A web-based approach to BMS, BIM and IoT integration: a case study

Lasitha Chamari, Ekaterina Petrova, Pieter Pauwels

Department of the Built Environment, Eindhoven University of Technology, Eindhoven, the Netherlands, l.c.rathnayaka.mudiyanselage@tue.nl, e.petrova@tue.nl, p.pauwels@tue.nl.

Abstract. Buildings are complex cyber-physical systems that rely on a combination of heterogeneous systems to provide smooth operation, energy efficiency, occupant comfort, well-being and safety. Building Management Systems (BMS) are central to these operations and generate a huge amount of data. Traditionally operated in a local server in a building, state-of-art BMS solutions are now moving towards the cloud. Internet of Things (IoT) meters and sensors are also increasingly used in buildings and hold tremendous potential for smart building monitoring and control. Another valuable data source is a properly developed and managed Building Information Model (BIM). BIM itself is moving towards Level 3, which is web-based and data-driven, as opposed to the file-based BIM of today. Improving building performance (e.g., energy, comfort, operational cost) relies on data from all the above systems. However, these data usually remain siloed within their own environment and do not provide an opportunity to perform evaluations across multiple systems. Moreover, valuable information about geometry, spatial location, and metadata about the building objects that the BIM models already contain remains unusable for building performance monitoring and reporting tools. Integrating these heterogeneous data sources will provide ample opportunities to improve building performance. Even though some commercial tools enable the integration of sensor data with BIM models, such tools remain largely proprietary and are not compatible with other applications. This study presents a methodology to integrate multiple information sources at their system level in a distributed manner. The Industry Foundation Class (IFC) model of the building is used as the primary source of information for creating a semantic building graph. Since the semantics of BMS sensors was not originally available in the IFC model, the Brick ontology is used to semantically describe BMS sensors in the graph. Sensor data related to spaces in the BIM model is visualized by selecting a space from the 3D model via the web application. Each data stream remains in its optimum environment and the connections are made via an Application Programming Interface (API).

Keywords. Data integration, BMS, IoT, BIM, Semantic Web, Brick
DOI: https://doi.org/10.34641/clima.2022.228

1 Introduction

Buildings are equipped with many systems such as HVAC, lighting, security, water, energy, etc. A building Management System (BMS) is used to centralize, automate and make the management of the above systems more efficient. By doing so, more efficient building operation is achieved at a reduced labour and energy cost, while ensuring a safe and comfortable environment [1]. Internet of Things (IoT) devices are also extensively used in many new buildings [2], giving rise to a large volume of real-time data about energy usage, ambient conditions (temperature, humidity, illuminance), as well as machine and equipment related parameters such as vibration, faults, alarms, etc. This data is used for reporting, monitoring, dashboards, fault detection, energy forecasting, etc. Smart buildings produce an abundance of such data, forming a virtual representation of the physical building, known today as a “Digital Twin” [3]. Other than the time series data, another invaluable source of information is the Building Information Model (BIM) [4] which is an essential component of the Architecture, Engineering and Construction (AEC) industry. A BIM model can be defined as the digital representation of a building that contains semantic information about the objects [5]. A properly developed and managed BIM model is a data source of high fidelity. It includes geometry, spatial location, and a broad representation of metadata about the properties of the building, its subsystems, devices, Mechanical, Electrical and Plumbing (MEP) equipment, etc. [6].

1.1 Data integration for smart buildings

To manage a building holistically and leverage the benefits of smart operation, various domains such as Supervisory Control and Data Acquisition (SCADA) systems, floor plans, IoT devices, 3D models need to be integrated. With the ever-increasing demand for smart building applications (Digital Twins, real-time monitoring, energy flexibility), there is a need to develop applications that depend on data available across multiple decentralized systems. This is where the importance of data integration arises. However, this integration is far from easy. There is no straightforward way, for example, to integrate a Honeywell BMS with a BIM model developed in Autodesk Revit, because they are incompatible by definition and rely on different modelling approaches, languages and protocols.

Therefore, a significant body of research in the AEC industry focuses on how to exchange and integrate data [3]. Many such attempts are revolving around Industry Foundation Classes (IFC), which is a vendor-neutral data model for exchange of AEC data [3] relying on the EXPRESS schema language. IFC is an open international standard for BIM data that is exchanged and shared among software applications used by the various participants in the construction and facility management sectors [7]. However, not all aspects of the built environment can be modelled in IFC. For example, a crucial part of the operational phase of a building is the dynamic sensor data, which cannot be easily represented using IFC. Even though the IFC standard describes many common Heating Ventilation & Air Conditioning (HVAC), lighting, and sensor devices, they are not capable of representing the context of the devices contained within [8].

1.2 Semantic web technologies

Semantic technologies and linked data have gained traction in the built environment. IFC itself is also available as ifcOWL ontology [9]. Linked data models are built using formal ontologies and thereby provide the opportunity for extending and linking with other domain-specific ontologies, exchanging heterogeneous information [10], and deriving new information based on the semantic graph [8]. Semantic graphs, also referred to as knowledge graphs, can also be queried using various query languages [9]. Linked Building Data (LBD) is an initiative with a focus on making building data web-ready [11]. A number of vocabularies and ontologies such as BOT, PRODUCT, and PROPS have been developed using a linked data approach [12]. Many other ontologies, such as SSN [13], RealEstateCore [14], Haystack [15], Brick [16], etc., have also emerged to fulfil various information requirements throughout the building lifecycle.

1.3 Current status of data integration in buildings

With such technologies available, one might anticipate a connected building with readily available data from multiple systems co-exising in a common platform. However, this is far from reality. Data is maintained in isolated silos and there is little interaction between different datasets. For example, BIM models developed in the design phase often remain disconnected from the operational data. Most of the sensors and devices are also not included in the BIM models at the design stage. Again, although the spatial information is already available in the BIM model, this information is duplicated in the BMS due to the lack of interoperability. Therefore, despite the number of systems or amount of data being collected, it is difficult to interpret information across domains due to little interaction between islands of data [17]. Available data, their formats, naming conventions, and standards also differ significantly among each building and vendor [18]. This absence of a common data collection structure leads to unstructured data that is difficult to discover, integrate, process and use. This has made the integration of data siloes an essential requirement in the built environment.

According to the literature [13, 14], a common use case is integrating sensor data into the BIM model using commercial BIM authoring tools like Revit or Navisworks. Although they may serve as appropriate tools for visualizing sensor data in a 2D or 3D environment, such approaches rely on vendor-specific software and do not provide reusable components for other applications. Alternative approaches rely on integrating datasets including building geometry, the relationship between spaces, sensors and actuators, and time series data into a single semantic graph in Resource Description Framework (RDF) using semantic web approaches [21]. Sensor data is further retrieved for visualization in charts and colour-coded maps in 2D plans. Other studies [16, 17] suggest that retaining sensor data in a database that is optimized to store time series data is more efficient. Apart from BIM and sensor data integration, another path is integrating BIM with Linked Building Data. Visualizing BIM models on the web provides a vendor-neutral platform for collaboration and exchange of data, as well as means to integrate data across domains [11]. As such, a web-based server is proposed in [11], which allows users to upload BIM models in IFC format, and visualise it in a graphical user interface. Attaching Linked Building Data (as graphs) to the project allows querying the BIM model based on the graph.

However, implementations integrating systems, BIM, sensor data, and semantic graphs in a decentralized manner are still in their infancy. Also, the built environment is not static. More and more energy-efficient buildings are demanded, and various sensors, devices and systems are continuously added to buildings. Therefore, the building information models also evolve. Data-driven smart building applications such as forecasting, fault detection and diagnosis, energy balancing, etc., are also increasingly deployed, and the output from these algorithms needs to be communicated to level building controllers, making the information flow bidirectional. Therefore, other essential components for future integration are the building controllers.

1.4 Objective and scope

This paper presents the first step towards integrating heterogeneous systems in a building. The study approaches multiple information sources at their system-level and demonstrates how to integrate them in a distributed manner. The proposed approach uses an IFC file to create the initial graph, followed by another graph to represent the semantics of BMS sensors. Each data stream remains in its optimum environment and the connections are made via Application Programming Interface (API), rather than file-based transfers. The developed platform can be used to collectively view and query the building data in its entirety. It also provides the opportunity for visualizing sensor data. Future work will address the integration of real-time data and bi-directional communication with building controllers.

2 Case study building

The use case building is a 12-storey renovated university building consisting of offices, study, and lecture rooms. It has been renovated in 2019 and contains a BMS, Energy Management System (EMS), and a Lighting Management system. The 4th to 11th floor of the building are considered a Living Lab and are used for research regarding indoor environmental quality, energy efficiency, lighting, Digital Twin implementation, etc.

2.1 BMS time series data

The campus building has a Honeywell BMS. A data dump from the BMS containing occupancy, temperature, and CO_2 data is used for this study. They constitute 1,314,720 data points per month. Measurements are stored on an hourly basis and the extracted data from the BMS is stored in xlsx format (**Tab. 1**). The table columns contain the timestamps and the sensor IDs. Mapping sensor IDs and their description are available as shown in **Tab. 2.** In **Tab. 2**, sensor location is included in the description. For example, sensor 11NR008TE-001TRL is in room no.128 of the 8th floor.

2.2 IoT time series data

Envision Manager is the API available to interact with the Lighting Management System for monitoring and controlling dimming levels and colour temperature. Furthermore, an ESP32-based IoT sensor network is in development to collect and display real-time temperature, humidity and illumination data. In this study, this sensor network is used to demonstrate how real-time sensor data can be integrated into the platform.

Tab. 1 – BMS data extraction.

Timestamp	11NR008TE-001TRL	11NR008TE-003TRL
28-02-2021 00:02:00	22.1	21.1
28-02-2021 01:02:00	22.0	20.9
28-02-2021 02:02:00	21.9	20.8
28-02-2021 03:02:00	21.7	21.2

Tab. 2 – BMS data point mapping table.

Item name	Description
11NR008TE-001TRL	Room temperature 8_128
11NR008QT-040CO2	*CO2 measurement 8_323
11NR008LT-001PIRTM	Presence 8_128

2.3 BIM model

The BIM model used for this study contains the 8th and 9th floors of the case study building. It is developed in Autodesk Revit and includes space-related information (room name, room number, and the floor) and equipment-related information (sensors and lighting fixtures) as shown in **Fig. 1.**

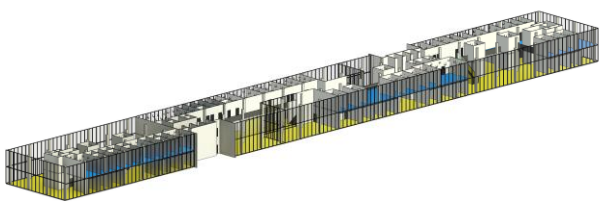

Fig. 1– BIM model developed in Revit

3 Implementation

This section describes the approach on how to integrate the various data sources identified above. In brief, that includes how to,

1. Use the IFC file as the data source for the 3D viewer, as well as the main source of building information (geometry, spatial information).
2. Convert the IFC file into a web browser compatible format.
3. Generate the semantic graph of the building using the IFC file. This carries information about different spaces and sensors/equipment contained within each space.
4. Use the BMS sensor mapping table to populate the graph about sensors.
5. Choose appropriate databases to store 3D models, time series data, and graph data.
6. Provide a web application for the user to interact with the BIM model and time series data of the building.
7. Visualize real-time and historical data in the front-end web app.

The underlying data conversion and storage infrastructure are shown in **Fig. 2**.

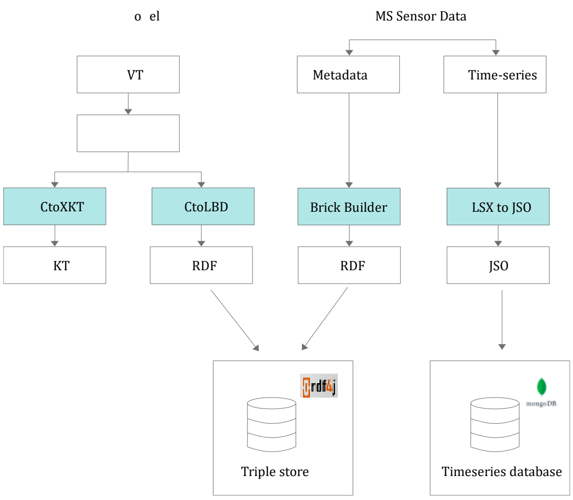

Fig. 2 – Data conversion and storage infrastructure.

3.1 Creating the semantic graph

A number of metadata representations are available that are useful in different phases of the building's lifecycle. IFC is an industry-wide standard data schema for BIM and covers many aspects in the design and construction phases of a building [3]. Geometry, building elements, and product properties are represented in IFC. IFCtoLBD converter [24] is a tool used to generate RDF triples using an IFC STEP file. It makes use of BOT, RDFS, and PROPS ontologies. The tool presented in [24] is used in this study to generate the RDF model of the building. Part of the IFC file containing information about the room "test 1 area" is shown in **Fig. 3.** In IFC, there is a Globally Unique Identifier (GUID) for every element and ''0KLkXPBfvES9D1y7EjijkE' is the GUID of the space.

```
#1266=IFCSPACE('0KLkXPBfvES9D1y7EjijkE',#42,'9',$,$,#1245,#1263,'test 1 area',.ELEMENT.,.SPACE.,$);
#1269=IFCSPACETYPE('0H12gSHLX9KAhW1PyBN1jS',#42,'test 1 area 9:543779',$,$,$,$,'543779',$,.NOTDEFINED.,$);
#1270=IFCPROPERTYSINGLEVALUE('Name',$,IFCLABEL('test 1 area'),$);
#1271=IFCPROPERTYSET('2b$cWxcoTB9AhqrEJK8Q2s',#42,'Pset_AirSideSystemInformation',$,(#1270));
```

Fig. 3 – Part of the IFC file containing IFCSPACE info.

The above IFC file, when converted to RDF, is shown in **Fig. 4.** "Space_1266" is a *bot:Space*, in which the number 1266 is related to the line number of the IFC file. The room can be uniquely identified by its GUID represented using *props:hasCompressedGuid* relationship.

```
inst:space_1266 a bot:Space ;
bot:containsElement inst:lightFixture_241729,
        inst:lightFixture_241879,
        inst:sensor_239793 ;
    props:hasCompressedGuid "0KLkXPBfvES9D1y7EjijkE"^^xsd:string ;
    bot:hasGuid "1456e859-2e9e-4e70-9341-f073adb2db8e"^^xsd:string ;
```

Fig. 4 - Part of the graph generated by the IFCtoLBD converter.

The Revit model of the building does not contain the BMS sensors, and, therefore, they are not available in the above graph. Compared to other structured representations of other building elements, the only source of metadata of the sensors is the mapping table shown in **Tab. 2** . However, the type of sensor and its location is available in this table. Other information such as their units and parent components were unavailable. To represent the BMS data points, this study relies on the Brick Ontology. Brick is built with a focus on supporting energy applications based on BMS in commercial buildings. It can represent metadata information of a sensor such as its location, function, and type. It also captures the relationships between those entities.

The RDF graph is created using the Brick-builder [25] tool. It is a simple method to generate a graph by providing the entities and their relationships.. A CSV file containing all the entities (shown in **Fig. 5**) and a text file defining the relationships, (shown in **Fig. 6**) form the inputs to the Brick-builder tool to generate the graph. Sensor ID, description, room number, room GUID, space ID, room name and the Brick sensor tags are included in this file. This CSV file is generated by joining the extracted spaces and GUID relationships from the graph generated from IFCtoLBD converter, and the BMS mapping table. By having the space identifier (e.g. space_1266) included in the graph, this identifier can be used to merge the two graphs.

```
11NR008QT-301CO2,"*CO2 MEASUREMENT 8_140",8.140,0KLkXPBfvES9D1y7EjijkE,space_1266,"test 1 area",CO2_Sensor
11NR008LT-302PIRTM,"PRESENCE 8_445",8.445,0LnrpxIav0Ju4gTy7JmW7k,space_6846,"Zone4- R9",Occupancy_Sensor
11NR008TE-302TRL,"ROOM TEMPERATURE 8_445",8.445,0LnrpxIav0Ju4gTy7JmW7k,space_6846,"Zone4- R9",Temperature_Sensor
```

Fig. 5 – Part of the CSV file.

```
inst:$5 rdf:type brick:Room
inst:$1 rdf:type brick:$7
inst:$1 brick:hasLocation inst:$5
inst:$1 rdfs:label $2
```

Fig. 6 – Relationships defined among columns of the CSV file.

This results in a graph with the relationships as shown in **Fig. 7**.

```
inst:11NR008QT-301CO2 a brick:CO2_Sensor ;
    rdfs:label "*CO2_MEASUREMENT_8_140"^^xsd:string ;
    brick:hasLocation inst:space_1266 .

inst:11NR008LT-302PIRTM a brick:Occupancy_Sensor ;
    rdfs:label"PRESENCE_8_445"^^xsd:string;
    brick:hasLocation inst:space_6846 .

inst:11NR008TE-302TRL a brick:Temperature_Sensor ;
    rdfs:label "ROOM_TEMPERATURE_8_445"^^xsd:string ;
    brick:hasLocation inst:space_6846 .
```

Fig. 7 – Relationships defined among columns of the CSV file

The first RDF graph (**Fig. 4**) semantically describes the context of the building and the second graph (**Fig. 7**) describes BMS sensors. These two graphs are stored in RDF4J database [26] as two separate graphs. RDF4J is a native Triplestore. An instance of the resulting visual graph using GraphDB is shown in **Fig. 8**. GraphDB [27] is another Triplestore that also provides visual representation of a graph.

3.2 Time series data

As stated, time series data from BMS sensors including temperature, occupancy and CO_2 are available as records in xlxs format as shown in **Tab. 1**. This data type fits well in a time series database. Therefore, MongoDB Timeseries Collection is used. Time series collection has the advantage of improved query efficiency and reduced disk usage for time series data, compared to normal document collections [28]. Before uploading to MongoDB, the timestamp was converted into Coordinated Universal Time (UTC) and data were formatted to JavaScript Object Notation (JSON) with the timestamp, sensor ID, and the value. A sample MongoDB document is shown in **Fig. 9.**

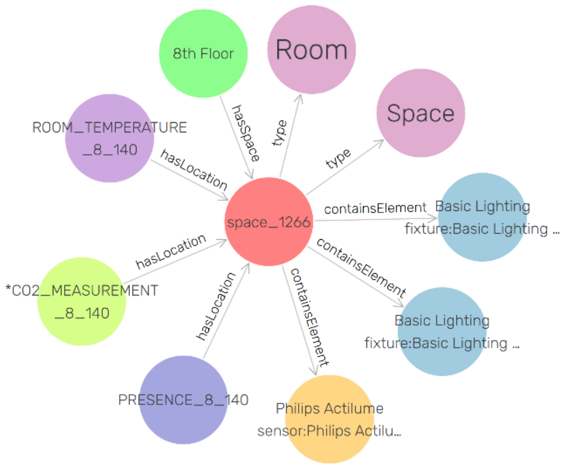

Fig. 8 – Graph representation of space_1266 using GraphDB.

```
{"timestamp":{"$date":"2021-03-31T22:02:00.000Z"},"sensor_id":"11NR008QT-301CO2","_id":{"$oid":"61bfcef2c72be614e6cda971"},"value":425}
{"timestamp":{"$date":"2021-03-31T22:02:00.000Z"},"sensor_id":"11NR008LT-302PIRTM","_id":{"$oid":"61bfceeac72be614e6caf861"},"value":2}
{"timestamp":{"$date":"2021-03-31T22:02:00.000Z"},"sensor_id":"11NR008TE-302TRL","_id":{"$oid":"61bfcec8c72be614e6c0d471"},"value":20.1}
```

Fig. 9 – Sensor data in MongoDB time series collection.

3.3 BIM-SIM web application

BIM-SIM is the web application developed to support the above tasks. The front end is built using React, a JavaScript framework. One intended functionality of the application is to view the building geometry in 3D using a web browser. The proposed approach utilises xeokit, an open-source JavaScript 3D graphics Software Development Kit (SDK) from xeolabs [29] [30] to render the 3D file in the browser. It provides functionalities like tree view, filtering elements by type, user event such as pick elements, etc. We used XKT format to view the 3D BIM model in the browser. XKT is a binary format and allows fast rendering. When converting, it preserves metadata in the IFC including the GUID. This GUID is used to uniquely identify the spaces in this application. The IFC model can be transformed into XKT, using several open-source command-line tools as described in [31]. The conversion flow is shown in **Fig. 10,** which is adapted from [31]. This conversion is implemented as a separate entity and is not a part of the web application. After the conversion process, the XKT file can be imported to the web application. **Fig. 11** shows the XKT file loaded in the web app. Plotly.js, a JavaScript graphing library, is utilised for charts. The NestJS framework is used to develop the API with Mongoose Object Data Modeling (ODM).

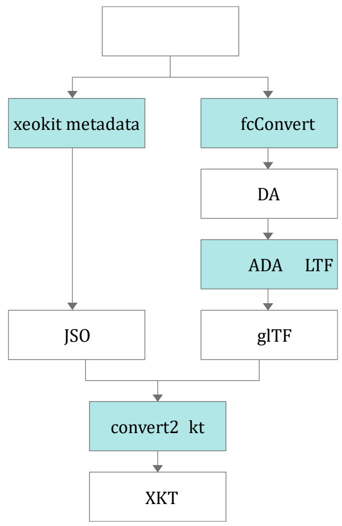

Fig. 10 - IFC to XKT conversion procedure [20].

Fig. 11 - XKT file loaded in the web app.

3.4 Functionality

When a particular space is selected from the 3D model, the front end application picks the element's GUID, which is used to call the API, where a SPARQL query is executed to find all sensors contained in that space. A SPARQL query is sent in the request body in a POST request to the RDF4J server. Then the "Info window" is populated by the sensors contained in that space. These sensors are extracted from the RDF graph using the query shown in **Fig. 12.**

```
PREFIX brick:
<https://brickschema.org/schema/Brick#>
PREFIX inst:
<http://linkedbuildingdata.net/ifc/resources20201208_005325/>
PREFIX props: <https://w3id.org/props#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
select * where {
    ?space props:hasCompressedGuid
'0KLkXPBfvES9D1y7EjijkE' .
    ?sensors brick:hasLocation ?space .
    ?sensors rdf:type ?sensor_type .
}
```

Fig. 12 - SPARQL query that runs in the API.

This query results three sensors that are contained in that space as shown in **Tab. 3.**

Tab. 3 – SPARQL query results.

	space	sensors	sensor_type
1	inst:space_1266	inst:11NR008LT-301PIRTM	brick:Occupancy_Sensor
2	inst:space_1266	inst:11NR008QT-301CO2	brick:CO2_Sensor
3	inst:space_1266	inst:11NR008TE-301TRL	brick:Temperature_Sensor

These results are available via the web application un er "3 sensors" as shown in **Fig. 13** – Info window populated with sensors from SPARQL query.

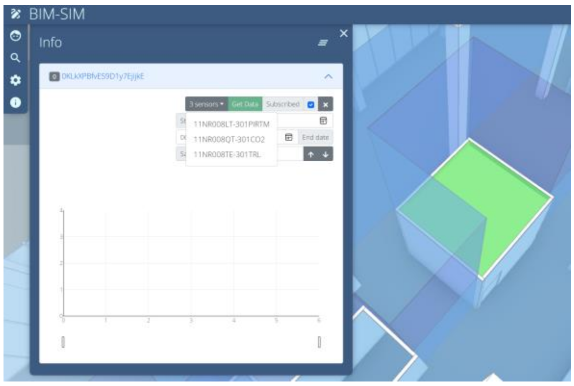

Fig. 13 – Info window populated with sensors from SPARQL query.

Then, selecting a particular sensor calls the API which sends a query to the MongoDB database using the sensor ID to retrieve historical data, and the returned data is displayed in a chart as shown in **Fig. 14**. The user can adjust the time interval and the sample size of data as well. When displaying sensor data, its metadata is also displayed. In this example, it is limited to the type of sensor (Brick CO2 Sensor), but this can be extended to display the location and other relationships as well, when available. This is particularly important as opposed to extracting time series data from a database, without knowing its provenance. Usually, naming conventions of BMS data points are ambiguous and do not provide much information about the sensor.

4 Discussion and conclusion

A large volume of data is produced in each phase of a building. In its operational phase, these data become highly valuable for many applications. This paper demonstrates how various data sources in a building can be integrated in a distributed manner.

Compared to using commercially available BIM tools for visualizing sensor data, this paper provides a vendor-neutral method. Furthermore, as opposed to creating one central information model, we acknowledge the fact that it is more efficient to retain time series data in their optimum environment. Since there are 1,314,720 records per month, if all the sensor observations were included in the graph, this will add $1{,}314{,}720 \times 12 \cong 15M$ additional triples to the graph, without any value addition.

Fig. 14 – CO_2 Sensor data visualization.

The BIM model is used as the main information source to develop the knowledge graph of the building. Secondly, BMS sensor semantics were introduced as another graph, using the Brick ontology. The IFC file is also used as the 3D model of the building, but it is limited to the XKT format to which it is translated, thereby preserving the geometry information in an industry standard. The web application provides the opportunity to interact with the knowledge graph, the 3D model, and the BMS sensor data. This integration was achieved by leveraging semantic web technologies, an open-source 3D graphics SDK, an API, and state-of-art web tools. Although the demonstration was done for an interactive monitoring application, other similar applications like reporting on energy usage and indoor climate are also possible via the same resources. Visualizing real-time data through the web application is possible using the MQTT JavaScript client with WebSocket. This is planned as a future implementation. Most importantly, this platform will further be extended to introduce building controllers and will be implemented on top of the existing BMS infrastructure. Although basic semantics (location, type) were used to describe BMS sensor data, more metadata such as unit, mounting height, accuracy, shall be described as required by applications in the future.

5 Acknowledgement

This project received funding from the Dutch Ministry of Economic Affairs and Climate Policy and Ministry of the Interior and Kingdom Relations under the MOOI program.

References

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

[2] Bilal M, Oyedele LO, Qadir J, Munir K, Ajayi SO, Akinade OO, et al. Big Data in the construction industry: A review of present status, opportunities, and future trends. Adv Eng Informatics [Internet]. 2016;30(3):500–21. Available from: http://dx.doi.org/10.1016/j.aei.2016.07.001

[3] Pauwels P, Costin A, Rasmussen MH. Knowledge Graphs and Linked Data for the Built Environment [Internet]. Springer International Publishing; 2022. 157–183 p. Available from: http://dx.doi.org/10.1007/978-3-030-82430-3_7

[4] Tang S, Shelden DR, Eastman CM, Pishdad-Bozorgi P, Gao X. A review of building information modeling (BIM) and the internet of things (IoT) devices integration: Present status and future trends. Autom Constr [Internet]. 2019 May 1 [cited 2022 Mar 15];101(June 2018):127–39. Available from: https://linkinghub.elsevier.com/retrieve/pii/S0926580518305764

[5] Pauwels P, Petrova E. Information in Construction. 2020.

[6] Eastman C, Teicholz P, Sacks R, Liston K. BIM Handbook, a Guide to Building Information Modelling 2nd ed. John Wiley & Sons, Inc, Hoboken. 2011.

[7] ISO - ISO 16739-1:2018 - Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries — Part 1: Data schema [Internet]. [cited 2022 Jan 6]. Available from: https://www.iso.org/standard/70303.html

[8] Fierro G. Self-Adapting Software for Cyberphysical Systems. 2021; Available from: http://www2.eecs.berkeley.edu/Pubs/TechRpts/2021/EECS-2021-159.html

[9] Pauwels P, Terkaj W. EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology. Autom Constr. 2016 Mar 1;63:100–33.

[10] Gruber TR. A Translation Approach to Portable Ontology Specifications. Knowl Aquis. 1993;5(2).

[11] Malcolm A, Werbrouck J, Pauwels P. LBD Server: Visualising Building Graphs in Web-Based Environments Using Semantic Graphs and GlTF-Models. Adv Sci Technol Innov. 2021;(2020):287–93.

[12] Bonduel M, Oraskari J, Pauwels P, Vergauwen M, Klein R. The IFC to linked building data converter - Current status. CEUR Workshop Proc. 2018;2159:34–43.

[13] Haller A, Janowicz K, Cox SJD, Lefrançois M, Taylor K, Le Phuoc D, et al. The modular SSN ontology: A joint W3C and OGC standard specifying the semantics of sensors, observations, sampling, and actuation. Semant Web. 2018;10(1):9–32.

[14] Hammar K, Wallin EO, Karlberg P, Hälleberg D. The RealEstateCore Ontology. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 2019.

[15] Project Haystack - Tags [Internet]. 2020 [cited 2021 Aug 22]. p. 2016. Available from: https://project-haystack.org/tag.

[16] Balaji B, Bhattacharya A, Fierro G, Gao J, Gluck J, Hong D, et al. Brick: Metadata schema for portable smart building applications. Appl Energy [Internet]. 2018;226(February):1273–92. Available from: https://doi.org/10.1016/j.apenergy.2018.02.091

[17] Suzuki LR. Data integration approaches for smarter operation of large commercial buildings. 2015;

[18] Bhattacharya AA, Hong D, Culler D, Ortiz J, Whitehouse K, Wu E. Automated metadata construction to support portable building applications. BuildSys 2015 - Proc 2nd ACM Int Conf Embed Syst Energy-Efficient Built. 2015;3–12.

[19] Desogus G, Quaquero E, Rubiu G, Gatto G, Perra C. Bim and iot sensors integration: A framework for consumption and indoor conditions data monitoring of existing buildings. Sustain. 2021;13(8).

[20] Oti AH, Kurul E, Cheung F, Tah JHM. A framework for the utilization of Building Management System data in building information models for building design and operation. Autom Constr [Internet]. 2016;72:195–210. Available from: http://dx.doi.org/10.1016/j.autcon.2016.08.043

[21] Rasmussen MH, Frausing CA, Hviid CA, Karlshoj J. Demo: Integrating building information modeling and sensor observations using semantic web. CEUR Workshop Proc. 2018;2213(1):48–55.

[22] Donkers A, Yang D, De Vries B, Baken N. Real-Time Building Performance Monitoring using Semantic Digital Twins. Proc 9th Linked Data Archit Constr Work [Internet]. :12. Available from: http://ceur-ws.org/Vol-3081/05paper.pdf

[23] Hu S, Corry E, Curry E, Turner WJN, O'Donnell J. Building performance optimisation: A hybrid architecture for the integration of contextual information and time-series data. Autom Constr [Internet]. 2016;70:51–61. Available from: http://dx.doi.org/10.1016/j.autcon.2016.05.018

[24] pipauwel/IFCtoLBD: IFCtoLBD converts IFC (Industry Foundation Classes STEP formatted files into the Linked Building Data ontologies. (github.com) [Internet]. Available from: https://github.com/pipauwel/IFCtoLBD

[25] gtfierro/brick-builder (github.com). Available from: https://github.com/gtfierro/brick-builder

[26] Eclipse RDF4J™ | The Eclipse Foundation [Internet]. [cited 2022 Jan 11]. Available from: https://rdf4j.org/

[27] GraphDB Downloads and Resources (ontotext.com) [Internet]. [cited 2022 Jan 11]. Available from: https://graphdb.ontotext.com/

[28] Time Series Collections — MongoDB Manual [Internet]. [cited 2022 Jan 7]. Available from: https://docs.mongodb.com/manual/core/timeseries-collections/

[29] xeokit - 3D Web Programming Toolkit for BIM and Engineering Visualization [Internet]. [cited 2022 Jan 6]. Available from: https://xeokit.io/

[30] xeokit/xeokit-sdk: Open source JavaScript SDK for viewing high-detail, full-precision 3D BIM and AEC models in the Web browser. (github.com) [Internet]. Available from: https://github.com/xeokit/xeokit-sdk

[31] Converting IFC Models to XKT using 3rd-Party Open Source Tools (notion.so) [Internet]. [cited 2022 Jan 6]. Available from: https://www.notion.so/Converting-IFC-Models-to-XKT-using-3rd-Party-Open-Source-Tools-c373e48bc4094ff5b6e5c5700ff580ee

The datasets generated during and/or analysed during the current study are not available because of the privacy policy, but the authors will make every reasonable effort to publish them in near future.