A Reference Architecture for Data-Driven Smart Buildings Using Brick and LBD Ontologies

Pieter Pauwels [a], *Gabe* Fierro [b]

[a] Department of the Built Environment, Eindhoven University of Technology, Eindhoven, the Netherlands, p.pauwels@tue.nl.

[b] Colorado School of Mines, Golden, CO 80401, United States of America, gtfierro@mines.edu.

Abstract. With the increasing adoption of sensors, actors and IoT devices in existing buildings, the real estate sector is becoming increasingly automated. Not only do these devices allow to monitor these buildings (energy use, occupancy, indoor air quality, etc), they also enable model predictive control (MPC) through building automation and control systems (BACS). A critical feature to enable these is the metadata associated to data streams obtained from the building. Such metadata allows building operators to assess what these data streams are, what they are measuring and how. This can be achieved using metadata schemes and vocabularies, such as Brick, Haystack, Linked Building Data, Industry Foundation Classes. Merging these model-based metadata schemas (semantics) with data-driven monitoring and control (machine learning) into a functional system architecture is a considerable challenge. In this paper, we review the mentioned technologies and propose a draft reference architecture based on state-of-the-art research. This reference architecture is evaluated using a set of predefined criteria.

Keywords. Smart buildings, Linked Data, Ontology, Metadata, Brick, FDD, MPC, Asset Management
DOI: https://doi.org/10.34641/clima.2022.425

1. Introduction

The built environment is rapidly digitizing. The design and engineering stages of projects are increasingly impacted by digital tools. The management and operation of facilities is also experiencing increasing use of IT technologies. This motivates the creation of smart buildings, many of which are primarily data-driven. A smart building is hereby understood as a building for which the operation is monitored, and in which the building can intervene itself in the operation, towards optimized use. Optimised use of a smart building hereby often targets either an optimised energy use [1], as energy is becoming increasingly expensive and scarce; or an optimisation towards the end user, in which case comfort stands central.

To optimize smart buildings using smart IT technologies, data monitored from these buildings is a primary source of information (temperature, electricity use, indoor air quality, etc.). Analytic techniques examine monitored data and retrieve patterns upon which actions can be taken [2,3]. Metadata schemas and annotations [1] can “tag” these data sources with contextual details that make analytics results more easily interpretable (e.g., needing to know which room an anomaly occurs in, and knowing the features of this room, such as presence of windows, specific devices, users, etc.) or potentially easier to configure and program [4]. While standardisation of these annotations is often sought, the majority of buildings rely on ad hoc labels and naming conventions to describe their data streams. A growing family of metadata ontologies and vocabularies for buildings aims to address these issues, but it is not always clear how these technologies complement or compete with one another, or how exactly they fit into a modern view of a data-driven building [1,3].

In this paper, we investigate available metadata schemas for data-driven smart buildings. We study how these metadata schemas can be combined with key data analytics technologies (monitored data), and we demonstrate a set of contemporary system architectures (e.g. [3]). Eventually, this paper presents an opinionated *reference architecture* that explicitly demonstrates how these metadata technologies can be combined with data monitoring equipment, support data-driven applications in buildings, and thus enable the development of smart data-driven buildings.

The rest of the paper proceeds as follows. Section 2 provides relevant background on existing smart building ontologies and other relevant vocabularies in addition to a brief overview of the capabilities and features of existing data platforms. Section 3 applies a set of motivating applications to defining the design requirements for a representative platform architecture enabling data-driven applications. Section 4 presents the representative architecture built on the Brick [5] and Linked Building Data (LBD) ontologies [6], complemented with data monitoring devices and software as well as control algorithms aiming at model-predictive control (MPC).

2. Background

Before being able to build a reference architecture, a review is needed of the different components that are needed inside that architecture. In this Section, we investigate (1) available smart building ontologies, (2) existing data platforms, and (3) platform design patterns.

2.1 Smart Building Ontologies

In the last decade, plenty of ontologies and vocabularies have been created, investigated and proposed [7]. These ontologies cover diverse areas. For one, in the design and engineering domain, there is the long-standing IFC ontology or vocabulary [8] that is available in the Web Ontology Language since 2016 [9]. In parallel to that ontology, there exists the combination of Linked Building Data (LBD) ontologies [6], which are diverse combinations of building topology ontology (BOT – Fig. 1 – [10]), building element ontology (BEO) [11], Building Product Ontology (BPO) [12], MEP ontology [13], diverse props ontologies that provide specialised property sets, the Ontology for Property Management (OPM) [14], the File Ontology for Geometry Formats (FOG) [15,16], and the Ontology for Managing Geometry (OMG) [15,17]. With the modular approach towards the creation of these ontologies [6-7], it is fairly possible to create more ontologies and use those in combination. In that case, one needs to be careful of the overlap between multiple ontologies, and how a combination of two ontologies may best be used [7].

When considering also the asset management phase and existing buildings and infrastructure in general, diverse other ontologies can be of additional use, such as SAREF4BLDG [18], the Damage Topology Ontology (DOT) [19], Real Estate Core (REC) [20], and so forth. However, in the smart building area, and in the operational phase of the building, a different community of ontologies exists, that is much more focused on the data streams collected in the building as well as its systems and devices.

The most predominant ontology in this ecosystem is the Brick ontology [5, 21]. At its core, Brick consists of a class brick:Location, which groups the definition of Building, Space, Floor, Site, and Zone. These location classes are very similar to the core BOT classes (Fig. 1). They can be combined using the simpler hasPart and part of relations, thus enabling to build any building topology, in a less rigid manner compared to BOT (no domains or ranges in object properties). Further to the Location class, Brick also defines an Equipment class, System class and a Point class, each holding a considerable inheritance tree below to define specific subclasses of both. As such, the Brick ontology is very well capable of defining the system of a building, including those points of relevance for data monitoring.

The Project Haystack metadata model [22] differs substantially from other building ontologies in that it is not rooted in RDF technologies. Though similar in scope to Brick, Haystack instead describes the type, behavior and other properties of entities in the model using combinations of tags --- atomic terms with pre-defined meaning. Recent efforts have begun to standardize the choice of tags to ensure consistent and interpretable use, but the majority of Haystack usage is unrestricted and correspondingly irregular [23].

The Flow System Ontology (FSO) is aimed entirely at the flows, ports and interfaces in an HVAC system [24]. FSO allows to define systems, components and connections, including energy storage devices, supply units, and fluid and heat flows. As such, it is much closer to the design and engineering area compared to Brick and Haystack. The ontology can mainly be used to define and compute flows through a system, and design its dimensions accordingly to acquire an optimised HVAC system.

Existing ontologies such as QUDT, SSN/SOSA, O&M, Time, and others can easily be aligned and combined with the above mentioned ontologies, as is also commonly done.

2.2 Existing Data Platforms

In addition to the above ontologies and vocabularies, many data platforms have emerged to

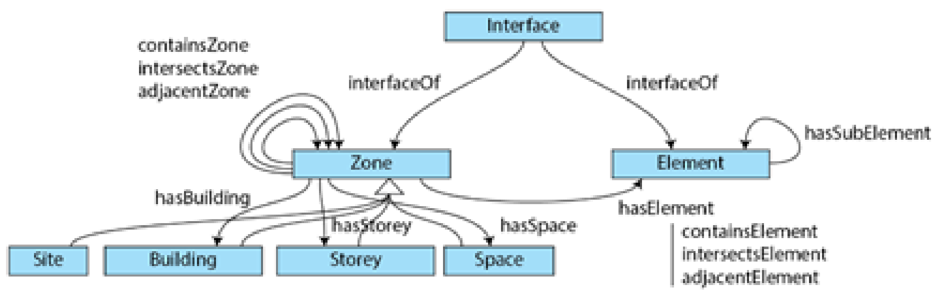

Fig. 1 – Main classes and properties in the BOT ontology.

help users organize, analyse and otherwise make sense of the data delivered by buildings. These include not only commercial platforms but also smaller open-source and community-driven efforts. The platforms examined below all make use of some sort of metadata to support data-driven smart buildings. The data itself is dominated by streams of time-series data collected from sensors and control networks, but also includes asset management data and static attributes of the building itself.

The SkySpark platform [25] ingests data from buildings using a variety of data adaptors and provides an API for constructing data-driven applications over that data. The results of these applications, like fault histories, are rendered in a web-based dashboard. SkySpark uses Haystack tags as the underlying metadata representation and defines a simple filter language over those tags to help users browse available data streams, and to allow applications to identify relevant data sources.

Google's Digital Buildings [26] effort defines an open-source ontology and schema for describing and reporting building data. The platform is not publicly documented; however, the ontology documentation demonstrates that it is possible to organize related data streams by the metadata provided by the ontology. If the ontology gets properly published in the future, this would make it easier for applications to discover relevant data because they only need to know the name of the collection containing the data they need.

The Azure Digital Twins platform [27] is another recent metadata-based data platform for smart buildings. The platform models properties of and interactions between data-producing entities in a cyber-physical environment (such as a building) as a graph using the RealEstateCore ontology [20] and a custom Digital Twin Description Language (DTDL). Applications query the graph to access live and historical data about the environment which is gathered using other data ingestion services.

Recent academic work highlights the importance of metadata in implementing and deploying data-driven applications [2,3,7,28-32]. The BOSS work develops a set of operating system-style services for executing portable and fault-tolerant applications [29]. Portable applications are those which require little to no adjustment to their implementation to execute in a variety of environments. BOSS achieves this through late-binding of resources discovered via a graph-oriented query mechanism. The Mortar work [28] further develops the notion of application portability through the construction of an analytics platform incorporating Brick-based graph models of buildings and a large public release of data. LBD reference architectures are different in style, also among each other. Malcolm et al. [30] defines a relatively standard web infrastructure in which multiple RDF triple stores are made available in combination with a web-based 3D viewer using the glTF data format. Ontological data integration happens in a modular fashion, similar to how it was described in Schneider et al. [32], but with the monitoring data outside the RDF graphs. This LBDServer namely also includes timeseries databases, and can then enable the joint access of building data together with monitoring data, within a state-of-practice web development framework. Werbrouck et al. [31] leverages this work and aims to make it available through the SOLID web data framework [33].

In the majority of the above systems, data is collected in state-of-the-art databases and amended with one or more metadata schemas to provide structure, standardisation and reference.

2.3 Platform Design Patterns

A critical question is exactly how metadata models can be used to build data-driven applications for buildings. We identify three representative design patterns, and identify later on in this section where these may be of more value and fit. In the first pattern, the metadata model is used as a reference for the programmer. A software developer may use certain representations --- such as building information models (BIM), one wire diagrams, blueprints and floor plans --- to help themselves understand the structure and composition of the building in its subsystems. Although this metadata is not consumed in an automated fashion, it helps the programmer understand what algorithms or approaches are appropriate, and which data sources and I/O points to consider when implementing the desired functionality.

The second design pattern uses the metadata model as an internal addressing and filtering system for manipulating data in a particular platform, as typified by Project Haystack. A data platform may contain the data and interfaces to those data sources and relevant I/O points. The programmer needs a way to identify which endpoints, API calls and parameters in the system are required to implement the desired functionality. This metadata may be added as an aftermarket feature, such as in a platform like SkySpark or the Google digital building stack, or it may consist of point names placed there by the installer of the building management system. An advantage of this second approach is that it is highly flexible. As no rigid ontology or standard is followed in annotating the data (no or hardly no metadata schema), the implementer or data engineer has a lot of freedom to work with the data collected from sensors and devices. These can simply be retrieved and used, provided that the implementer or user keeps track of how the data is actually structured. This last part is therefore also a weakness of this system: flexibility leads to many disagreements and confusion and then errors [23].

In the third design pattern, the metadata model is used to contextualize and further model resources that are contained in external, incumbent systems. In this context, metadata is implemented with linked data (specifically RDF graphs). This approach is like the second design pattern but is critically different in that it includes formal or semi-formalized information that can be extended and validated. This semi-formalised information is enabled by the use of an OWL ontology in creating the RDF graphs. The scope of these metadata models is also not limited to the features provided by anyone ontology. Instead, the developer of this model can choose which elements, aspects or assets of the building are necessary for an intended suite of applications. An advantage of this third approach is that it normalizes the representation of metadata from various other systems, such as the metadata provided by the prior two patterns. Linked data decouples the meaning, or semantics, of metadata from its representation in a particular siloed system. Multiple ontologies can be used to represent the building [32] and thus provide an agreed upon representation and context of the data collected for the building.

Given the rising interest in developing ontologies based on linked data technologies (RDF graphs), we choose to examine data platforms of the third type. Metadata represented as linked data can be harder to integrate into existing data platforms for several reasons. First, the RDF format [34], describing a directed labelled multigraph, is quite abstract compared to the relational or object-oriented models many software developers are familiar with. Second, software supporting linked data is often quite generic due to the wide family of possible uses of linked data. It is not always obvious how to use a linked data system in conjunction with traditional aspects of a data system, such as a time series database [30]. Third, some ontologies such as SSN and QUDT formalize how to represent data and telemetry inside the RDF graph. However, this is far from the most intuitive or high-performing way of representing and accessing this data [32,35]. This can cause potential confusion for consumers or implementers of a linked data-based system.

2.4 Domains of Relevance

While this paper orients entirely towards data-driven smart buildings, there still exist several subdomains or areas of relevance. Each of these subdomains has a specific aim and corresponding business value. Depending on that business value, one or the other reference architecture may be more relevant.

A first domain of relevance is *asset management*. In a range of cases, primary relevance in building management is asset management. In such cases, the amount of monitored data is relatively small, and semantics play a much bigger role. As an example, a system or platform may be devised that has a semantic building model at its core, and additions are made using RDF graphs to further enrich the building model and its assets [30,31]. Enrichments then typically consist of annotations of damage and material wear (e.g. DOT in [19]). That damage can be represented using RDF graphs (e.g. using the DOT ontology [19]), and this often includes references to the collected measurements (e.g. element creep, cracks in concrete, photos, etc.). In this subdomain, the aim of the data-driven part in the data-driven smart building is mainly to collect data for analysis. The smart building is not self-repairing or self-cleaning, and therefore this can hardly be called a data-driven smart building.

A second domain of relevance is the systems and control domain. In this domain, a detailed view is required of the system inside the building, as well as how it is functioning (e.g. the purpose of the BACS [36] and FSO ontologies [24]). In many cases, this includes feedback loops and control models that allow to intervene in the building systems. Such subdomain typically relies on a Building Management System (BMS) that inhibits a detailed physical model of the building system (model-based), *and* provides access to monitoring data that is collected about those systems' operation. As such, the state of the system can at any given moment be tracked, and automated routines allow to intervene in the building (e.g. closing blinds, initiate heating, etc.). These systems tend to rely on two components, namely (1) a good physical model or simulation model that can compute at any time the expected performance and actions in the system, and (2) a live data-stream that logs the system status and can always be compared with the simulation model. If corresponding actions are taken, then this leads to Model-Predictive Control (MPC) [37].

Historically, semantics and metadata have had little to contribute to this specific target (MPC, systems and control) because of the reliance on detailed, hand-built, white-box models of the physics and dynamics of the system. However, recent work on grey-box and black-box models has heightened the need for systems which can organize the data required to fit these models. Both the FSO and BACS [24,36] ontologies aim to represent the system and all its components and flows using a semantic model. Models of the system's dynamics can then be determined by the structure of the related graph and historical telemetry of the system. The predictions from the model are used to implement MPC. MPC is the dominant control strategy studied in intelligent building research and has been demonstrated to provide thermal comfort to occupants while meeting energy consumption constraints, among other successes [37].

The last domain of relevance is data monitoring and prediction [2-3]. In this domain, the main purpose is to collect data for monitoring the building's function and predict its state over time, not limited to the state of the system and its devices. In this case, the focus lies heavily on data monitoring points (sensors) that provide access to monitored data. In this scenario, the focus lies a lot more on the actual time-series data that is collected, and a lot less on the definition of the system or its flow in full detail. A more commonly explored business opportunity in this case lies in the detection of patterns in the sensor data, to be able to detect and diagnose faults over time (Fault Detection and Diagnosis – FDD). As there is much less need for a full representation of the system, this approach more often relies on the above outlined Approach 2, where data points are tagged and annotated, but mainly kept flexible and simple.

3. Platform Requirements

Those who would implement a platform for data-driven buildings must take into consideration the diversity enumerated above. Data-driven use cases in buildings consist of a variety of usage scenarios serving different goals. Implementation of these use cases may be facilitated through the incorporation of structured metadata. In this section, we examine the requirements that the above use cases -- model predictive control and fault detection – place upon a hypothetical data platform. These requirements will be incorporated into a representative architecture discussed in Section 4.

3.1 Requirements for an MPC application

Recall that MPC is an advanced control strategy which uses live and historical building telemetry to construct and refine a predictive model that allows a controller to implement an optimal policy. At its most essential, any MPC implementation requires three components. First, an MPC implementation needs a way of optimizing a control sequence relative to the predictions output by a model. The optimization itself is usually outsourced to an existing optimizer, but the MPC controller must be able to interface with that optimizer and use its results as part of the control process. Secondly, an MPC implementation requires a means of recording live observations about the state of the building. MPC controllers must constantly relay the feedback of their decisions to the optimizer to ensure they are making the best control decision at each opportunity. It is not always necessary for an MPC implementation to retain all historical data; often only the most recent data (e.g. one week) is required to update the model and inform the optimizer. Thirdly, an MPC implementation requires a means of actuating the environment with the control decisions reached by the optimizer. Depending on the deployment, it may be necessary to provide certain desirable properties for the actuation. As the BOSS work [29] details, transactional control can be beneficial in physical settings where having a building in an intermediate state can waste energy or even cause harm. Other deployments may be content with having an MPC controller write low-priority control signals that can be overridden by other safety software.

If the MPC implementation is using a white-box model to provide the predictions, then the supporting metadata only needs to describe the relevant data so it can be found by the controller. For grey-box and black-box models, the controller will require some knowledge about the structure and composition of the building in order to know what kinds of data sources would be relevant and what kind of model to train. While MPC applications may be less data-intensive, they generally are much more complicated on the systems and control side (signal processing and communication on the edge), which impacts software (and hardware) architecture significantly.

3.2 Requirements for an FDD application

FDD applications leverage a wide variety of computational models and algorithmic techniques. Some FDD approaches use equipment model- and manufacturer-specific information to identify errors in the equipment's operation (closer to FSO and BACS approaches – [24,36]). Others use timeseries analysis techniques to identify deviations from normal patterns, without having to know anything specific about the operation of a system [2,3]. Still others adopt an expert-system approach by evaluating rules over live telemetry from the building.

In contrast to MPC applications, which may require information about the composition and topology of an entire subsystem (usually HVAC), FDD applications typically require much more localized metadata. Many FDD rules are concerned with the operation of individual pieces of equipment. This means that the supporting metadata should capture the identification of equipment for FDD, and the properties of that equipment required to properly configure the FDD. These applications are therefore much more detailed and data-intensive, and are more likely needing multiple types of databases in their software architecture. Most FDD applications do not need the ability to actuate the building. This simplifies the platform requirements, especially compared with MPC. As with MPC, FDD application require the platform to provide a means of recording live observations about the state of equipment in the building.

4. A Representative Architecture

From a software architecture perspective, the requirements detailed above are nothing new. Many existing solutions --- building management systems, IoT platforms, SCADA systems, and so on --- incorporate a classic "three-layer cake" (Figure 2) above the hardware. A hardware abstraction layer interacts directly with deployed devices to provide standardized I/O interfaces to the rest of the platform. A system services layer provides storage, routing, scheduling and other essential functionality. Finally, an application layer on top presents an API and possibly an execution environment for software running on the platform.

This basic architecture is very flexible and leaves room for implementation choices such as which communication protocol and database to use. However, it is much less clear how metadata ontologies fit into this picture: what types of technologies should be used, and how do they interact with the other components of the architecture? Furthermore, which metadata schemas should be used and to what extent? This particularly related to the two top layers in the given three-tier architecture in Fig. 2. Depending on the top-level application(s), database infrastructure and storage techniques can differ. Furthermore, application logic, e.g. computational models and black box prediction models, impact the stack further towards the two top tiers of the architecture.

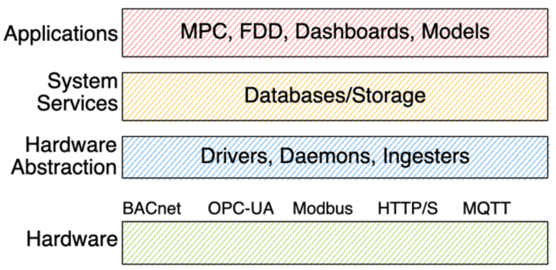

Fig. 2 – The three-layer architecture interacting with cyber-physical hardware (bottom) through a variety of protocols.

Based on the previous section, it is clear that the FDD case has the highest impact on the data infrastructure, as this has more challenging requirements on data infrastructure and prediction algorithms. The asset management case is less oriented towards live measurements, yet needs a similar semantic core to be managed for the building. The MPC case lastly is much more oriented towards the lower layers of the software stack and the devices (signal processing, communication and control). Hence we continue the construction of a reference architecture in this section with the FDD case primarily in mind, complemented with asset management features.

In the following sections, we first outline what the data infrastructure and computational setup might best be in the first and second tier of the 3-tier stack of Fig. 2. This includes a draft schema of how different databases and algorithms may best be combined and how all data can be interlinked without a too large data management overhead. Second, we elucidate in more detail how ontologies can support the software in the architecture. This is performed via the incorporation of a *metadata graph* that contains the relevant information to configure applications and enable them to discover important data sources and I/O points. The structure and content of the graph is informed by a family of RDF-based ontologies.

4.1 Computation-friendly Data Infrastructure

The first and second layer in the three-tier entire architecture of Fig. 2 are key in building a computation-friendly data infrastructure. As indicated before, the data infrastructure is highly required to include multiple types of databases (see Fig. 3). This cannot be a single database, not in RDF, not a relational database, not an object-oriented database (key-value store), because of the need for a very diverse set of data of different nature. Hence, the core of the reference system architecture needs to consist of a combination of these, as was also argued in many other places [2,3,7,30-32,35]. While flexibility may be useful in some cases, a highly interconnected network of data is better off with a defined and rigidly enough structure. Therefore,, it is highly recommended to adopt the more rigid ontologies for the management of the semantic data, in which case the LBD ontologies and the BRICK ontology and the IFC ontology are a better choice.

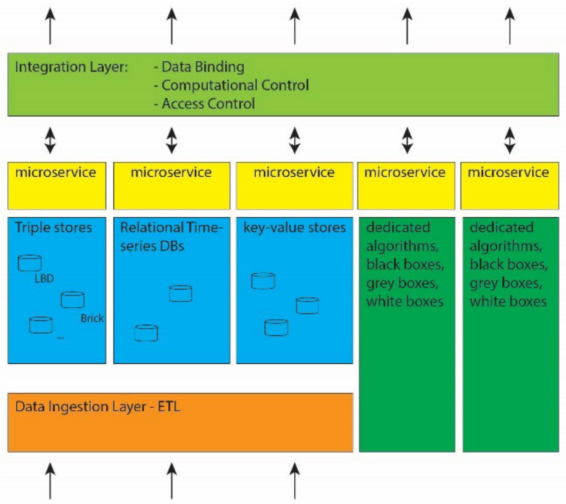

Fig. 3 – Core data infrastructure in combination with pre- and post-processing layers as well as computational components behind a microservice architecture.

To be able to connect to the upper and lower tiers, appropriate connectors are required, which can pre-process and post-process data coming in and going out of the system (top and bottom layers in Fig. 3). For example, in the case of asset management, a legacy IFC dataset needs to be transformed into the different core data models used in the system (not IFC). Similarly, timeseries data and the reference tags with which they are classified need to be put in correctly in the BRICK ontology as well as the corresponding timeseries databases. Furthermore, dedicated algorithms for FDD need to be provided in addition to the data, and these can be made accessible through microservices through the upper tier layer of the data infrastructure (Fig. 3).

4.2 Architecture of Supporting Ontologies

When considering more closely the metadata schemas and ontologies deployed in the proposed data infrastructure, we can clearly see how an “architecture” of ontologies is available that together forms the structure of a system metadata graph. In this case (FDD and asset management), we recommend here to adopt an entity-centric approach for the metadata supporting a data platform for buildings. Entities are the physical, virtual and logical “things” in a building which produce and consume data. Examples of physical entities are sensors, equipment, devices, rooms and pipes. Examples of virtual entities are setpoints, commands, alarms and other digital registers. Examples of logical entities are zones, groups and other collections of entities. The identification of these entities constitutes the central part of the graph in Figure 4. The other parts of the graph contain different perspectives on those entities.

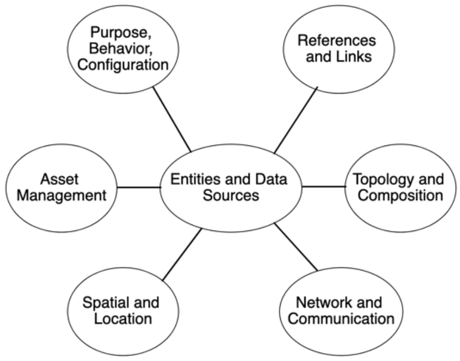

Fig. 4 – Categories of metadata required to support data-driven smart buildings. Circles represent ontology-based graphs.

Due to the diverse needs of applications [38], it is important that the graph provide multiple ways of identifying and understanding each entity. For example, FDD applications will require more *asset management* metadata to determine what kind of models and rules to use for each piece of equipment. FDD applications concerned with the health of the communication network will use *network and communication* metadata to diagnose connectivity issues. After faults are detected, an FDD application can leverage *spatial and location* metadata to tell a human operator where the broken device is. In contrast, MPC applications will require *topology and composition* and *purpose, behaviour and configuration* metadata to properly construct and configure the models of the system that allow optimal control sequences to be computed and executed. In each of these applications, *references and links* metadata (top layer in Fig. 3) provides the link between the representation of entities in the graph and how applications would interact with those entities: where historical data is, how to control the device's behaviour with an API, and so on.

5 Conclusion

The digital built environment transforms into a smart digital built environment. This leads to an abundance of data, algorithms and systems being commonly available for the creation of data-driven smart buildings. In recent years, several ontologies and metadata schemas have been proposed for supporting these smart buildings. In this paper, we reviewed mainstream approaches in that regard. While previous research focused a lot on schemas and vocabularies, this paper proceeded towards localising these ontologies in a draft reference system architecture. This draft architecture consists of three tiers (applications, systems and services, hardware abstraction). The systems and services layers has been detailed further in this paper, as a combination of different types of databases that is accompanied by a solid metadata infrastructure based on BRICK and LBD ontologies. The value of such a system lies primarily in FDD and asset management applications, yet, can be a solid basis also for MPC. Future work will focus on stress-testing the proposed architecture in multiple cases and examples internationally.

6. References

[1] Pritoni M, Paine D, Fierro G, Mosiman C, Poplawski M, Saha A, Bender J, Granderson J. Metadata Schemas and Ontologies for Building Energy Applications: A Critical Review and Use Case Analysis. Energies. 2021; 14(7):2024.

[2] Petrova E, Pauwels P. Semantic Enrichment of Association Rules Discovered in Operational Building Data for Reuse of Building Performance Patterns. In Proceedings of the 37th International Conference of CIB W78. Sao Paulo. 2020. p. 308-326.

[3] Xie X, Moretti N, Merino J, Chang J Y, Parlikad A K. Ontology-Based Spatial and System Hierarchies Federation for Fine-Grained Building Energy Analysis. Proceedings of the 38th International Conference of CIB W78, Luxembourg, 2021, pp 368-377.

[4] Pritoni M, Weyandt C, Carter D, Elliott J. Towards a Scalable Model for Smart Buildings. 2018. ACEEE Summer Study on Energy Efficiency in Buildings.

[5] Balaji B, Bhattacharya A, Fierro G, Gao J, Gluck J, Hong D, Johansen A, Koh J, Ploennigs J, Agarwal Y, Berges M, Culler D, Gupta R, Kjærgaard M B, Srivastava M, Whitehouse K. Brick: Towards a Unified Metadata Schema For Buildings. In Proceedings of the 3rd ACM International Conference on Systems for Energy-Efficient Built Environments (BuildSys 2016). Association for Computing Machinery, pp. 41–50.

[6] Linked Building Data Community Group. W3C Community Group Web Portal. https://www.w3.org/community/lbd/.

[7] Pauwels P, Costin A, Rasmussen M H. Knowledge Graphs and Linked Data for the Built Environment. 2022. Book Chapter in M. Bolpagni et al. (eds.), Industry 4.0 for the Built Environment, Structural Integrity 20. pp. 157-183.

[8] BuildingSMART. Industry Foundation Classes Specifications Database. https://technical.buildingsmart.org/standards/ifc/ifc-schema-specifications/.

[9] Pauwels P, Terkaj W. EXPRESS to OWL for construction industry: towards a recommendable and usable ifcOWL ontology. Automation in Construction. 2016;63:100-133.

[10] Rasmussen MH, Lefrançois M, Schneider GF,

Pauwels P. BOT: The building topology ontology of the W3C linked building data group. Semantic Web. 2021;12(1):143-161.
[11] BEO – Building Element Ontology. https://pi.pauwel.be/voc/buildingelement#.
[12] Wagner A, Rüppel U. BPO: The building product ontology for assembled products. In Proceedings of the 7th Linked Data in Architecture and Construction Workshop. 2019. CEUR Workshop Proceedings (Vol. 2389, pp. 106–119). https://ceur-ws.org/Vol-2389/08paper.pdf
[13] MEP – MEP ontology. https://pi.pauwel.be/voc/distributionelement#
[14] Rasmussen M H, Lefrancois M, Bonduel M, Hviid C A, Karlshoj J. OPM: An ontology for describing properties that evolve over time. In Proceedings of the 6th Linked Data in Architecture and Construction Workshop. 2018. CEUR Workshop Proceedings (Vol. 2159, pp. 24–33). https://ceur-ws.org/Vol-2159/03paper.pdf
[15] Bonduel M, Wagner A, Pauwels P, Vergauwen M, Klein R. Including widespread geometry formats in semantic graphs using RDF literals. In: Proceedings of the 2019 European Conference for Computing in Construction. European Council on Computing in Construction; 2019. p. 341–50.
[16] Bonduel M. The FOG ontology - File Ontology for Geometry formats. https://w3id.org/fog#
[17] Wagner A. Ontology for Managing Geometry (OMG). https://w3id.org/omg#.
[18] Garcia-Castro R, Poveda-Villalon M. SAREF extension for building (SAREF4BLDG). 2020. https://saref.etsi.org/saref4bldg/v1.1.2/
[19] Hamdan A-H, Bonduel M., Scherer R. An ontological model for the representation of damage to constructions. In Proceedings of the 7th Linked Data in Architecture and Construction Workshop. 2019. CEUR Workshop Proceedings (Vol. 2389, pp. 64–77). http://ceur-ws.org/Vol-2389/05paper.pdf
[20] RealEstateCore ontology. https://www.realestatecore.io/
[21] Brick consortium, Brick: a uniform metadata schema for buildings. https://brickschema.org/.
[22] Project Haystack. Ontology. https://project-haystack.org/doc/docHaystack/Ontology
[23] Fierro G, Koh J, Agarwal Y, Gupta R K, Culler D E. Beyond a House of Sticks: Formalizing Metadata Tags with Brick. In Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BuildSys '19). Association for Computing Machinery, New York, NY, USA, 125–134.
[24] Kukkonen V, Kücükavci A, Seidenschnur M, Rasmussen M H, Smith K M, Hviid C A. An ontology to support flow system descriptions from design to operation of buildings. Automation in Construction. 2022; 134:104067.
[25] SkyFoundry. SkySpark. https://skyfoundry.com/
[26] Google. Google Digital Buildings. https://github.com/google/digitalbuildings
[27] Microsoft. What is Azure Digital Twins? 2021. https://docs.microsoft.com/en-us/azure/digital-twins/overview
[28] Fierro G, Pritoni M, Abdelbaky M, Lengyel D, Leyden J, Prakash A, Gupta P, Raftery P, Peffer T, Thomson G, Culler D E. Mortar: An Open Testbed for Portable Building Analytics. ACM Trans. Sen. Netw. 2020; 16(1); pp. 1-31.
[29] Dawson-Haggerty S, Krioukov A, Taneja J, Karandikar S, Fierro G, Kitaev N, Culler D E. BOSS: Building Operation System Services. In Proceedings of the 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 13). 2013. Pp. 443-457.
[30] Malcolm A, Werbrouck J, Pauwels P. LBD server: Visualising Building Graphs in web-based environments using semantic graphs and glTF-models. In Eloy S, Leite Viana D, Morais F, editors, Formal Methods in Architecture: Proceedings of the 5th International Symposium on Formal Methods in Architecture (5FMA), Lisbon 2020. Cham: Springer. 2021. p. 287-293.
[31] Werbrouck J, Pauwels P, Beetz J, van Berlo L. Towards a decentralised common data environment using linked building data and the solid ecosystem. In: Kumar B, Rahimian F, Greenwood D, Hartmann T, editors: Proceedings of the 36th CIB W78 2019 Conference. 2019. p. 113–23
[32] Schneider GF, Rasmussen MH, Bonsma P, Oraskari J, Pauwels P. Linked building data for modular building information modelling of a smart home. In: Karlshoj J, Scherer R, editors. eWork and eBusiness in Architecture, Engineering and Construction. CRC Press; 2018. p. 407–414.
[33] Solid. Solid: your data, your choice. https://solidproject.org/
[34] Manola F, Miller E, McBride B. RDF 1.1 Primer. W3C Working Group Note. 24 June 2014. http://www.w3.org/TR/rdf11-primer/
[35] Pauwels P, Terkaj W, Krijnen T F, Beetz J. Coping with lists in the ifcOWL ontology. In Proceedings of the 22nd EG-ICE Workshop 2015, Eindhoven, The Netherlands (pp. 111-120).
[36] Terkaj W, Schneider GF, Pauwels P. Reusing domain ontologies in linked building data : the case of building automation and control. In: Proceedings of the Joint Ontology Workshops 2017 Episode 3: The Tyrolean Autumn of Ontology. 2017.
[37] Drgoňa J, Arroyo J, Figueroa I C, Blum D, Arendt K, Kim D, Perarnau Ollé E, Oravec J, Wetter M, Vrabie D L, Helsen L. All you need to know about model predictive control for buildings. Annual Reviews in Control. 2020; 50; pp. 190-232.
[38] Bhattacharya A, Ploennigs J, Culler D E. 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 (BuildSys '15). Association for Computing Machinery, New York, NY, USA, 33–34.