status, which can be linked to the Brick Point class, a subclass of the Brick class, using the *brick:hasPoint* property. Finally, the Points are related to their timeseries data by using the *ref:hasTimeseriesId* relationship.

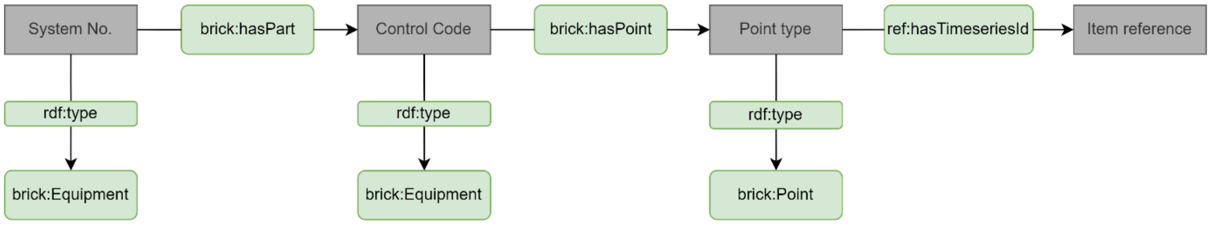

Figure 5: Relationships among the metadata identifiers.

However, not all available points had relevant Brick classes, and not every point followed this logical pattern. There were a few point labels which did not follow this logic.

1. 25 out of 763 did not follow the *brick:equipment* class. However, all these 25 points belonging to a miscellaneous category are irrelevant enough to be included.
2. 157 points did not match the *brick:hasPart* relationship, mainly because they also redundantly described the *brick:point* type, which was again found under "point type".

Since we identified the logical arrangement of the points in the BAS, we then created the metadata schema of the building using the Brick Builder[10] CSV to RDF tool. Part of the RDF graph for an AHU is shown in **Figure 6**. This graph is further uploaded to GraphDB[11], an RDF store optimized for graph data. This graph will be used in the next step when integrating with time-series sensor data.

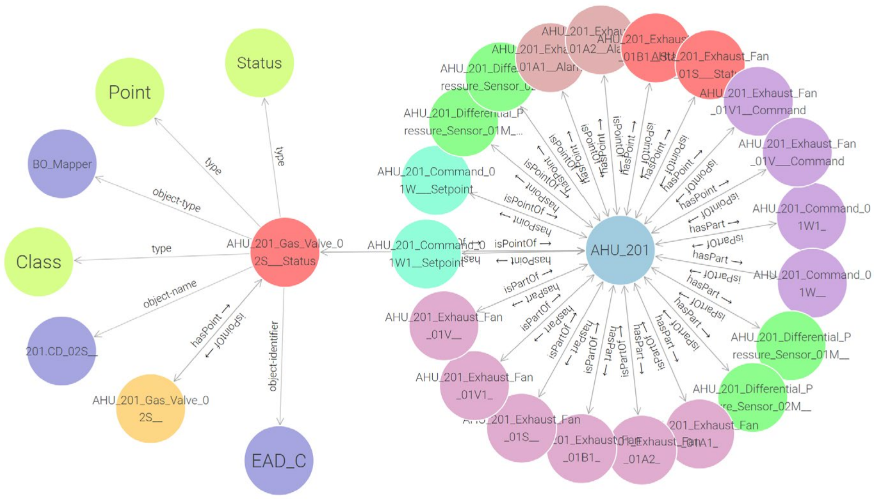

Figure 6: Part of the metadata schema containing the AHU and its Points represented graphically using GraphDB interface.

[10] https://github.com/gtfierro/brick-builder
[11] https://www.ontotext.com/products/graphdb/

4.2. Integration with time-series data

One of the main aims of the developed RDF graph is to provide users with a seamless experience while exploring the vast amounts of time-series data available. In this context, we considered time-series data for one year, which was pre-processed to transform and handle missing values and then recorded in TimescaleDB. We developed an Application Programming Interface (API) to execute queries against the metadata schema and fetch relevant time-series data from TimescaleDB. To facilitate user exploration of both metadata and time-series data, we also provided a Grafana dashboard. API facilitates data exchange between the Timescale DB, GraphDB and the Grafana web application. This API also enables users to authenticate and authorize themselves to access the databases and query time-series data.

Using the Grafana dashboard, users can query the BAS points of interest by selecting the equipment type from a dropdown list (green box in see **Figure 7**). This list contains all the equipment types in the generated metadata schema. For example, in **Figure 7**, the equipment type AHU is selected as the *brick:Equipment*. This selection triggers a SPARQL query that requests the time-series identifiers related to all the points related to *brick:AHU* in the graph. The resulting point list is then displayed (red box in **Figure 7**), and users can select one or more of these identifiers to explore the time-series data. In the example shown in **Figure 7**, a *brick:Differential_Pressure_Sensor* has been selected for exploring the time-series data. This selection results in an API request being sent to the time-series database with the time-series identifier, and the charts are subsequently populated with data. Overall, this approach provides an efficient and user-friendly means of exploring large amounts of time-series data with their standard semantics.

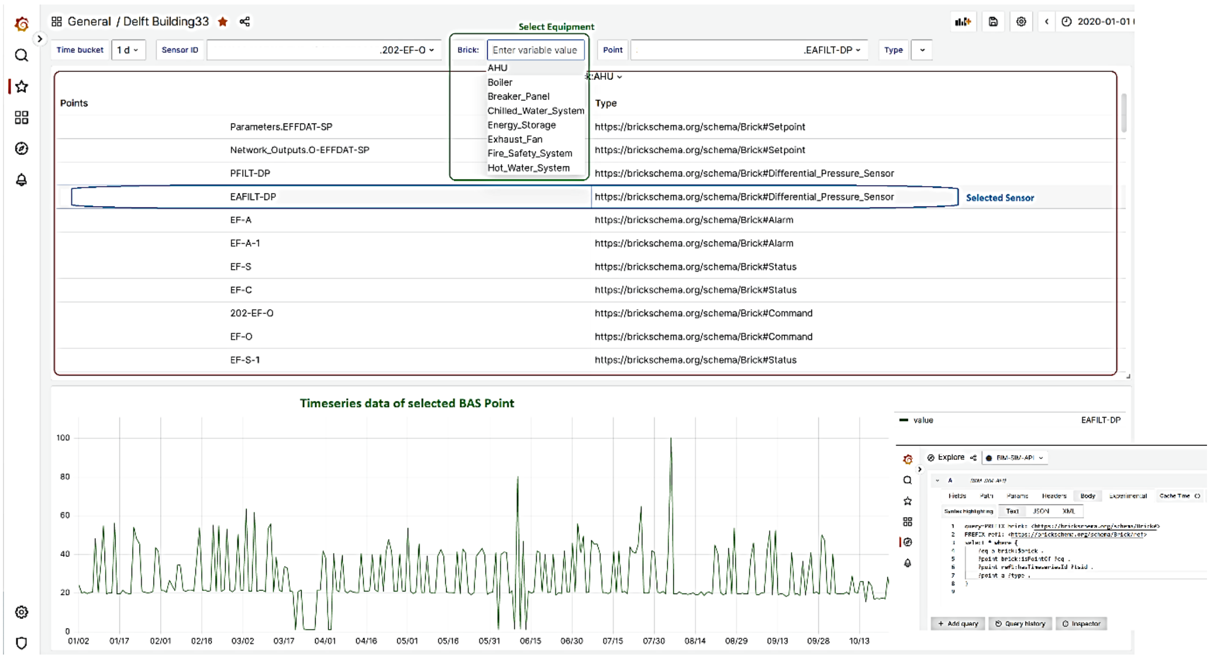

Figure 7: Grafana application integrated with metadata schema of the building.

5. Conclusion

This research article presents a five-step methodology for creating a semantic graph and its integration with time-series data, with a further demonstration on how this generated metadata schema can be queried through the Grafana web application. Using the generated schema, we could quickly sort, and group BAS data based on their equipment type and directly link the resulting time-series data to a visualization environment. This led to a significant improvement in the quality of data access.

However, like any research, this methodology also has its limitations. Notably, the proposed method for linking metadata is limited in its ability to describe the complex relationships between equipment components and points, for example, fluid flow and spatial relationships. These relationships are typically available in P&ID in pdf formats, which is a large barrier to extracting their relationships. A potential research direction is investigating how these image-based relationships can be extracted and standardized since complex relationships between systems and components are necessary in some use cases. To this end, ontologies such as FSO [22] and TUBES [23] may be useful, depending on the type of application to be developed with the data. However, generating a semantic graph that covers every aspect is impossible. Recent developments, such as application-based semantic graph generation [18], aim to create semantically sufficient graphs for a given application.

Further improvements to this research include integrating the BIM model in the process of creating the semantic graph and using it as a visualization tool. However, initial evaluation of available BIM models has revealed inconsistencies in naming conventions between BIM and BAS and a lack of sufficient spatial information. An automated procedure for creating a metadata schema that addresses complex relationships within the building and its systems is not possible except if (1) the BIM model is appropriately modelled; (2) naming conventions are followed; and (3) mapping between naming conventions and the Brick ontology is available, or the Brick ontology is used to annotate the BIM model in the first place.

An important takeaway from this project is the need to enforce a metadata schema by building owners to the BAS providers, at least for newly constructed buildings. This will enable easy and fast integration of systems, allowing development efforts to focus more on energy-saving algorithms rather than data extraction methods. Overall, the proposed method has the potential to facilitate more efficient and effective analysis of building systems and the data, but further research is necessary to address the limitations and improve its applicability to a wider range of buildings and use cases.

6. Acknowledgements

The Brains for Buildings project received funding from the Dutch Ministry of Economic Affairs and Climate Policy and the Ministry of the Interior and Kingdom Relations under the MOOI program. We would also like to express our gratitude to the data platform team from TU Delft for their invaluable support in providing the researchers with the BAS data sets used in this study.

7. References

[1] Honeywell, "Engineering Manual of Automatic Control for Commercial Buildings," 1997.

[2] A. Bhattacharya, J. Ploennigs, and D. Culler, "Short Paper: Analyzing Metadata Schemas for Buildings," in *Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments*, Nov. 2015, pp. 33–34, doi: 10.1145/2821650.2821669.

[3] A. A. Bhattacharya, D. Hong, D. Culler, J. Ortiz, K. Whitehouse, and E. Wu, "Automated metadata construction to support portable building applications," *BuildSys 2015 - Proc. 2nd ACM Int. Conf. Embed. Syst. Energy-Efficient Built*, pp. 3–12, 2015, doi: 10.1145/2821650.2821667.

[4] B. Balaji *et al.*, "Brick: Metadata schema for portable smart building applications," *Appl. Energy*, vol. 226, no. September 2017, pp. 1273–1292, 2018, doi: 10.1016/j.apenergy.2018.02.091.

[5] V. Charpenay, S. Kabisch, D. Anicic, and H. Kosch, "An ontology design pattern for IoT device tagging systems," *Proc. - 2015 5th Int. Conf. Internet Things, IoT 2015*, pp. 138–145, 2015, doi: 10.1109/IOT.2015.7356558.

[6] M. Compton *et al.*, "The SSN ontology of the W3C semantic sensor network incubator group," *J. Web Semant.*, vol. 17, pp. 25–32, 2012, doi: 10.1016/j.websem.2012.05.003.

[7] K. Janowicz, A. Haller, S. J. D. Cox, D. Le Phuoc, and M. Lefrançois, "SOSA: A lightweight ontology for sensors, observations, samples, and actuators," *J. Web Semant.*, vol. 56, no. July

2019, pp. 1–10, 2019, doi: 10.1016/j.websem.2018.06.003.
[8] W. Terkaj, G. F. Schneider, and P. Pauwels, “Reusing domain ontologies in linked building data: The case of building automation and control,” *CEUR Workshop Proc.*, vol. 2050, 2017.
[9] K. Hammar, E. O. Wallin, P. Karlberg, and D. Hälleberg, “The RealEstateCore Ontology,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2019, vol. 11779 LNCS, doi: 10.1007/978-3-030-30796-7_9.
[10] A. Schumann, J. Ploennigs, and B. Gorman, “Towards automating the deployment of energy saving approaches in buildings,” *BuildSys 2014 - Proc. 1st ACM Conf. Embed. Syst. Energy-Efficient Build.*, pp. 164–167, 2014, doi: 10.1145/2674061.2674081.
[11] B. Balaji, C. Verma, B. Narayanaswamy, and Y. Agarwal, “Zodiac: Organizing large deployment of sensors to create reusable applications for buildings,” *BuildSys 2015 - Proc. 2nd ACM Int. Conf. Embed. Syst. Energy-Efficient Built*, pp. 13–22, 2015, doi: 10.1145/2821650.2821674.
[12] J. Koh, B. Balaji, V. Akhlaghi, Y. Agarwal, and R. Gupta, “Quiver: Using Control Perturbations to Increase the Observability of Sensor Data in Smart Buildings,” 2016, [Online]. Available: http://arxiv.org/abs/1601.07260.
[13] D. Hong, J. Ortiz, K. Whitehouse, and D. Culler, “Towards automatic spatial verification of sensor placement in buildings,” *BuildSys 2013 - Proc. 5th ACM Work. Embed. Syst. Energy-Efficient Build.*, 2013, doi: 10.1145/2528282.2528302.
[14] J. Gao and M. Bergés, “A large-scale evaluation of automated metadata inference approaches on sensors from air handling units,” *Adv. Eng. Informatics*, vol. 37, no. February, pp. 14–30, 2018, doi: 10.1016/j.aei.2018.04.010.
[15] J. Fürst, K. Chen, R. H. Katz, and P. Bonnet, “Crowd-sourced BMS point matching and metadata maintenance with Babel,” 2016, doi: 10.1109/PERCOMW.2016.7457102.
[16] M. Pritoni, A. Bhattacharya, D. Culler, and M. Modera, “Short paper: A method for discovering functional relationships between Air Handling Units and Variable-Air-Volume Boxes from sensor data,” *BuildSys 2015 - Proc. 2nd ACM Int. Conf. Embed. Syst. Energy-Efficient Built*, pp. 133–136, 2015, doi: 10.1145/2821650.2821677.
[17] S. Li, D. Hong, and H. Wang, “Relation Inference among Sensor Time Series in Smart Buildings with Metric Learning,” *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 04, pp. 4683–4690, Apr. 2020, doi: 10.1609/aaai.v34i04.5900.
[18] G. Fierro, A. Saha, T. Shapinsky, M. Steen, and H. Eslinger, “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*, Nov. 2022, vol. 1, no. 1, pp. 228–237, doi: 10.1145/3563357.3564083.
[19] P. Delgoshaei, M. Heidarinejad, and M. A. Austin, “A Semantic Approach for Building System Operations: Knowledge Representation and Reasoning,” *Sustainability*, vol. 14, no. 10, p. 5810, May 2022, doi: 10.3390/su14105810.
[20] L. Chamari, E. Petrova, and P. Pauwels, “A web-based approach to BMS, BIM and IoT integration: a case study,” in *REHVA 14th HVAC World Congress*, 2022, pp. 1–8, doi: https://doi.org/10.34641/clima.2022.228.
[21] C. Quinn and J. J. McArthur, “A case study comparing the completeness and expressiveness of two industry recognized ontologies,” *Adv. Eng. Informatics*, vol. 47, no. October 2020, p. 101233, 2021, doi: 10.1016/j.aei.2020.101233.
[22] V. Kukkonen, A. Kücükavci, M. Seidenschnur, M. H. Rasmussen, K. M. Smith, and C. A. Hviid, “An ontology to support flow system descriptions from design to operation of buildings,” *Autom. Constr.*, vol. 134, p. 104067, Feb. 2022, doi: 10.1016/j.autcon.2021.104067.
[23] N. Pauen, D. Schltitter, J. Frisch, and C. Van Treeck, “TUBES System Ontology: Digitalization of building service systems,” *CEUR Workshop Proc.*, vol. 3081, pp. 43–54, 2021.