

GRAPH-BASED DIGITAL DECISION SUPPORT SYSTEMS: INTRODUCING BTWIN, A TOOLKIT FOR PERFORMANCE-BASED BUILDING MANAGEMENT

Angelo Massafra¹, Ugo Maria Coraglia¹, Giorgia Predari¹, Riccardo Gulli¹

¹Alma Mater Studiorum University of Bologna, Bologna, Italy

Abstract

The digitization of the AECO industry is increasing the need for digital decision-support systems (DDSS) in building management. Despite technological progress, developing such systems still demands significant resources. In particular, integrating diverse building data remains a challenge. Addressing this, the paper introduces BTwin, a toolkit designed to simplify the prototype development of DDSSs supporting decision-making in performance-based building management. This software library facilitates linking data from multiple sources into graph networks, like building information models and time series databases, providing a low-code and open-source framework for creating user-friendly web applications using connector logic.

Introduction

The built environment strongly impacts global energy consumption and produces high environmental emissions. Although it sits at the crossroads of many EU policies, it is still affected by a persistent challenge of limited innovation, which is probably the primary impediment to its digital and ecological transformation.

To face these challenges, in recent years, the Architecture, Engineering, Construction and Operation (AECO) sector has started its digital transition, pushed by the emergence of cutting-edge technologies, including Artificial Intelligence (AI) (Zhang et al., 2022), Building Information Modeling (BIM) (Deng et al., 2021), Internet of Things (IoT) (Mannino et al., 2021), Building Performance Simulation (BPS) (de Wilde, 2023), and Digital Twins (DT) (Lu, 2022). Notably, DTs have risen prominently, serving as a catalyst for developing digital decision support systems (DDSS) to enhance decision-making in built asset management, leading to the spread of performance-oriented applications for multiple uses, such as space usage monitoring, analysis of energy needs, reporting of energy and water consumption, estimation of emissions due to building usage, as well as analysis of indoor environmental quality regarding thermal, visual, acoustic comfort, and air ventilation (Petri et al., 2023).

Despite their rapid growth, DTs are still in their infancy and require, on the one hand, prototyping and technical development and, on the other, further conceptualization (Boje et al., 2020). The lack of standardized definitions, the heterogeneous and vast amount of data and data

sources, and the not scientific approach that usually characterizes built asset management (Abuimara et al., 2021) hinder the organic development of such technology and, more in general, of DDSSs, limiting the possibility of easily transforming building data in usable knowledge.

Developing such systems poses a resource-intensive challenge, with economic, financial, and cultural benefits that are sometimes difficult to comprehend during the conceptualization phase (Ma et al., 2022). These challenges are highlighted by the AECO sector's lack of data integration, marked by the presence of numerous data silos belonging to various stakeholders and created using distinct languages and frameworks, leading to significant data interoperability issues (Merino et al., 2022). These digital solutions require high expertise from many domains to enable secure and streamlined data flow, reliable and scientifically valid information generation, and user-friendly, accessible and usable knowledge to non-digital experts. The scientific community, therefore, is called first to theorize these systems and second to prototype solutions demonstrating the tangible benefits of such technologies.

In order to respond to these needs, this paper introduces BTwin, a software toolkit designed for simple and cost-effective prototyping of DDSSs for building management. The toolkit allows for semantically linking building data from diverse sources (like BIM, BPS, meters, and sensors) within graph networks and visualizing it organically through interactive dashboards, offering a low-code and open-source solution for prototyping web-based DDSSs. Implemented on JSON-LD (JavaScript Object Notation for Linked Data) principles, BTwin structures data within graphs by following the axioms and meanings defined in a federated ontology. After linking data, the toolkit enables the creation of web applications that extract data from graphs using user-executable queries through a graphical user interface (GUI). It then presents this data in various formats, informing the application user about the key performance indicators (KPIs) that characterize building performance. The software library performs these tasks in Python, a programming language known for its simple syntax, making it extremely popular for various software projects.

In this article, after discussing the state of the art in developing graph-based DDSSs in the AECO sector, the theoretical framework that supported the development of the tool is presented. Furthermore, the toolkit is applied to

a simplified case study to demonstrate its applicability and provide some examples of its use.

Related work and background

Knowledge graphs

When people express their thoughts to other people in the places where they work, they, consciously or unconsciously, create graphs. It means identifying specific objects and labelling the relationships between them to explain a problem, share a concept, or explain a domain, in other words, to transfer knowledge.

Such graphs can be produced either by humans or computers. Those intelligible to computers typically comprise nodes (or vertices), denoting entities and subjects, alongside edges representing the connections between these nodes (relationships or links). Both nodes and edges can be characterized by semantic attributions and property descriptions (Paulheim, 2016).

Accordingly, organizing domain concepts and information in knowledge graphs (KGs) allows for, first, assigning semantic and ontological meaning to data, as well as enabling semantic query, extraction and analysis of data from complex data structures; second, proficient visualization and, therefore, comprehension of a knowledge domain and its segmentation into knowledge sub-domains.

Beyond these capabilities, KGs also excel in analyzing extensive datasets entrenched in diverse formats. Moreover, if coupled with semantic web technologies (Pauwels et al., 2017), KGs can allow fast accessibility to information on the web, an issue that has become a priority for modern DDSSs.

Ontologies and schemas for data representation

Understanding the terminology used in a KG is crucial for systematically organizing and sharing the knowledge it contains in a universally comprehensible manner. In KGs, organic information description can be achieved through ontologies, formal, shared, and explicit representations of a domain's conceptualization, crafted to ensure that all information is clearly defined for logical reasoning.

Recently, the surge in digital tools within the construction sector has led to the definition of various ontologies to describe the AECO world and its subdomains (Pritoni et al., 2021). Among the most used ontologies and data schemas, the following are considered relevant to this research: Industry Foundation Classes (IFC); Building Topology Ontology (BOT); Brick Schema; Semantic Sensor Network Ontology (SSN); the Sensor, Observation, Sample, and Actuator (SOSA) ontology, and the Energy Management Key Performance Indicator Ontology (EKO).

IFC, the open BIM standard supported by buildingSMART, is the most recognized schema for BIM data. Its integration with semantic web technologies is allowed by the IFC web ontology language (ifcOWL) (buildingSMART International, n.d.). The Building

Topology Ontology (BOT) is a simplified ontology proposed by W3C that exclusively addresses the core building concepts revolving around the building's topology, including physical and conceptual components and their relationships (Rasmussen et al., 2017). Brick is an open-source schema that standardizes the semantic descriptions of buildings' physical, logical, and system assets and interrelationships. Its primary focus is on smart building applications and equipment representation (Balaji et al., 2018). SSN describes sensors and their observations, features of interest and samples, procedures, and actuators. It is often used to describe BAS data with semantic tags (Compton et al., 2012). SOSA, instead, redesigns SSN to provide a lightweight, general-purpose specification for modeling the interactions between entities involved in observations, actuation, and sampling (Janowicz et al., 2018). Finally, EKO enhances multi-level energy management and energy performance information exchange (Li et al., 2019).

Graph-based approaches for digital twins

In the contemporary landscape, graphs have assumed a pivotal role across many data-driven applications, also influencing the AECO sector (Lygerakis et al., 2022).

Various researchers have introduced and tested the use of KGs to develop digital systems for building operations, contributing to forming a solid approach to the subject, although relying on various technologies and data schemas. For instance, Merino et al. experimented with using KGs to deliver the DT of an educational building aimed at fault detection and diagnosis of HVAC systems for facility management purposes (Merino et al., 2022). In their study, these authors mainly leveraged IFC and Brick schemas for data representation and a data lake architecture for data storage. Chamari and colleagues, instead, explored the integration of BMS, IoT, and BIM via KGs, utilizing IFC and Brick within RDF graphs (Chamari et al., 2022). They proposed a hybrid approach for managing time series data, which was stored in a JSON-based MongoDB to overcome the limitations of graphs in handling time series with millions of data points. Differently, Hosamo et al. conducted interesting work on DT for fault detection and comfort optimization, integrating BIM, sensor time series, and maintenance records. Although the research refers to BOT, SSN, and Brick schemas, they used a custom ETL (Extract, Transform, Load) process to enable condition monitoring and predictive operation (Hosamo et al., 2023). Another significant example is the work of Ramonell and his team, who proposed a system for data integration within DTs, leveraging property graphs in Neo4j (Ramonell et al., 2023). This approach allowed for mapping semantic information according to different ontologies, keeping the graph as the interoperability backbone of the systems.

All these contributions, which propose hybrid strategies to enhance BIM-based KGs with time-series data, underscore the significance and utility of such technologies in modeling building knowledge,

highlighting both potentials and limitations. However, an aspect not covered in the mentioned articles is the Return on Investment (ROI) for developing these applications. Indeed, it remains a challenge to ascertain whether the benefits of developing such complex systems can be financially justified for the administrations that adopt them, especially considering that substantial funding programs are usually behind these research initiatives. Consequently, leveraging the theoretical frameworks established in these studies and the insights derived from these pioneering explorations, this paper seeks to introduce a toolkit designed to streamline the initial prototyping process of such digital environments. The goal is to make the process more cost-effective and accessible to research and development groups of smaller scale or with limited resources, thus promoting wider adoption and innovation in this field.

Materials and methods

Ontology definition

BTwin assumes that a building can be represented as a graph composed of nodes (subjects) and edges (predicates). To define organically information, nodes are designated to classes (and subclasses), whereas edges are associated with types of relationships, following the ontology depicted in Figure 1. This federated ontology, designed by leveraging BOT, IfcOWL, Brick, SSN, SOSA, and EKO, integrates these already established ontologies within a unified and extensible ontological model.

At the foundational level, the federated ontology employs the BOT's zones (i.e., 'bot:Building', 'bot:Site', 'bot:Storey', and 'bot:Space'). These objects establish a hierarchy necessary for representing the building's spatial configuration. BOT serves as the backbone, allowing for the detailed specification of physical locations and the

inclusion of elements (such as sensors) within spaces. BOT is also used to model 'bot:Interface' elements resembling interface elements bounding the spaces (e.g. walls, floors, roofs, windows and doors).

Interlaced with BOT is IfcOWL, which is used in the ontology to map the properties of spatial elements as 'ifc:Property' within 'ifc:PropertySet', instrumental in providing detailed descriptions of element properties and their attributes. Moreover, IfcOWL allows for the representation of sensor elements, considered as physical electronic devices, using the 'ifc:Sensor' class.

Enhancing the ontology is Brick. The Brick framework enables classifying sensor elements as 'brick:Sensor'. While there seems to be an overlap with 'ifc:Sensor', Brick sensors are denoted as 'brick:Point', which are input points symbolizing the value captured by a device or instrument engineered to detect and measure various variables. This indicates that a single IfcSensor device can embody multiple points (thanks to the 'brick: hasPoint' relationship). For instance, a 'brick:HumiditySensor' and a 'brick:TemperatureSensor' points may continuously provide data to an 'ifc:Sensor' device. The link between sensors and spaces is given by the relationship 'brick:hasLocation'. Additionally, Brick facilitates the organization of spaces into zones utilizing the 'brick:Zone' class, which can encapsulate groupings of rooms such as energy zones or fire compartments. Furthermore, the relationship 'brick:hasLocation' establishes semantic interoperability with the BOT spatial hierarchy, thereby enriching the data model with spatial context and associations.

The ontology further incorporates the SSN and SOSA ontologies through the 'sosa:Observation' and 'sosa:ObservableProperty' classes. These are pivotal for modeling the data generated by sensors (thanks to the

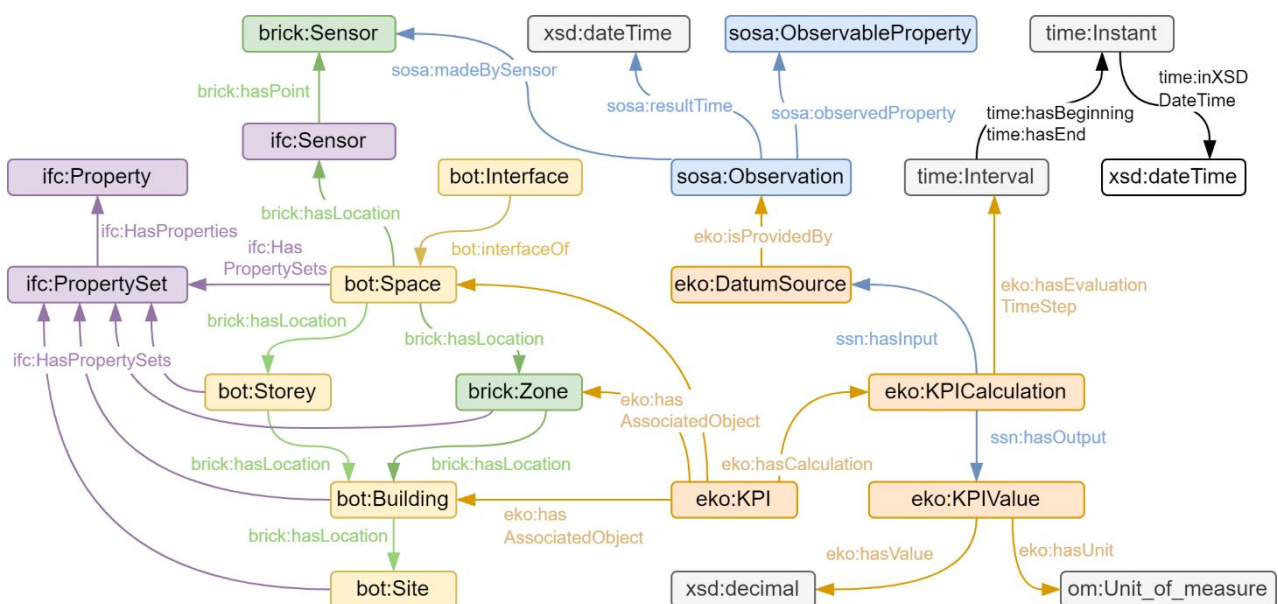


Figure 1: The federated ontology supported by BTwin.

'sosa: madeBySensor' relationship) and the observational processes in a semantically rich context.

EKO, which includes classes such as 'eko:KPI', 'eko:KPICalculation', and 'eko:KPIValue', is instead used to describe the performance aspects of spatial elements. These classes are used to define, calculate, and store the values of performance metrics, which are crucial for evaluating building performance. The 'eko:hasAssociatedObject' relationship provides the necessary linkages between performance-related and spatial concepts. Additionally, this part of the ontology integrates time-related classes, such as 'time:Instant' and 'time:Interval', to represent temporal aspects, which are essential for capturing the dynamics of sensor performance data and observations over time.

Data format

The federated ontology is enhanced with a distinct syntax employing JSON-LD to illustrate entities and their interconnections in graphs. JSON-LD is a streamlined format for Linked Data, leveraging the prevalent JSON structure to enable JSON data to function seamlessly globally. It significantly improves the ease with which data can be read and written by humans, offering a clear advantage over other formats like the EXPRESS format used in IFC. Due to its extensibility and adaptability, this data model allows the serialization of data and the creation of graph-based data models. Additionally, it is lightweight, rendering it suitable for data exchange within web environments and the development of web APIs and applications.

Specifically, BTwin adopts the JSON-LD format to represent static elements, whereas it sources dynamic data, like sensor observations, from databases tailored for time-series data (e.g., SQL or MongoDB), which are more adept at managing large data volumes than graph databases. Within the JSON-LD data format, each building element corresponds to a JSON object. This JSON object functions as a dictionary, possessing a unique identifier, the indication of its class within the ontology, and the relationships with the other objects within the KG. In BTwin, this data format can be read and transformed into a KG using specific functions that leverage the NetworkX library, a Python package that provides a comprehensive set of tools for creating, manipulating, and analyzing graph networks.

Figure 2 presents an example of a JSON-LD syntactic description in BTwin. The 'context' section enables the mapping of terms used in the document to Internationalized Resource Identifiers (IRIs) to provide precise meanings. In this part, reference ontologies (such as Brick and BOT) are mentioned, as well as the keys used in various JSON dictionaries to describe the objects. On the other hand, the 'graph' section describes the nodes that form the graph and maps the existing relationships between them ('relationships' keys).

The example in Figure 2 shows the topological structure of a sample building through BOT. This structure

comprises <storeyX>, defined as a 'bot:Storey' and representing a level within the building, and by <space1>, which is a 'bot:Space' located on <storeyX>. The Brick Schema is employed to standardize descriptions of sensor data, with <sensor1> being an 'ifc:Sensor' situated in <space1>, and having a temperature 'brick:Point' connected to <temperaturePoint1>. Moreover, IFC is used to model the properties of spatial entities, i.e., <PSet_SpaceThermalRequirements1> is depicted as a set of thermal requirements for <space1>, including a property about the temperature setpoint for winter. The interconnections between the elements, given by the 'relationships' key, are essential for decoding the spatial hierarchy and placing sensors within the building.

```
{
  "@context": {
    "bot": "https://w3id.org/bot#",
    "brick": "https://brickschema.org/schema/Brick#",
    "ifc": "https://standards.buildingsmart.org/.../OWL#",
    "name": "http://schema.org/name"
  },
  "@graph": [
    {
      "@id": "storeyX",
      "@type": "bot:Storey"
    },
    {
      "@id": "space1",
      "@type": "bot:Space",
      "relationships": {
        "brick:hasLocation": [
          {
            "@id": "storeyX", "@type": "bot:Storey"
          }
        ]
      }
    },
    {
      "@id": "sensor1",
      "@type": "ifc:Sensor",
      "relationships": {
        "brick:hasLocation": [
          {
            "@id": "space1", "@type": "bot:Space"
          }
        ]
      }
    },
    {
      "@id": "temperaturePoint1",
      "@type": "brick:Sensor",
      "relationships": {
        "brick:isPointOf": [
          {
            "@id": "sensor1", "@type": "brick:Sensor"
          }
        ]
      }
    },
    {
      "@id": "PSet_SpaceThermalRequirements1",
      "@type": "ifc:IfcPropertySet",
      "name": "PSet_SpaceThermalRequirements",
      "relationships": {
        "ifc:IfcPropertySetDefinition": [
          {
            "@id": "space1",
            "@type": "bot:Space"
          }
        ]
      },
      "ifc:hasProperties": [
        {
          "@type": "ifc:IfcPropertySingleValue",
          "name": "SpaceTemperatureWinterMin",
          "nominalValue": {
            "type": "ifc:IfcThermodynamicTemperatureMeasure",
            "value": 20
          }
        }
      ]
    }
  ]
}
```

Figure 2. Example JSON-LD representation.

Toolkit architecture

BTwin consists of a Python library dependent on other reference libraries widely supported by the programming community. Leveraging the principles of linked data and using a hybrid approach for integrating KGs with time-series data allows for fast prototyping of DDSS dashboards. A

s stated, this software library employs a data format based on JSON-LD. This format can be parsed and converted into network graphs by relying on specific functions integrated with the NetworkX library.

Using the so-called 'connector' logic, the tool organically integrates with external libraries and BIM formats, promoting scalability and modularity in the data modeling steps. Alternatively, for simpler cases, static graph data can be modeled directly within BTwin without the use of third-party software.

Table 1 below shows the alignment of the federated ontology adopted by BTwin with some of the classes of the BIM formats supported by the toolkit: IFC, Autodesk Revit, Topologic, and EnergyPlus' Input Data Format (IDF). The representation of such formats in KGs is obtained through the use of connectors that rely on IfcOpenShell, Autodesk Revit APIs, Topologicpy, and Eppy, respectively.

For example, the connector with Topologic allows the representation of Topologic elements directly within graphs (e.g., Topologic cells are transformed into nodes labelled as 'bot:Space'). Similarly, the connector with EnergyPlus, developed in BTwin by integrating the APIs of the Eppy library, enables the linking of simulation results to the spatial elements whose behavior is simulated (provided that the zones in the IDF have the same name as the zones in the KG), as well as the calculation of energy-related KPIs. Additional connectors allow for interfacing with Autodesk Revit, using the APIs provided by Autodesk, and IFC, using IfcOpenShell. These connectors are useful for extracting the properties of spatial elements within BIM models and associating them with the graph nodes. Furthermore, compatibility with Arduino for modeling sensor data has been tested, and preliminary experiments have been conducted with ChatGPT APIs to enable the extraction of information from graphs using textual prompts and interpreting it through OpenAI's Large Language Models (LLMs).

For better computational efficiency, dynamic data, such as sensor readings, are not stored in graphs but in more streamlined formats within dedicated databases, such as SQL. These databases employ a straightforward schema that includes only the sensor's unique identifier, the timestamp, and the recorded value. The connection to the time-series data is established via a link to the brick:Sensor's UID. An illustrative example of tabular time series data is depicted in Table 2. This approach enables time-series databases to handle and store many records efficiently. Meanwhile, the graph database maintains relationships among static elements separately,

allowing for the processing of time-series data solely when computing KPIs.

Once the static and dynamic data are interconnected using this hybrid approach, the toolkit enables various functions for querying, aggregating, processing, and visualizing the data stored within the KG. In particular, the visualization module relies on Plotly and Dash libraries for the creation of web dashboards. The results are web micro-applications that can be easily published on the Internet using solutions such as Docker and AWS, following, for instance, methods similar to the one demonstrated by (Betti et al., 2022).

Table 1: BIM formats supported by BTwin and class alignment.

BTwin	IFC	Autodesk Revit	Topologic	Energy Plus
bot: Building	IfcBuilding	Building	Cell Complex	Building
bot: Storey	IfcBuilding Storey	Level	-	-
bot:Space	IfcSpace	Space	Cell	Space
brick: Zone	IfcZone	Zone	Cluster	Zone
bot: Interface	IfcWall, IfcRoof, IfcSlab, IfcDoor, IfcWindow	Wall, Roof, Floor, Door, Window	Face, Aperture	Surface

Table 2: Example of a time-series data

Timestamp	Value	brick:Point UID	Unit
2023-11-15 10:00	18.3	tempPointX	°C
2023-11-15 10:15	18.6	tempPointX	°C
2023-11-15 10:30	18.8	tempPointX	°C

Demonstration

This section applies the toolkit to develop a sample application for monitoring indoor environmental quality parameters, namely temperature, humidity, and illuminance. For illustrative purposes, a fictional case study is employed to demonstrate a potential workflow. This hypothetical building comprises four spaces: three square offices, each with dimensions of 5x5 meters and a height of 3 meters, and an adjacent corridor measuring 15x2 meters with the same height. This setup creates a data-sharing environment that, through a user-friendly interface, enables the evaluation of the number of hours during which the setpoints for temperature, humidity, and

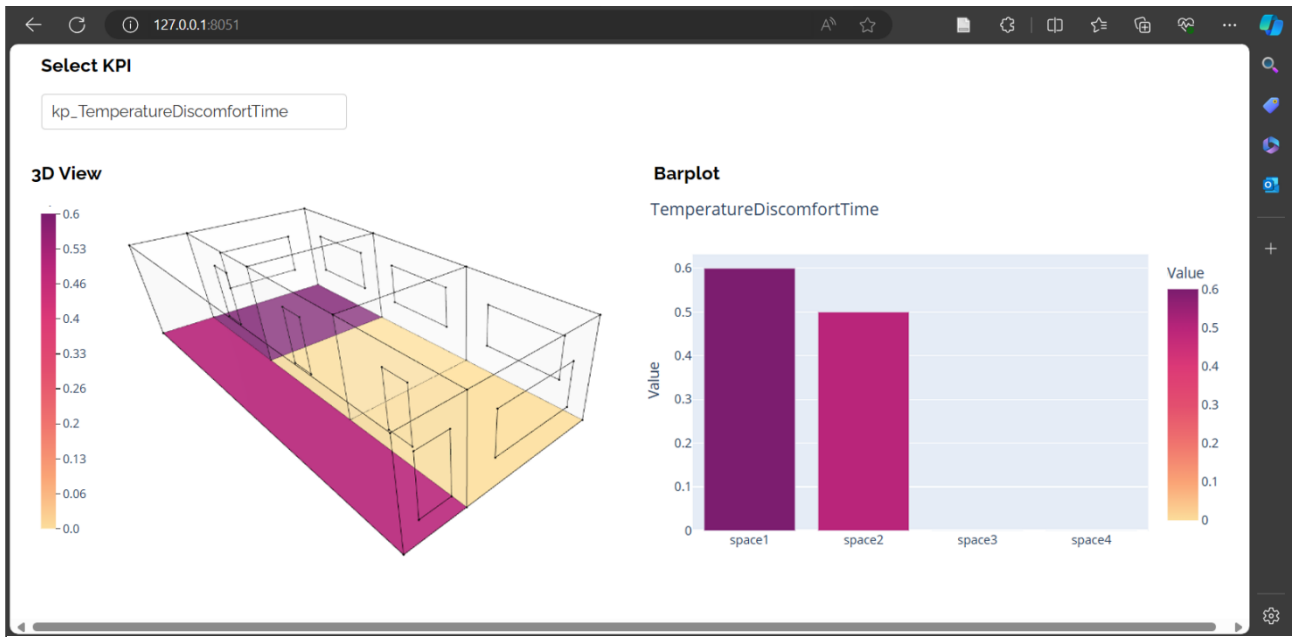


Figure 3: A dashboard developed for the sample application.

lighting are not met in these four spaces. The assessment is based on fictional measurements recorded on a winter day, as shown in Figure 3.

The architecture of the prototype DDSS is shown in Figure 4, while the main steps followed for the application's implementation are listed below.

1. First, the spatial hierarchy of the building was modeled. In particular, Topologic was used to create the case study's BIM model and the connector between Topologic and BTwin was used to transform it into JSON-LD format. The Topologic model also served for 3D visualization within Plotly.
2. Subsequently, sensor elements were added to the JSON-LD as 'ifc:Sensor' and 'brick:Sensor' elements. Moreover, relationships between these and the spatial elements were established.
3. Once all the spatial and sensor elements were identified, the properties of the spatial elements were added to the JSON-LD document. For each space, attributes related to target values of environmental comfort, defined in the KG as properties of the spaces grouped within property sets, were added. The values of the added properties were set, for example, at a minimum value of 20°C for temperature, a range between 30% and 60% for humidity, and a minimum value of 250 lux for lighting.
4. Thanks to NetworkX's connector, a KG containing the spatial elements, sensors, and property sets was created starting from the JSON-LD data. It is depicted in Figure 5.
5. Then, the data related to the measurements taken by the sensors was formatted in a MySQL database according to the notation in Table 2. For each space, temperature, humidity, and illuminance

observations were recorded once per hour over a 10-hour working day in winter, resulting in 120 records.

6. Information was then extracted from the KG and MySQL to calculate the internal environmental comfort KPIs, space by space. These KPIs indicate the number of hours in which the target comfort values were not met. After being calculated, such KPIs were connected to the node representing the evaluated space in the KG for each space.
7. Thanks to Plotly's connector, a dashboard was created through Dash to display the KPIs on the geometries of the building's 3D model, previously created in Topologic, within graphs and tables (Figure 3).

From a methodological point of view, the following process can be repeated for different information uses, such as analyzing the energy needs of spaces, detecting indoor air quality, monitoring space usage, and so on.

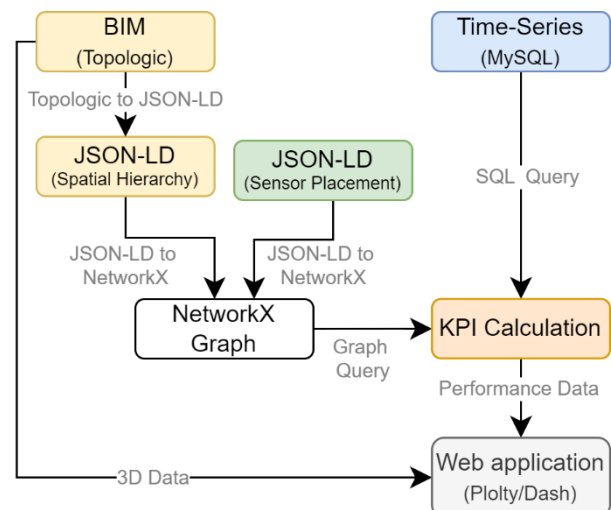


Figure 4. The system architecture of the prototype DDSS.



Figure 5: Knowledge graph developed for the sample application.

Conclusion

The AECO sector is currently experiencing a digital revolution, resulting in an increased need for instruments that facilitate decision-making in the realm of building management. Despite technological strides, a scarcity of these tools persists, primarily due to the substantial technical and economic investments required for their creation.

To provide a solution for streamlining the prototyping process of web-based DDSSs and address the gap in data integration in the AECO sector, this article presented BTwin, a software library designed for low-code development of DDSSs aimed at building performance-based management. The article, serving as a whitepaper for BTwin, provided the theoretical background of the toolkit, describing the axioms, concepts, and principles on which it is based. Additionally, it offered demonstrations using a simple fictional case study building to illustrate an example of its application.

This open-source toolkit, developed using a JSON-LD format and several Python functions, features a range of methods designed for semantically integrating building data from various sources, such as BIM, BPS, meters, and sensors, within KGs. It also enables data visualization through interactive dashboards, facilitating the monitoring and analysis of building performance-related aspects in a user-friendly manner. This approach could enable developers to easily create DDSS prototypes, allowing stakeholders to quickly preview dashboards during the development processes of these digital systems and provide feedback from a user-centered design perspective, thus ensuring tools tailored to actual needs. Moreover, its human-readable design makes it valuable for students and learners in the fields of BIM, DT, and KGs.

The toolkit is currently in development, with plans to expand its capabilities. Future enhancements will involve the integration of additional connectors to align the JSON-

LD format with more complex data structures, such as gbXML (Green Building XML), to ensure compatibility with widely used BIM and web data exchange standards. Experiments will be conducted to utilize BTwin's KGs in more robust databases than NetworkX, such as Neo4j or MongoDB, and to incorporate real-time time-series databases. Moreover, special focus will be placed on refining the interfaces to simplify user access to data within the KG, enabling data queries through textual prompts. This feature will be developed utilizing Large LLMs like ChatGPT, which are experiencing rapid growth in the industry. Ultimately, future versions will demonstrate the tool's application in more complex real-world case studies, guiding the presented approach towards a DT perspective.

Acknowledgements

The publication was realized by a researcher with a research contract co-funded by the European Union - PON Research and Innovation 2014-2020 pursuant to Art. 24, paragraph 3, lett. a), of Law no. 240 of 30 December 2010 and subsequent amendments and Ministerial Decree no. 1062 of 10 August 2021.

References

- Abuimara, T., Hobson, B.W., Gunay, B., O'Brien, W., Kane, M., 2021. Current state and future challenges in building management: Practitioner interviews and a literature review. *Journal of Building Engineering* 41, 102803. <https://doi.org/10.1016/j.jobbe.2021.102803>
- Balaji, B., Bhattacharya, A., Fierro, G., Gao, J., Gluck, J., Hong, D., Johansen, A., Koh, J., Ploennigs, J., Agarwal, Y., Bergés, M., Culler, D., Gupta, R.K., Kjærgaard, M.B., Srivastava, M., Whitehouse, K., 2018. Brick: Metadata schema for portable smart building applications. *Applied Energy* 226, 1273–1292. <https://doi.org/10.1016/j.apenergy.2018.02.091>

- Betti, G., Tartarini, F., Nguyen, C., Schiavon, S., 2022. CBE Clima Tool: a free and open-source web application for climate analysis tailored to sustainable building design. *Automation in Construction* 114, 103179. <https://doi.org/10.48550/ARXIV.2212.04609>
- Boje, C., Guerriero, A., Kubicki, S., Rezgui, Y., 2020. Towards a semantic Construction Digital Twin: Directions for future research. *Automation in Construction* 114, 103179. <https://doi.org/10.1016/j.autcon.2020.103179>
- buildingSMART International, n.d. International Foundation Classes (IFC). Available online at: <https://technical.buildingsmart.org/standards/ifc> (Accessed 29/03/2024)
- Chamari, L., Petrova, E., Pauwels, P., 2022. A web-based approach to BMS, BIM and IoT integration. *CLIMA 2022 Conf. 2022: CLIMA 2022 The 14th REHVA HVAC World Congress*. <https://doi.org/10.34641/CLIMA.2022.228>
- Compton, M., Barnaghi, P., Bermudez, L., García-Castro, R., Corcho, O., Cox, S., Graybeal, J., Hauswirth, M., Henson, C., Herzog, A., Huang, V., Janowicz, K., Kelsey, W.D., Le Phuoc, D., Lefort, L., Leggieri, M., Neuhaus, H., Nikolov, A., Page, K., Passant, A., Sheth, A., Taylor, K., 2012. The SSN ontology of the W3C semantic sensor network incubator group. *Journal of Web Semantics* 17, 25–32. <https://doi.org/10.1016/j.websem.2012.05.003>
- de Wilde, P., 2023. Building performance simulation in the brave new world of artificial intelligence and digital twins: A systematic review. *Energy and Buildings* 292, 113171. <https://doi.org/10.1016/j.enbuild.2023.113171>
- Hosamo, H.H., Nielsen, H.K., Kraniotis, D., Svennevig, P.R., Svidt, K., 2023. Improving building occupant comfort through a digital twin approach: A Bayesian network model and predictive maintenance method. *Energy and Buildings* 288, 112992. <https://doi.org/10.1016/j.enbuild.2023.112992>
- Janowicz, K., Haller, A., Cox, S.J.D., Phuoc, D.L., Lefrançois, M., 2018. SOSA: A Lightweight Ontology for Sensors, Observations, Samples, and Actuators. <https://doi.org/10.48550/ARXIV.1805.09979>
- Li, Y., García-Castro, R., Mihindikulasooriya, N., O'Donnell, J., Vega-Sánchez, S., 2019. Enhancing energy management at district and building levels via an EM-KPI ontology. *Automation in Construction* 99, 152–167. <https://doi.org/10.1016/j.autcon.2018.12.010>
- Lu, Q., 2022. Digital twins in the built environment: fundamentals, principles and applications. ICE Publishing, London.
- Lygerakis, F., Kampelis, N., Kolokotsa, D., 2022. Knowledge Graphs' Ontologies and Applications for Energy Efficiency in Buildings: A Review. *Energies* 15, 7520. <https://doi.org/10.3390/en15207520>
- Ma, X., Li, X., Yuan, H., Huang, Z., Zhang, T., 2022. Justifying the Effective Use of Building Information Modeling (BIM) with Business Intelligence. *Buildings* 13, 87. <https://doi.org/10.3390/buildings13010087>
- Mannino, A., Dejacco, M.C., Re Cecconi, F., 2021. Building Information Modeling and Internet of Things Integration for Facility Management. *Literature Review and Future Needs. Applied Science* 11, 3062. <https://doi.org/10.3390/app11073062>
- Merino, J., Xie, X., Moretti, N., Chang, J.Y., Parlikad, A.K., 2022. Data integration for digital twins in the built environment. Presented at the 2022 European Conference on Computing in Construction. <https://doi.org/10.35490/EC3.2022.172>
- Paulheim, H., 2016. Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic Web* 8, 489–508. <https://doi.org/10.3233/SW-160218>
- Pauwels, P., Zhang, S., Lee, Y.-C., 2017. Semantic web technologies in AEC industry: A literature overview. *Automation in Construction* 73, 145–165. <https://doi.org/10.1016/j.autcon.2016.10.003>
- Petri, I., Rezgui, Y., Ghoroghi, A., Alzahrani, A., 2023. Digital twins for performance management in the built environment. *Journal of Industrial Information Integration* 33, 100445. <https://doi.org/10.1016/j.jii.2023.100445>
- Pritoni, M., Paine, D., Fierro, G., Mosiman, C., Poplawski, M., Saha, A., Bender, J., Granderson, J., 2021. Metadata Schemas and Ontologies for Building Energy Applications: A Critical Review and Use Case Analysis. *Energies* 14, 2024. <https://doi.org/10.3390/en14072024>
- Ramonell, C., Chacón, R., Posada, H., 2023. Knowledge graph-based data integration system for digital twins of built assets. *Automation in Construction* 156, 105109. <https://doi.org/10.1016/j.autcon.2023.105109>
- Rasmussen, M.H., Pauwels, P., Hviid, C.A., Karlshøj, J., 2017. Proposing a Central AEC Ontology That Allows for Domain Specific Extensions, in: *Lean and Computing in Construction Congress - Volume 1: Proceedings of the Joint Conference on Computing in Construction.*, pp. 237–244. <https://doi.org/10.24928/JC3-2017/0153>
- Zhang, F., Chan, A.P.C., Darko, A., Chen, Z., Li, D., 2022. Integrated applications of building information modeling and artificial intelligence techniques in the AEC/FM industry. *Automation in Construction* 139, 104289. <https://doi.org/10.1016/j.autcon.2022.104289>