Complex Systems Informatics and Modeling Quarterly (CSIMQ)
eISSN: 2255-9922
Published online by RTU Press, https://csimq-journals.rtu.lv
Article 259, Issue 46, March/April 2026, Pages 97–116
https://doi.org/10.7250/csimq.2026-46.05

A Hybrid LLM–Knowledge Graph Architecture for Information Retrieval in Smart Buildings

Emanuele Laurenzi[1]*, Massimo Callisto De Donato[2]*, and Daniele Porumboiu[2]

[1]FHNW - University of Applied Sciences and Arts Northwestern Switzerland, Riggenbachstrasse 16,
4006 Olten, Switzerland
[2]University of Camerino, Computer Science Division, via Madonna delle Carceri 7,
62032 Camerino, Italy

emanuele.laurenzi@fhnw.ch, massimo.callisto@unicam.it, daniele.porumboiu@unicam.it

Abstract. The Smart Building paradigm promises a future where buildings are intelligent, adaptive, and sustainable, offering real-time information retrieval that supports decision-making to enhance energy efficiency, occupant comfort, and security. However, achieving this paradigm is highly complex, one major reason being the seamless integration of (a) physical and functional representations of buildings (i.e., Building Information Modeling) and (b) real-time IoT (i.e., Internet of Things) data. To address this challenge, we propose a hybrid LLM–Knowledge Graph approach in which, on the one hand, a knowledge graph retains the buildings' knowledge structures and a time-series database stores IoT data. On the other hand, a Large Language Model (LLM) serves as a mediator between a facility manager and the knowledge graph and IoT data, facilitating data-driven decision-making processes. Adopting the Design Science Research (DSR) methodology, we executed two distinct iterations within the implementation phase to investigate contrasting knowledge representation paradigms: Resource Description Framework (RDF) and Labeled Property Graph (LPG). Each iteration was evaluated to assess the effectiveness of the corresponding systems. Finally, we compared the RDF- and LPG-based implementations of the proposed architecture and drew insights.

Keywords: Smart Building, Building Information Modeling, Internet of Things, Hybrid Artificial Intelligence, LLM–Knowledge Graph Architecture Ontologies, Knowledge Graphs, Labeled Property Graph, Large Language Model.

1 Introduction

Building Information Modeling (BIM) and the Internet of Things (IoT), while powerful individually, achieve a new level of significance when their inherent strengths are combined [1]. BIM strives to provide a comprehensive digital representation of a building's physical and functional characteristics, offering a centralized repository of information throughout the building's

---

* Corresponding author

Reference: E. Laurenzi, M. Callisto De Donato, and D. Porumboiu, "A Hybrid LLM–Knowledge Graph Architecture for Information Retrieval in Smart Buildings", Complex Systems Informatics and Modeling Quarterly, CSIMQ, no. 46, pp. 97–116, 2026. Available: https://doi.org/10.7250/csimq.2026-46.05

Additional information. Author ORCID iD: E. Laurenzi – https://orcid.org/0000-0001-9142-7488, M. Callisto De Donato – https://orcid.org/0000-0001-6745-4785. PII S225599222600259X. Received: 7 January 2026. Accepted: 9 March 2026. Available online: 30 April 2026.

lifecycle, from design to demolition. IoT, on the other hand, furnishes a network of connected sensors and devices that generate real-time data on a building's operation and environment [2]. When integrated, this confluence of rich design data and dynamic operational data enables unprecedented insights. Domain experts like facility managers can optimize energy consumption based on actual usage patterns, predict maintenance needs based on sensor readings, and enhance occupant comfort through automated adjustments, leading to substantial cost savings and improved building performance. However, integrating BIM and IoT remains challenging. Because they evolved separately, differences in data formats, communication systems, and interoperability hinder data sharing and collaboration between the fields [1]. Overcoming these issues is essential to realizing the full potential of a connected, data-driven approach in construction and building management. This challenge is often exacerbated by the fact that most IoT devices and sensors are deployed during the operational phase, while they are not typically integrated into the initial BIM models created during the design phase [3]. Consequently, real-time data generated by IoT systems (such as energy usage, environmental conditions, or equipment status) is frequently excluded from BIM models. This exclusion limits the ability to utilize BIM as a dynamic, evolving tool that accurately reflects the building's performance and operational conditions throughout its lifecycle.

As discussed in Section 2, recent works addressing the integration of BIM with IoT rely extensively on semantic technologies such as ontologies and knowledge graphs. While these offer clear advantages in resolving interoperability across heterogeneous data models, their use for effective information retrieval still requires substantial engineering effort to develop interfaces capable of supporting stakeholder decisions during building operation. In contrast, this article proposes a hybrid LLM–Knowledge Graph approach that combines a structured knowledge graph for BIM and IoT-sensed data with a Large Language Model (LLM) for natural language information retrieval, ultimately supporting decision-making for facility managers during the building's operational phase.

The primary stakeholder we focus on is the facility manager. Beyond traditional maintenance duties, facility managers are increasingly acting as data-driven strategists responsible for meeting environmental targets and reducing operational costs, which are goals that require the integration of IoT sensors with BIM for real-time operational monitoring

This work extends our previous contribution in [4], where a hybrid AI approach integrating LLMs, Knowledge Graphs, and IoT data was proposed to support data-driven decision-making in smart buildings. Building on this foundation, we systematically contrast two competing knowledge representation paradigms, namely Resource Description Framework (RDF) and Labeled Property Graph (LPG), and investigate how each shapes information retrieval performance when combined with LLMs. Following the Design Science Research methodology [5], the implementation phase was iterated twice to instantiate two technically equivalent prototypes, one per paradigm, evaluated under identical experimental conditions to ensure comparability.

In detail, this article's extensions are the following:

- The Design Science Research (DSR) process has been re-iterated by introducing a second implementation and evaluation cycle for the comparison of the two knowledge representation paradigms, RDF- and LPG-based.
- The LLM model has been upgraded from Claude 3.5 Sonnet to the more recent Claude 4.5 Sonnet version.
- A systematic comparison between RDF and LPG models has been delivered, highlighting performance and structural differences when applying the same set of competency questions to both implementations.
- The latest implementation source code is available on GitHub at https://github.com/PROSLab/BIM-IoT-Assistant.

The remainder of this article is as follows. Section 2 describes the related work and motivates the research question. Section 3 introduces DSR as the adopted methodology. Section 4 describes the proposed hybrid LLM–Knowledge Graph architecture. The proof of concept and the discussion on the current limitations are elaborated in Sections 5 and 6, respectively. Finally, Section 7 summarizes and concludes the article.

2 Related Work

Recent literature reports significant efforts in integrating the two paradigms BIM and IoT, for various purposes. The tendency for BIM is to leverage ontologies and knowledge graph technologies [6], which narrowed the scope of the literature considered in this section.

In [7], the authors introduced a cross-source data management and analysis framework to support evacuation path planning and emergency response decisions in fire scenarios, supported by a specialized FireEvacuation ontology. The work in [8] proposed a service-oriented architecture for data-driven smart buildings, using semantic technologies as an integral part of the architecture, essential for adding context to operational data and creating links between diverse systems. The approach in [9] proposed a flexible energy modeling framework based on the SAREF ontology and its SAREF4BLDG extension. It offers models for typical systems and devices, and a method for linking and simulating components using, also, the SAREF4SYST extension.

Researchers in [10] presented a framework for integrating BIM and IoT data using an ontology-based mediation mechanism. It enables integrated access to local BIM and IoT data through query-rewriting processes.

The paper [11] introduced the Building Topology Ontology (BOT), which provides a high-level description of the topology of buildings, including stairs and spaces, the building elements they contain, and their web-friendly 3D models. They also describe how existing applications produce and consume datasets, combining BOT with other ontologies that describe product catalogs, sensor observations, or Internet of Things (IoT) devices. [12] demonstrates how the integration of BIM and IoT data can be used to monitor the indoor environmental quality of a building. With their approach, they were able to query the topology, static, and dynamic properties from a graph database and then query the corresponding sensor data from a time-series database.

The work in [13] conducted a review of the main ontologies and applications that support the development of Decision Support Systems and decision-making in the different phases of a building’s life cycle. This study also highlighted that most ontologies lack real-life applications and that some applications are focused mostly on the design phase of a building or its early operation, indicating their early development stage. Authors in [14] designed an ontology, called Building Performance Ontology (BOP), that integrates topological building information with static and dynamic properties for improving the monitoring of indoor environments.

Authors in [15] introduced a novel multi-layer architecture and a comprehensive framework for smart-building digital twins, with a primary focus on enabling semantic interoperability among smart-building digital twin applications. The approach combines a semantic and static representation of the building (BIM) with dynamic (IoT) data to satisfy the real-time requirements of smart-building digital twins, while preserving IoT data in its optimal time-series storage.

Similarly, in [16] the authors showcased the integration of construction documentation, facility management records, and real-time data obtained from building automation systems within a Cognitive Digital Twin. A W3C-compatible approach was created, drawing from the BOT ontology and integrating it with the Brick Ontology.

The authors of [17] developed a Digital Twin using a microservice architecture, which facilitates cloud deployment and enables modularly defined functionalities. The knowledge graph here serves as the contextual interface that provides a comprehensive view of all data and all models. The semantic information is stored in the Neo4j database and structured as a property graph, following the concepts and relations defined in the IFC schema.

The authors of [18] constructed a general City Information Model ontology to integrate BIM, geographic information system (GIS), and IoT data. A new ontology has been developed (BIM-GIS

Integration Ontology) and mapped with the Uniform Metadata Schema for Buildings (Brick)[†] and Semantic Sensor Network (SSN)[‡] ontologies.

In [19], the researchers described a realization of a Semantic Digital Twin through the use of modular knowledge graphs instead of using monolithic graph architectures. The advantage of the approach lies in the possibility of merging independently developed knowledge graphs into a single one that is easier to understand, better to reason with, and also reusable. In addition, when integrated with real-life systems, modular graphs improve performance by loading only the needed segments, eliminating problems with querying and reasoning in large stores.

The authors of [20] proposed an approach to calculate, analyze, and monitor energy performance indicators for buildings. The structured knowledge (building, energy system-related aspects, and performance indicators) is captured in ontologies, and an application layer serves as an orchestrator that integrates the latter with a time-series database containing sensed data. This approach proved efficient in its architectural design, which, by combining them, keeps ontologies and sensed data in separate databases, thereby easing their maintainability.

Although semantic-based approaches have advantages, information retrieval presents a common challenge, demanding either substantial expertise in ontology design for complex (SPARQL[§] or Cypher[¶]) queries or the development of integrated, user-friendly interfaces.

Two graph models are gaining popularity as semantic approaches in Smart Buildings namely, RDF and LPG. The authors of [21] use a Smart Home Dataset and a kitchen dataset to compare the two graph models both qualitatively and quantitatively, finding that native labeled property graphs are less complex and outperform atomic RDF in complex graph traversals, while RDF shows qualitative advantages for multi-domain and multi-stakeholder environments through ontologies and HTTP URIs, making it a more stable interoperability format.

Recent advances in Information Systems suggest combining the strengths of both Knowledge Graphs and Large Language Models to address above mentioned limitations [22], [23]. Such a combination falls into the research realm of Hybrid Artificial Intelligence, because it considers two approaches from the two sides of AI: the Symbolic AI (i.e., KG) and Sub-symbolic AI (i.e., LLM).

Therefore, the research question we investigated in this work is as follows:

To what extent does a hybrid LLM–Knowledge Graph architecture support effective natural language-based information retrieval for facility managers, and how do RDF- and LPG-based implementations compare in terms of retrieval effectiveness when integrating BIM and IoT data?

3 Methodology

The methodology used to answer the research question of this work is the DSR, which proposes five main phases [5]: problem awareness, suggestion, development, evaluation and conclusion.

During the *problem awareness phase*, we deepen the understanding of the problem from both research and application perspectives. From the literature, we identified the architecture proposed in Donkers et al. [12] (for real-time building performance monitoring using semantic digital twins) to be the closest to our problem; therefore, we decided to build our approach from it. Their architecture integrates knowledge structures of BIM and IoT data to monitor the indoor environmental quality (e.g., air quality index) of a building. Specifically, they used an Open Smart Home (OSH) dataset [24] conformed to the BOT [11] and BOP [14] ontologies and built a custom Python component to navigate the knowledge graph, pick a property (e.g., temperature, humidity, and illuminance), and return the corresponding value from a time-series database for an overall calculation.

From an application perspective, we focused on the same OSH dataset and analyzed it using the developed prototype.

[†] https://brickschema.org/

[‡] https://www.w3.org/TR/vocab-ssn/

[§] https://data.europa.eu/en/about/sparql

[¶] https://neo4j.com/docs/getting-started/cypher/

The OSH dataset was developed by the Fraunhofer Institute for Building Physics in Nuremberg, Germany [25] and provides both static and dynamic aspects of a real smart home environment and is intended to support investigations into energy efficiency, control strategies, and building performance analysis. The OSH scenario represents a two-story building; the ground floor is shown in Figure 1. This floor has a bathroom, kitchen, lobby, and toilet, each enclosed by four walls and a ceiling. Walls include doors, windows, and sensors. A gas boiler provides hot water to radiators in all rooms except the lobby. Windows have manual shutters for shading.

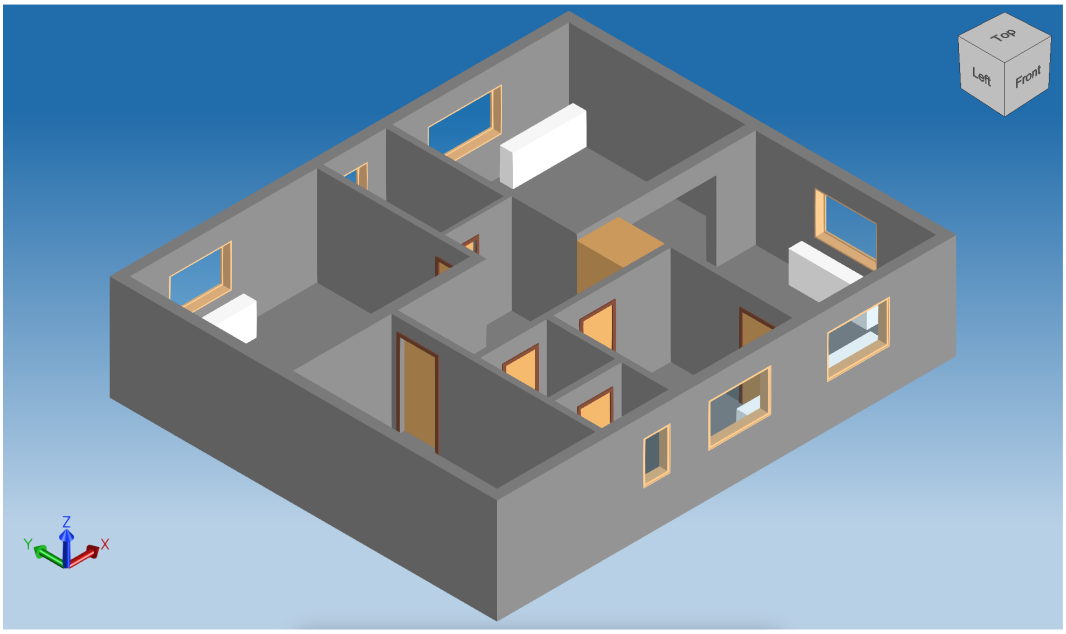


Figure 1. Rendering of first building level from Open Smart Home Data [24]

Static data is available in IFC[‖] and Revitprotect[*] formats, and also in RDF and makes use of concepts from the BOT and PROPS[♯] ontologies. Dynamic data, representing sensor readings, is provided in CSV[♭] and RDF formats, employing the SSN/SOSA[♮] ontologies. Measurements from the sensors span the period from March 9, 2017, to June 6, 2017, with a variable sampling rate up to 15 minutes. The smart home is equipped with a system offering the following capabilities:

- Wall-mounted sensors in rooms with space heaters (not in the staircase or lobby), measuring air temperature, illuminance, and humidity.
- Remote-controlled thermostat valves on each heater, logging the setpoint and local air temperature.
- A base station that communicates wirelessly with sensors and actuators, connects to the internet for weather forecasts, and provides a virtual outdoor temperature per room.
- A smartphone application for controlling setpoints, scheduling, and monitoring real-time measurements.

In the *suggestion phase*, we proposed a novel hybrid LLM–Knowledge Graph architecture, where an LLM enables the integration of concepts from a knowledge graph and respective IoT values from a time-series database (see Section 4).

In the *implementation phase*, we implemented the proposed architecture in a technical prototype, where the user interface takes the form of a virtual assistant or bot (see Section 5).

[‖] https://technical.buildingsmart.org/standards/ifc/ifc-formats/

[*] https://www.spatial.com/glossary/revit-file-format

[♯] https://github.com/MadsHolten/props

[♭] https://data.europa.eu/apps/data-visualisation-guide/csv-files

[♮] https://www.w3.org/TR/vocab-ssn/

Expanding our previous findings [4], we re-engineered the knowledge representation layer by substituting the RDF-based knowledge graph with an LPG model. This choice goes beyond a simple technical replacement and enables a critical comparative analysis of these traditionally competing graph technologies. In doing so, we also challenge the binary perception of RDF and LPG as mutually exclusive, instead exploring their respective efficiencies in supporting real-time traversal. Comparisons between RDF and LPG representations are not new (e.g., [21]). However, prior work has primarily focused on modeling and interoperability concerns. By contrast, our contribution lies in analyzing these paradigms from the perspective of LLM-driven information retrieval within a hybrid LLM–Knowledge Graph architecture.

Finally, in the *evaluation phase*, we focus on evaluating the effectiveness of our approach because existing solutions for information retrieval in BIM–IoT integrated systems still require substantial engineering effort, i.e., the development of complex query interfaces or domain-specific visualization tools, to make structured knowledge accessible to non-technical stakeholders such as facility managers. In this context, we define effectiveness as the system's ability to correctly answer competency questions, i.e., questions that facility managers find helpful to answer. The strategy of answering competency questions is a well-known evaluation technique in ontology engineering [26] that assesses the effectiveness of the developed ontology. The competency questions were derived from the presented dataset and validated by domain experts. An example of such a competency question is the following: *For thermal performance and maintenance purposes, it is important to know the material composition, the structure, and the dimensions of specific walls. Please provide these*. Answering a competency question in our proposed architecture means that the system shall be able to interpret a natural language query and return a response that is both (1) semantically relevant to the expressed information need and (2) factually grounded in the structured knowledge graph, without requiring the user to have prior knowledge of the underlying query language or data schema. Establishing this foundational capability is a necessary precondition before evaluating complementary dimensions such as answer quality or computational sustainability, which we identify as priorities for future work.

To provide insights into how both prototypes can be effective in information retrieval, we conducted two distinct evaluations, one with the RDF-based knowledge graph and one with the LPG, while keeping the same settings (i.e., same data sets and same LLM model) (see Section 5.2).

4 The Hybrid LLM–Knowledge Graph Architecture

In this section, we describe the proposed hybrid LLM–Knowledge Graph architecture that combines knowledge graph and LLM capabilities to support information retrieval for domain experts. Such an architecture is valid for both RDF-based knowledge graphs and LPG-based implementations.

Given the ultimate goal of supporting information retrieval for domain experts such as facility managers, we followed three main design principles in our architecture:

1. We first took the architecture in [12] as a blueprint for integrating the IoT data with structured knowledge of buildings. Unlike [12], we replaced the static Python component with an LLM-based component to promote scalability of the approach.
2. Next, as suggested in [20], we kept the ontologies and sensed data in different databases (i.e., graph database and time-series database, respectively) and ensured the traceability of data values and sensors from the two databases through ID mapping.
3. Finally, for the retrieval capability, we leveraged the knowledge graph-based question answering (KG-QA) pattern [27] – in RDF-based knowledge bases, this is also known as "Text-to-SPARQL translation". Rather than relying on the LLM's internal weights, our architecture uses prompt engineering to teach models to generate SPARQL queries. These queries are executed against the knowledge graph, ensuring the retrieval of "ground truth"

structured data. This approach is proving to be more effective than conventional Retrieval Augmented Generation (RAG) [28], which struggles with query-focused summarization (QFS) tasks, thereby not suitable to answer our competency questions with accuracy.

The proposed architecture conceptually comprises three main layers. (1) The interaction layer provides user access through a virtual assistant capable of addressing the information needs of domain experts. (2) A mediation layer based on a Large Language Model acts as an intermediary, combining sub-symbolic capabilities with symbolic knowledge representations. (3) The data and knowledge layers contain, respectively, the knowledge graph describing the BIM and the sensed IoT data. The resulting architecture is illustrated in Figure 2.

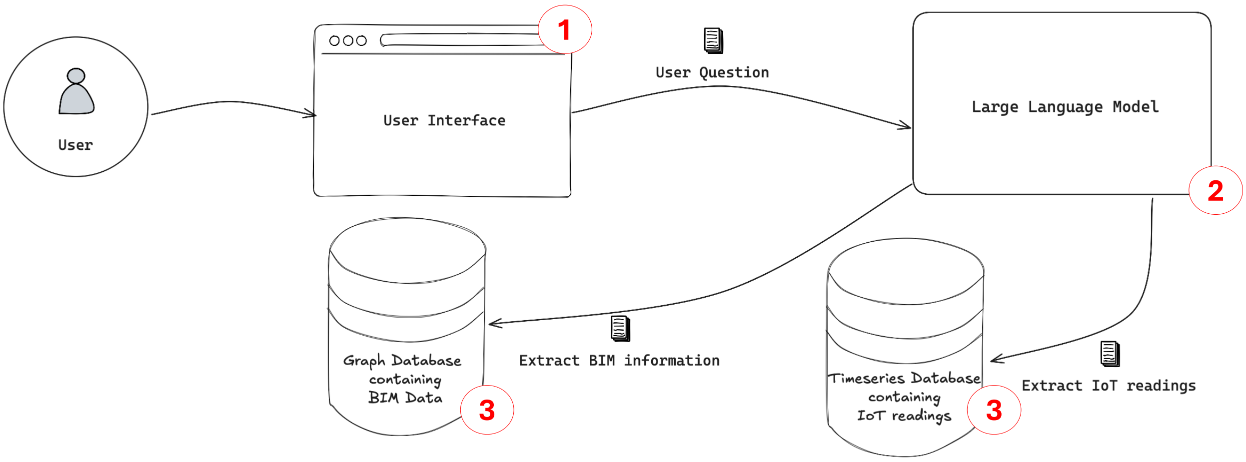


Figure 2. Overview of the proposed hybrid LLM–Knowledge Graph architecture and the related conceptual components: (1) user interaction layer, (2) the hybrid AI mediation layer based on a Large Language Model, and (3) the data and knowledge layer integrating a knowledge graph and a time-series database for BIM and IoT data.

The hybrid LLM–Knowledge Graph architecture relies on a **knowledge graph** to semantically represent the structure of both the BIM model and the IoT infrastructure. According to ontology development methodologies [26], the selection of existing ontologies is an essential step to ensure reusability, interoperability and scalability. Following the findings from Donkers et al. [12], we selected BOT and BPO ontologies. The BOT ontology is used to describe the physical structure of the building, including elements such as spaces, walls, and their topological relationships, making it suitable for modeling knowledge about the structure of the building. The BPO ontology, on the other hand, is employed to represent knowledge about sensor and actuator data. This combination allows us to create a more comprehensive model that represents the integration of BIM and IoT information effectively. The combined semantic information including building structure and IoT infrastructure information are stored into RDF format using the knowledge graph.

While a graph database model is well-suited for storing entities and their semantic relationships in RDF triple stores, it does not scale efficiently when dealing with large volumes of historical IoT data [29]. To handle the actual IoT sensor data, a **time-series database** is used as a complementary component to the knowledge graph [30]. Its role is to efficiently store and manage the sensor readings over time, such as temperature, humidity, and light levels. The database enables fast retrieval of the most recent sensor values, supports aggregation queries over historical data, and allows for horizontal scalability when managing large volumes of IoT measurements. The linkage between the two layers is established via *unique sensor identifiers* that are consistently represented in both the knowledge graph and the time-series database.

The combination of the knowledge graph and the time-series database results in two distinct data storage systems, typically each with its own query language such as SPARQL for semantic information retrieval from the knowledge graph, and dedicated NoSQL[△]-like query languages for accessing the time-series database [31]. Moreover, having two separate data stores requires the user to define additional aggregation mechanisms, potentially increasing the complexity of retrieving meaningful insights about the building.

To overcome this limitation, we introduce the **LLM** component with a retrieval-augmented generation capability [28], which provides uniform access to both data sources. The LLM acts as a mediator in the information retrieval process, being enhanced with external knowledge retrieved from both the knowledge graph and the time-series database. We supplemented the LLM with a dedicated user interface, which acts as a *virtual assistant* capable of understanding and responding to natural language queries from domain experts. This assistant enables users to ask complex questions about building components, their properties, and sensor readings, providing a more intuitive and accessible approach to information retrieval tasks related to the building's knowledge structure and IoT data.

In summary, in the proposed architecture, the LLM served as an integration component for the virtual assistant, featuring the following capabilities: *Query Interpretation*: the model will analyze natural language queries from users, understanding the intent behind questions about building components, sensor data, or their relationships. *Database Query Generation*: Based on the interpreted user intent, the model will generate appropriate queries for our graph and time-series databases. *Result Interpretation and Explanation*: After receiving raw data from the databases, the model will interpret the results and generate human-readable explanations for the end-user.

5 Proof of Concept

In this section, we describe both (1) the technical prototype that implements the suggested hybrid LLM–Knowledge Graph architecture, and (2) the evaluation of the approach's effectiveness. For the latter, we used the competency questions that, in natural language, have been added as prompts to the virtual assistant. The results have been compared qualitatively with the OSH dataset, which serves as the ground truth. The source code for the implementation is available on GitHub[▽].

5.1 Artifact Development

For storing the BIM knowledge structure, we followed two cycles. In the first cycle, we used **GraphDB** by Ontotext[▲], a semantic graph database compliant with W3C standards and designed to store and manage RDF data. GraphDB is fully compatible with widely used ontology standards and supports the SPARQL query language, which enables users to retrieve and manipulate data stored in the database.

In the second cycle, we implemented the same BIM knowledge structure using an LPG model based on **Neo4j**[▼], a graph database designed for managing and querying highly connected data. Neo4j adopts Cypher as its native declarative query language for querying and manipulating data stored in the database.

For storing the IoT data, we used the time-series database **InfluxDB**[◦] by InfluxData. InfluxDB is optimized for fast ingestion, querying, and aggregation of time-series data, making it well-suited for monitoring, IoT, and real-time analytics applications. InfluxDB provides *Flux*, a powerful functional query language designed explicitly for time-series workloads. It supports data manipulation and statistical analysis, while enabling joins across data streams to enrich real-time metrics with contextual information from external systems.

[△] https://www.mongodb.com/resources/basics/databases/nosql-explained

[▽] BIM-IoT-Assistant: https://github.com/PROSLab/BIM-IoT-Assistant

[▲] GraphDB: https://www.ontotext.com/products/graphdb/

[▼] Neo4j: https://neo4j.com/product/

[◦] InfluxDB: https://www.influxdata.com/

For the LLM component, we integrated **Claude 4.5 Sonnet**•, developed by Anthropic (Claude provider), using the standard cloud API offered by OpenRouter◇. We selected Claude over other LLMs because, at the time of implementation, it was among the best-performing for graph query generation tasks [32].

Compared to other LLM models, Sonnet demonstrated a stronger ability to follow logical steps and reasoning, which was pivotal for interpreting user queries about building data.

Compared to its siblings in the Claude 4 family, the Claude 4.5 Sonnet offers a superior approach. Namely, **Claude 4 Haiku**: While Haiku is faster, Sonnet offered more advanced reasoning capabilities and handles complex queries better, making it more suitable for our needs where query interpretation complexity is high. **Claude 4 Opus**: Opus might offer marginally higher capability for extremely complex tasks, but Sonnet provided a better balance of speed and user intent understanding. As elaborated in the next section, tests demonstrated that for our case, Sonnet's capabilities were sufficient to answer the competency questions, while its faster response times allowed for a smooth user experience. In practice, Claude enabled us to interpret natural-language queries from users by identifying the underlying intent behind questions about the building's knowledge structure and IoT sensor data. It then generated appropriate queries for both the knowledge graph and the time-series database, processed the raw results, and produced human-readable explanations.

To integrate LLM and data sources, we employed **LangChain**♢, a popular Python-written framework designed to facilitate the development of LLM-driven applications. LangChain is based on three core components: *Chains* which represent deterministic sequences of steps to handle user inputs such as prompting, parsing, and transformation; *Tools* which are custom Python modules used to integrate the external systems through APIs and allow the LLM to obtain augmented information used to enrich the responses; *Agents* which use the LLM to reason over tasks and dynamically decide which Tools or Chains to invoke based on the available context.

In the first cycle, we used the Chain *OntotextGraphQAChain* from LangChain. This Chain is designed to interact with GraphDB, generating SPARQL queries based on user input and retrieving data from the database. To generate queries, the LLM requires the database schema in advance, as well as instructional prompts. We configured the Chain to automatically generate SPARQL queries using Claude based on the input and retrieve relevant building information from GraphDB. This is done by passing to the Chain the ontologies as the input schema of the database and a set of prompts to instruct the LLM. For the prompt definition, we adopted a few-shot prompting strategy [33] and the best practices for writing prompts suggested by Anthropic. Some of these best practices are: 1) the prompt should be specific about the task the LLM is asked to perform; 2) provide context about the task to perform; 3) provide examples of the expected output. Following the guidelines, we defined prompts supplying the LLM with representative query examples that provide explicit guidance on query structure, expected patterns, and constraints. Listing 1.1 shows an excerpt from the prompt we defined to query the knowledge graph. The full set of prompts used in the evaluation is available on the project's GitHub repository.

In the second cycle, we used the Chain *GraphCypherQAChain* from LangChain. Similar to the aforementioned one, this Chain is designed to interact with Neo4j, generating Cypher queries from user input and executing them on the database. Through the already integrated Awesome Procedures On Cypher (APOC) library, the schema of the database can be extracted dynamically. We, thereby, leveraged the Neo4j APOC library to generate a dynamic schema extraction. The prompts we used in the chains were the same used in the first cycle with the only difference that the instructions and examples were converted from SPARQL queries to Cypher queries. Also, in this case the full set of prompts used in the evaluation is available on the project's GitHub repository.

• Claude: https://www.anthropic.com/claude

◇ OpenRouter:https://openrouter.ai/

♢ LangChain: https://www.langchain.com/

To retrieve the data from the IoT sensors, we developed a Tool that executes a custom Chain used by the LLM to retrieve the sensor identifier from the knowledge graph. The LLM then, informed with the identifier, queries InfluxDB to retrieve the actual time-series values. An Agent orchestrates these two components by driving user queries based on the inputted context.

The web user interface has been developed using Streamlit[□], a Python library that supports the user in the interaction with the virtual assistant. The web application implements a chat-like interface through which users can submit queries in natural language and receive responses.

Finally, we configured an LLM-based agent to choose the right capability (or Tool) based on the user query, i.e., selecting the custom chains for either building-related questions or for sensor data questions.

```
You are an expert GraphDB Developer translating user questions into
SPARQL to answer questions about a building and the elements contained
in it. Use only the provided relationship types and properties
in the schema.

Do not use any other relationship types or properties that are
not provided.

Your answers should be concise and to the point. Do not include any
additional information that is not requested. Answer with only the
generated SPARQL statement.

Try to use meaningful aliases for the nodes and relationships in the
query. Here there are some examples of how to respond to the
user's question:

<example>
Tell me about the bathroom in the building
PREFIX bot: <https://w3id.org/bot#>
PREFIX props: <https://w3id.org/props#>
SELECT ?room ?relationship ?value
WHERE {{
  ?room
  props:longNameIfcSpatialStructureElement_attribute_simple ?name.
  FILTER(CONTAINS(?name, "Bathroom"))
  ?room ?relationship ?value
}}

What can be found in the kitchen?
PREFIX default1: <https://w3id.org/bot#>
PREFIX default2: <https://w3id.org/props#>
SELECT ?space ?element
WHERE {{
  ?space
  default2:longNameIfcSpatialStructureElement_attribute_simple ?name.
  FILTER(CONTAINS(?name, "Kitchen"))
  ?space default1:containsElement ?element.
}}
....
```

Listing 1.1. An excerpt from the prompt used to generate SPARQL queries over GraphDB.

[□] Streamlit: https://streamlit.io/

5.2 Artifact Evaluation

To evaluate the effectiveness of the proposed approach, we used the two versions of technical prototypes developed. Namely, the first evaluation focused on RDF-based knowledge graph technology, in which we imported the OSH dataset into GraphDB using its RDF representation, which conforms to the BOT and BOP ontologies. In the second evaluation run, we focused on LPG-based graph technology. The same data was imported into a local instance of Neo4j. For uploading the data into Neo4j, the rdflib_neo4j[♮] library was used. This library is a Python package that allows users to convert RDF data into Linked Property Graphs and upload them to a Neo4j database.

Next, valid for both evaluation runs, the IoT sensor data were imported into the time-series database InfluxDB in a two-step approach: we first converted the input CSV files into the InfluxDB line protocol format and then imported them into the time-series database.

For the **first evaluation run**, we passed both the BOT and BOP ontologies, in TTL[△] format to the chain that automatically had to generate SPARQL queries.

However, this configuration frequently led to exceeding the context window of the LLM. Although Claude 4.5 Sonnet offers a significantly larger context window of up to one million tokens, we found that processing the full dataset remained unfeasible and computationally expensive. Similar to [4], to reduce the number of input tokens, we then passed only the instances and respective relationships of the dataset, thus omitting the schema. This reduced the total token count to under 60,000 and allowed us to define a set of prompt examples to guide the LLM in generating the appropriate SPARQL queries.

We applied a similar strategy in the chain developed to retrieve the information about IoT data. However, Claude consistently failed to generate SPARQL queries to extract the sensor identifiers from the GraphDB, which was required to then query InfluxDB. More in detail, the LLM attempted to retrieve sensor measurements directly from GraphDB rather than the sensor identifiers. We assume that this behavior may be attributed to the excessive amount of contextual information provided, which likely caused the model to misinterpret the user's intent.

To address this issue, we modified the chain to use only the schema, i.e., classes, object properties, and data properties. The corresponding SPARQL query is shown in Listing 1.2. This solution significantly reduced the amount of contextual information passed to the LLM, lowering the input to fewer than 5,000 tokens and enabling the generation of SPARQL queries that correctly extracted the sensor identifiers.

As a result of the adjustments described above, the virtual assistant was able to respond to both building-related and sensor-related queries successfully.

For the **second evaluation run**, we used the prototype version that includes Neo4j graph database. By leveraging the APOC library, the database schema was extracted dynamically, significantly reducing the input overhead to less than 6,000 tokens. However, we observed that Claude faced greater challenges in generalizing the provided few-shot examples for extracting the building-related information. This required a more sophisticated prompt engineering effort with respect to the first prototype. At this stage, the virtual assistant was able to respond to most of the competency questions. The final prompts used for the LLM to generate the correct Cypher and Flux queries to retrieve both building information from Neo4j as well as the sensor data from InfluxDB, can be found in the GitHub repository.

5.3 Results and Comparison

We ran the full set of competency questions, which are relevant to facility managers, against the two technical prototypes to evaluate the approach's effectiveness. In doing so, we simulate the

---

[♮] rdflib-neo4j: https://neo4j.com/labs/rdflib-neo4j/

[△] rdflib-neo4j: https://medium.com/@sharathvyas/turtle-ttl-a-human-readable-format-for-rdf-data-and-shacl-42a154ad6f29 interaction of a facility manager looking for information about the building structure and the data generated by the IoT infrastructure. To verify the correctness of the returned information, the responses were validated against the underlying data sources. In particular, for building-related information, SPARQL queries (in the RDF-based version) and Cypher queries (in the LPG-based version) were manually executed to cross-check the structural and semantic accuracy of the retrieved data. For IoT-related queries, corresponding InfluxDB queries were executed to validate the correctness of the returned sensor measurements.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
CONSTRUCT {
    ?class a rdfs:Class .
    ?class ?objectProperty ?relatedClass .
    ?class ?dataProperty "" .
}
WHERE {
    {
        # Extract classes
        SELECT DISTINCT ?class
        WHERE {
            ?instance a ?class .
        }
    }
    {
        # Extract object properties for each class
        SELECT DISTINCT ?class ?objectProperty ?relatedClass
        WHERE {
            ?instance1 a ?class.
            ?instance2 a ?relatedClass.
            ?instance1 ?objectProperty ?instance2.
        }
    }
    {
        # Extract data properties for each class
        SELECT DISTINCT ?class ?dataProperty
        WHERE {
            ?instance a ?class.
            ?instance ?dataProperty ?dataValue.
            FILTER NOT EXISTS { ?dataValue a ?moreclass.}
        }
    }
}
```

Listing 1.2. SPARQL query used as the schema in the custom chain to extract sensor identifiers.

The evaluation demonstrated that the virtual assistant is able to answer the majority of the competency questions in both the RDF- and LPG-based versions of the technical prototype. However, the different data models also led to distinct LLM-generated query patterns, resulting in differences in the answers. To compare these outcomes, we assigned scores to each response, summarized in Table 1:

- **Correct and Complete** (+ +): The LLM response is accurate.
- **Partially Correct** (+): The LLM response is mostly accurate, but some information is missing.
- **Interpretation Failure** (-): The LLM was unable to interpret the query results, although the generated query was correct.
- **Generation Failure** (- -): The LLM was unable to generate a syntactically or logically correct query.

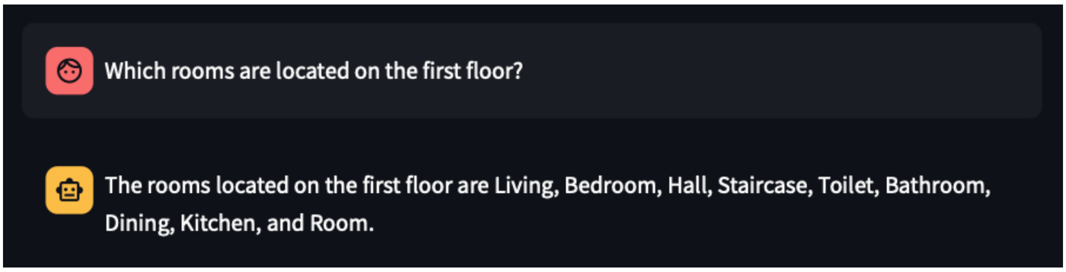
To assign a score to each response, we evaluated both the amount and the usefulness of the information returned by the system. In particular, we assessed whether the provided information was sufficient to validate the results against the underlying data sources. Responses containing more complete, directly usable information required less time and effort for verification and therefore received higher scores. Conversely, responses that were incomplete or required further interpretation received lower scores.

Table 1. Evaluation of the answers obtained using GraphDB against Neo4j

#ID	Competency Question	GraphDB	Neo4j
CQ1	List the rooms located on the first floor	+ +	+ +
CQ2	Information about a room	+	+ +
CQ3	Walls adjacent to a room	+ +	+
CQ4	Details of a specific wall	+	+ +
CQ5	Rooms adjacent to another room	- -	-
CQ6	Count the sensors in the building	- -	- -
CQ7	Count the humidity sensors in the building	- -	+ +
CQ8	Where are the humidity sensors located	- -	- -
CQ9	What is the temperature measured in a specific room	+ +	+ +

The results showed that most of the competency questions were answered correctly. In the following, we describe a few of the competency questions that have been answered correctly. In Section 6, we report on the competency question that the virtual assistant failed to answer or answered incorrectly.

In the first competency question (CQ1), we asked for information about the rooms located on the first floor. In both versions of the prototype, the assistant correctly listed all the rooms present in the dataset (see Figure 3).



Which rooms are located on the first floor?

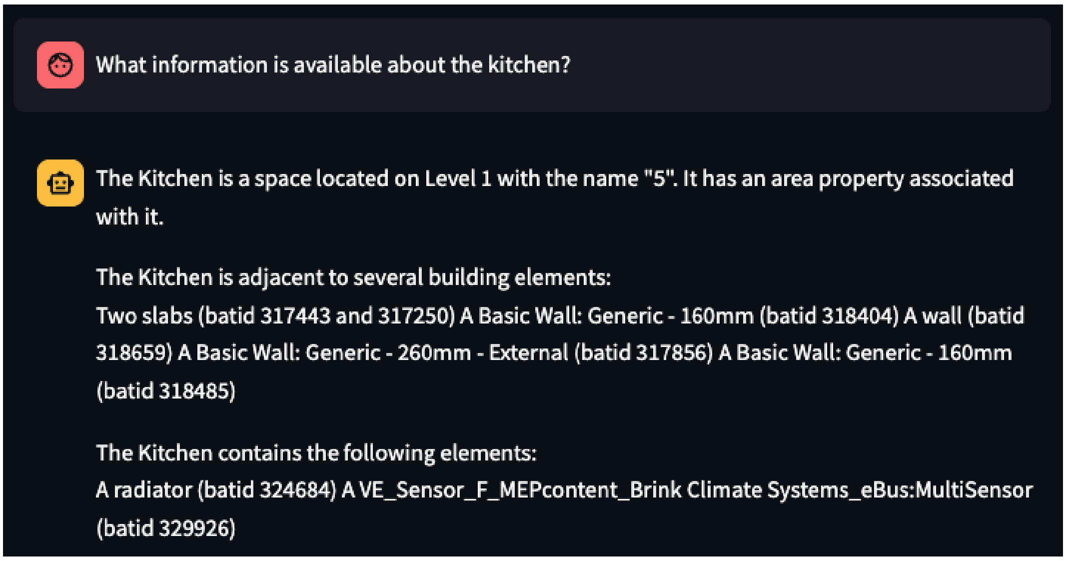
The rooms located on the first floor are Living, Bedroom, Hall, Staircase, Toilet, Bathroom, Dining, Kitchen, and Room.

Figure 3. Correct response to the user query regarding competency question CQ1 “List the rooms located on the first floor” using both RDF-based and LPG-based prototypes.

A similar result was obtained for the second competency question when we requested information about a specific room, such as the kitchen. In this case, using the LPG-based versions of the technical prototype, the virtual assistant accurately retrieved the elements it contained, including instances of IoT sensors (see Figure 4). Instead, when using the RDF-based version of the prototype, the virtual assistant returned less information (see Figure 5).

In the third competency question (CQ3) on retrieving information about the building’s material composition (i.e., details about the walls between the kitchen and the toilet), when using the RDF-based version, the virtual assistant successfully returned complete and correct information (see Figure 6a). When using the LPG-based version, less information was returned for CQ3 (see Figure 6b).

In another competency question (CQ9), we queried information about the values recorded by a specific wall-mounted sensor by specifying a time range. In both versions of the prototype, the virtual assistant successfully responded to the query, providing the user with the temperature measurement and the timestamp of the observation (see Figure 7).



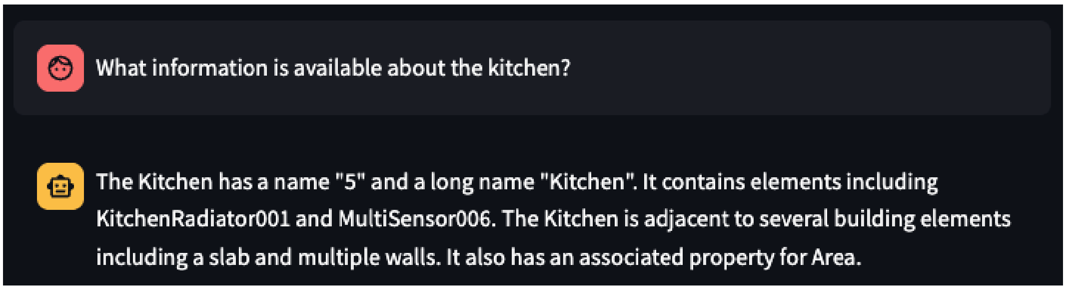
What information is available about the kitchen?

The Kitchen is a space located on Level 1 with the name "5". It has an area property associated with it.

The Kitchen is adjacent to several building elements:
Two slabs (batid 317443 and 317250) A Basic Wall: Generic - 160mm (batid 318404) A wall (batid 318659) A Basic Wall: Generic - 260mm - External (batid 317856) A Basic Wall: Generic - 160mm (batid 318485)

The Kitchen contains the following elements:
A radiator (batid 324684) A VE_Sensor_F_MEPcontent_Brink Climate Systems_eBus:MultiSensor (batid 329926)

Figure 4. Correct response to the user query corresponding to competency question CQ2 “Information about the kitchen room” obtained using the LPG-based prototype.



What information is available about the kitchen?

The Kitchen has a name "5" and a long name "Kitchen". It contains elements including KitchenRadiator001 and MultiSensor006. The Kitchen is adjacent to several building elements including a slab and multiple walls. It also has an associated property for Area.

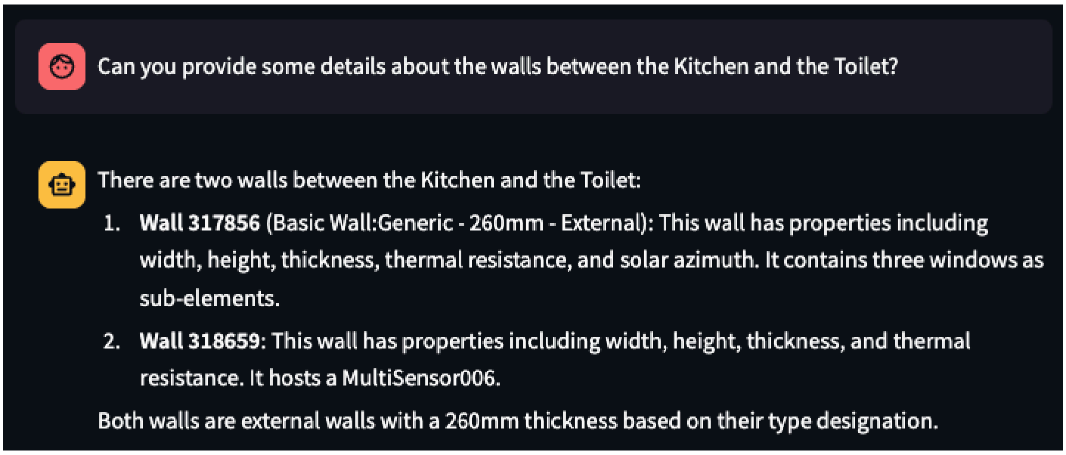
Figure 5. Correct response to the user query corresponding to competency question CQ2 “Information about the kitchen room” obtained using the RDF-based prototype.

6 Discussion and Limitations

Given the positive results of the evaluation, we regard the proposed hybrid LLM–Knowledge Graph architecture as a promising approach to overcoming integration issues between BIM and IoT in a scalable manner, thereby supporting facility managers in real-time information retrieval.

Although the evaluation was conducted on a laboratory dataset (OSH dataset), it provides an initial validation of the feasibility of the proposed architecture.

A further limitation concerns the level of stakeholder involvement in the design and evaluation phases of this study. The competency questions used to assess the effectiveness of the two prototypes were derived from domain literature and expert knowledge rather than from a structured elicitation process involving facility managers or building owners. While this approach ensures grounding in established domain requirements, it may not fully capture the information needs and decision-making priorities of end users in real operational contexts. Larger stakeholder participation in future iterations of this research, for instance, through co-design workshops or field evaluations in live building environments, could further refine the artifact.



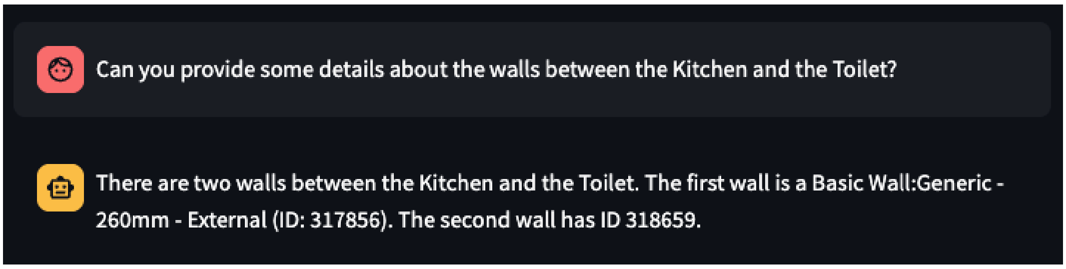
Can you provide some details about the walls between the Kitchen and the Toilet?

There are two walls between the Kitchen and the Toilet:

1. **Wall 317856** (Basic Wall:Generic - 260mm - External): This wall has properties including width, height, thickness, thermal resistance, and solar azimuth. It contains three windows as sub-elements.
2. **Wall 318659**: This wall has properties including width, height, thickness, and thermal resistance. It hosts a MultiSensor006.

Both walls are external walls with a 260mm thickness based on their type designation.

(a) Complete response to competency question CQ3 obtained using the RDF-based prototype.



Can you provide some details about the walls between the Kitchen and the Toilet?

There are two walls between the Kitchen and the Toilet. The first wall is a Basic Wall:Generic - 260mm - External (ID: 317856). The second wall has ID 318659.

(b) Partially complete response to competency question CQ3 obtained using the LPG-based prototype.

Figure 6. Comparative analysis of responses to competency question CQ3 "Details of a specific element of the building structure" using both the RDF-based and the LPG-based prototypes.

What is the most recent temperature recorded in the kitchen during the available time period?

	Measurement	Value	Date
0	Temperature	21.2600	2017-06-06 06:05:51

Figure 7. Correct response to the user query corresponding to competency question CQ9 "What is the temperature measured in a specific room," obtained using both the RDF-based and the LPG-based prototypes.

Compared to the approach in [12], which uses a Python script for the Thermal Comfort calculation, our approach enables extension to various use cases, as reflected in the competency questions. For new, unforeseen scenarios or questions where the answers are unsatisfactory, natural-language system prompts can be adjusted, providing higher flexibility than rebuilding or adapting a fixed, dedicated script.

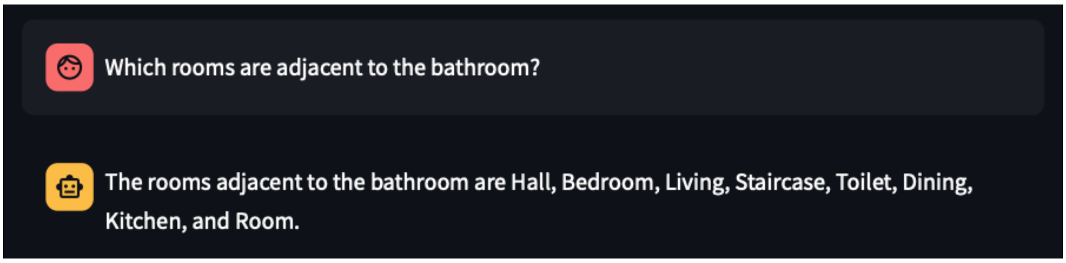
The virtual assistant interacts with a user in natural language, i.e., it interprets questions, generates appropriate queries for the databases, retrieves information meaningfully, and presents the results to the user in natural language.

When querying information about a room, the virtual assistant using Neo4j and the LPG returned more properties than the RDF graph-based version. This is mainly due to the fact that in the LPG we stored sub-details bundled inside the node (i.e., a JSON-like map inside a property). While this is supported by LPG and our use case, it is not considered a best practice due to maintainability and searchability concerns. Each level of nesting complicates queries and negatively impacts readability, maintenance, and processing speed [34].

On the contrary, while the RDF graph-based version followed best practices (i.e., instead of nesting properties within a node, we represented nested data in separate nodes connected by relationships), it required multiple SPARQL queries to retrieve the same properties. The system prompt also required more examples compared to the LPG graph-based version to generate correct queries. All system prompts can be found in the project repository[▽].

Moreover, during the conversion from RDF to LPG (automatically done with the rdflib_neo4j plugin of Neo4j), nodes in Neo4j were labeled with a common class in addition to their specific class, leading to side effects in which the LLM assumed certain properties were universally applicable across all node types. This resulted in incorrect Cypher queries being generated. Although this issue can be resolved by refining the conversion process, it emphasizes the importance of maintaining clear semantic definitions and distinctions between different classes (i.e., entity types).

Both versions of the prototypes failed to retrieve information where direct relationships were not present in the graphs. In particular, when we asked questions involving building structures, such as identifying rooms adjacent to another room, the virtual assistant based on the RDF prototype failed to answer correctly (see Figure 8). In fact, in the given building structure, rooms are not directly connected; instead, they are linked through intermediary elements such as walls and slabs. To address this issue, an ontology engineering work is required. When we executed the same query using the LPG-based prototype, the virtual assistant returned an empty response. We observed that, although the LLM had generated a syntactically correct Cypher query, it was unable to interpret the results due to the complexity of the properties returned by the traversal.



Which rooms are adjacent to the bathroom?

The rooms adjacent to the bathroom are Hall, Bedroom, Living, Staircase, Toilet, Dining, Kitchen, and Room.

Figure 8. Incorrect response to the user query corresponding to competency question CQ5 “Rooms adjacent to another room,” obtained using the RDF-based prototype. In the dataset, only the Hall, Bedroom, and Living room are adjacent to the Bathroom.

To complement the qualitative assessment discussed in Section 5.2, we conducted a quantitative evaluation to analyze the robustness of the two prototypes. We reported in Table 2 the results obtained by executing each competency question five times. For each run, we recorded: (i) the total number of tokens consumed, computed as the sum of input tokens in the prompt and output tokens generated by the LLM; and (ii) the end-to-end response time, measured from user query submission to the final response produced by the prototype. Finally, in all runs, we set the temperature parameter of the LLM to 0 to ensure deterministic outputs in the responses. With this setting, both prototype implementations returned the same textual responses across all five runs for each competency question, indicating stable and reproducible behavior. Moreover, no failures were observed during repeated executions, and both implementations consistently produced a response for the user.

In Table 2, we report the mean and standard deviation of the recorded metrics across the five runs. The standard deviation indicates stability by capturing the variability in token consumption and response time under identical execution conditions. The results show that the GraphDB-based implementation generally consumes more tokens than the Neo4j-based implementation (in 7 out

---

[▽] https://github.com/PROSLab/BIM-IoT-Assistant/tree/main/prompts of 9 competency questions). However, in two cases (CQ2 and CQ4), Neo4j exhibited higher token consumption. In terms of variability, both systems demonstrate comparable behavior, with low standard deviation values across most competency questions. Regarding response time, the GraphDB-based implementation is, on average, slightly slower than the Neo4j-based implementation. Nevertheless, the overall response times remain within a similar range for both systems. In this respect as well, both implementations exhibit stable performance, with limited dispersion across repeated executions.

Table 2. Quantitative evaluation over five runs per competency question

#ID	Backend	Avg Tokens	Std Dev (tokens)	Avg Time (s)	Std Dev (s)
CQ1	GraphDB	8548	0.00	9.56	0.72
	Neo4j	7250	0.00	6.94	0.48
CQ2	GraphDB	9318	0.00	10.28	0.15
	Neo4j	11661	10.64	10.52	0.48
CQ3	GraphDB	11658	13.86	12.99	0.53
	Neo4j	6471	0.00	7.50	0.58
CQ4	GraphDB	9694	0.00	11.18	0.34
	Neo4j	20765	0.00	8.41	0.51
CQ5	GraphDB	8854	0.00	9.27	0.44
	Neo4j	5931	0.00	6.53	0.51
CQ6	GraphDB	8211	0.00	6.96	0.41
	Neo4j	5880	0.00	5.18	0.52
CQ7	GraphDB	8263	8.44	6.95	0.47
	Neo4j	5904	0.00	5.21	0.54
CQ8	GraphDB	8248	0.00	7.29	0.47
	Neo4j	5893	3.58	6.49	0.36
CQ9	GraphDB	6338	1.79	9.52	0.87
	Neo4j	6129	1.79	6.82	1.58

Overall, the quantitative analysis provides further evidence of the robustness and operational consistency of the proposed hybrid LLM–Knowledge Graph architecture. Although currently stand-alone, the resulting solution can be integrated into 3D floor plan applications to further support facility managers in decision-making tasks.

However, effective deployment of the system in real-world building management scenarios requires careful consideration of relevant non-functional requirements, including cost, data privacy, and data security. For instance, using an LLM incurs extra operational costs related to token consumption. Therefore, a comprehensive cost analysis must consider the scale of deployment, query frequency, and the pricing model of the chosen LLM provider. Using cloud-based LLM services to process sensitive building data raises important privacy concerns, as this data may require anonymization, controlled access mechanisms, or on-premises LLM deployment solutions.

Finally, to ensure safe integration into operational workflows, the proposed architecture must be complemented with appropriate security measures related to API access, authentication, and data transmission.

7 Conclusion

In this article, we propose a novel hybrid artificial intelligence approach that combines a structured and ontology-based BIM representation with IoT-sensed data, leveraging LLM capabilities.

We adopted a DSR strategy to address the research question of how such a combination can support information retrieval for facility managers, who are focal stakeholders in the Smart Building paradigm. The proposed architecture has been implemented in two versions of a technical prototype: one RDF-based and one LPG-based, both having a virtual assistant as a user interface. For the artifact design and evaluation, we considered the open OSH RDF dataset, which is a suitable real-world case for BIM and IoT integration. As a result, the approach proved effective for information retrieval, with most competency questions correctly answered.

The results of our evaluation demonstrated that LPGs, because of their capability to represent nested properties into nodes, can provide less expensive data access (i.e., less refinement of prompts), particularly for queries that involve retrieving complex entity properties or traversing relationships. In practice, they returned all properties of an entity in a single query, making it an attractive option for dynamic, real-time queries where performance and simplicity are key. On the other hand, RDF graph-based version of the prototype proved to be more suitable for scenarios that require reasoning to infer additional knowledge, a more precise semantic representation and scalability. However, this comes at the cost of more refinement of prompts for complex query generation whitin the LLM framework.

Based on our results, several directions for future work are possible, and we group them into two main categories: (1) further application of the proposed architecture, (2) extending the architecture. Below, we elaborate on both.

(1) Further application of the proposed architecture: First, we intend to replicate the evaluation by considering additional real-world datasets as ground truth. This will help to assess further the benefits and generalizability of the proposed hybrid AI architecture.

Second, we plan to integrate and compare the performance of other LLM models to analyze how their specific capabilities influence information retrieval tasks. We also plan to involve facility managers and relevant stakeholders in the evaluation process to assess usability and the quality of decision support within real operational workflows.

(2) Extending the architecture: Given the evolution of LLMs towards agents and workflows (often referred to as Agentic AI), we aim to go beyond information retrieval and leverage Agentic AI to increase the complexity of use cases that can be regarded as useful by facility managers, such as the indoor environmental quality of the building. As a concrete action, we plan to integrate the proposed hybrid LLM–Knowledge Graph architecture into an existing Digital Twin platform defined to support rescue operations in emergency scenarios, such as earthquakes [35], [36]. By enabling intuitive access to both structural and real-time data, the proposed approach will assist emergency responders in understanding building conditions, assessing risks, and supporting informed decision-making during rescue operations.

Moreover, we intend to continue promoting the user-centric approach of Digital Twin platforms of buildings for domain experts by supplementing the proposed architecture with the ontology-based meta-modeling approach proposed in [6]. This will allow domain experts to structure knowledge into ontologies without requiring ontology expertise.

References

[1] X. Huang, Y. Liu, L. Huang, E. Onstein, and C. Merschbrock, "BIM and IoT data fusion: The data process model perspective," *Automation in Construction*, vol. 149, 2023. Available: https://doi.org/10.1016/j.autcon.2023.104792

[2] M. Jia, A. Komeily, Y. Wang, and R. S. Srinivasan, "Adopting Internet of Things for the development of smart buildings: A review of enabling technologies and applications," *Automation in Construction*, vol. 101, pp. 111–126, 2019. Available: https://doi.org/10.1016/j.autcon.2019.01.023

[3] L. Chamari, E. Petrova, and P. Pauwels, "A web-based approach to BMS, BIM and IoT integration: a case study," *CLIMA 2022 Conference*, 2022. Available: https://proceedings.open.tudelft.nl/clima2022/article/view/228

[4] M. Callisto De Donato, E. Laurenzi, and D. Porumboiu, "A hybrid artificial intelligence to support information retrieval in smart buildings," in *Joint Proceedings of the BIR 2025 Workshops and Doctoral Consortium*, vol. 4034, 2025, pp. 157–169. Available: https://ceur-ws.org/Vol-4034/paper79.pdf

[5] A. Hevner and S. Chatterjee, "Design science research in information systems," in *Design Research in Information Systems. Integrated Series in Information Systems*, vol. 22, 2010, pp. 9–22. Available: https://doi.org/10.1007/978-1-4419-5653-8_2

[6] E. Laurenzi, J. Allan, N. Campos, and S. Stoller, “An Ontology-Based Meta-modelling Approach for Semantic-Driven Building Management Systems,” in *Advanced Information Systems Engineering Workshops. CAiSE 2024. Lecture Notes in Business Information Processing*, vol. 521, 2024, pp. 200–211. Available: https://doi.org/10.1007/978-3-031-61003-5_18

[7] B. Pang, J. Shi, L. Jiang, and Z. Pan, “A Semantic Approach to Dynamic Path Planning for Fire Evacuation through BIM and IoT Data Integration,” *Advances in Civil Engineering*, vol. 2024, 2024. Available: https://onlinelibrary.wiley.com/doi/10.1155/2024/8839865

[8] L. Chamari, E. Petrova, and P. Pauwels, “An End-to-End Implementation of a Service-Oriented Architecture for Data-Driven Smart Buildings,” *IEEE Access*, vol. 11, pp. 117 261–117 281, 2023. Available: https://doi.org/10.1109/ACCESS.2023.3325767

[9] J. Bjørnskov and M. Jradi, “An ontology-based innovative energy modeling framework for scalable and adaptable building digital twins,” *Energy and Buildings*, vol. 292, article 113146, 2023. Available: https://doi.org/10.1016/j.enbuild.2023.113146

[10] M. Shahinmoghadam and A. Motamedi, “An Ontology-Based Mediation Framework for Integrating Federated Sources of BIM and IoT Data,” in *Proceedings of the 18th International Conference on Computing in Civil and Building Engineering. ICCCBE 2020. Lecture Notes in Civil Engineering*, vol. 98, 2020, pp. 907–923. Available: https://doi.org/10.1007/978-3-030-51295-8_63

[11] K. Janowicz, M. H. Rasmussen, M. Lefrançois, G. F. Schneider, and P. Pauwels, “BOT: The building topology ontology of the W3C linked building data group,” *Semantic Web: – Interoperability, Usability, Applicability*, vol. 12, no. 1, pp. 143–161, 2020. Available: https://doi.org/10.3233/SW-200385

[12] A. J. Donkers, D. Yang, B. de Vries, and N. H. Baken, “Real-time building performance monitoring using semantic digital twins,” in *9th Linked Data in Architecture and Construction Workshop, LDAC 2021*, vol. 3081, 2021, pp. 55–66. Available: https://ceur-ws.org/Vol-3081/05paper.pdf

[13] F. Lygerakis, N. Kampelis, and D. Kolokotsa, “Knowledge Graphs’ Ontologies and Applications for Energy Efficiency in Buildings: A Review,” *Energies*, vol. 15, no. 20, 2022. Available: https://doi.org/10.3390/en15207520

[14] A. Donkers, D. Yang, B. de Vries, and N. Baken, “Semantic Web Technologies for Indoor Environmental Quality: A Review and Ontology Design,” *Buildings*, vol. 12, no. 10, article 1522, 2022. Available: https://doi.org/10.3390/buildings12101522

[15] D. D. Eneyew, M. A. Capretz, and G. T. Bitsuamlak, “Toward Smart-Building Digital Twins: BIM and IoT Data Integration,” *IEEE Access*, vol. 10, pp. 130 487–130 506, 2022. Available: https://doi.org/10.1109/ACCESS.2022.3229370

[16] K. El Mokhtari, I. Panushev, and J. J. McArthur, “Development of a Cognitive Digital Twin for Building Management and Operations,” *Frontiers in Built Environment*, vol. 8, pp. 1–18, 2022. Available: https://doi.org/10.3389/fbuil.2022.856873

[17] C. Ramonell, R. Chacón, and H. Posada, “Knowledge graph-based data integration system for digital twins of built assets,” *Automation in Construction*, vol. 156, 2023. Available: https://doi.org/10.1016/j.autcon.2023.105109

[18] J. Shi, Z. Pan, L. Jiang, and X. Zhai, “An ontology-based methodology to establish city information model of digital twin city by merging BIM, GIS and IoT,” *Advanced Engineering Informatics*, vol. 57, no. 2022, article 102114, 2023. Available: https://doi.org/10.1016/j.aei.2023.102114

[19] I. Fatokun, A. R. N. Sheela, T. Mecharnia, M. Lefrançois, V. Charpenay, F. Badeig, and A. Zimmermann, “Modular Knowledge integration for Smart Building Digital Twins,” in *Proceedings of the 11th Linked Data in Architecture and Construction Workshop*, vol. 3633, 2023, pp. 123–138. Available: https://ceur-ws.org/Vol-3633/paper10.pdf

[20] C. Schmid, S. Acero Gonzalez, S. Stoller, W. Willuhn, E. Laurenzi, and J. Allan, “Continuous calculation of key performance indicators for buildings through an application layer connected to a knowledge graph,” in *LDAC 2025 Linked Data in Architecture and Construction*, vol. 2636, 2025, pp. 108–121. Available: https://ceur-ws.org/Vol-4113/paper8.pdf

[21] A. J. A. Donkers, D. Yang, and N. Baken, “Linked data for smart homes: Comparing RDF and labeled property graphs,” in *LDAC 2020 Linked Data in Architecture and Construction*, vol. 2636, 2020, pp. 23–36. Available: http://ceur-ws.org/Vol-2636/02paper.pdf

[22] A. d’Avila Garcez and L. C. Lamb, “Neurosymbolic AI: the 3rd wave,” *Artificial Intelligence Review*, vol. 56, no. 11, pp. 12 387–12 406, 2023. Available: https://doi.org/10.1007/s10462-023-10448-w

[23] R. Buchmann, J. Eder, H.-G. Fill, U. Frank, D. Karagiannis, E. Laurenzi, J. Mylopoulos, D. Plexousakis, and M. Y. Santos, “Large language models: Expectations for semantics-driven systems engineering,” *Data & Knowledge Engineering*, vol. 152, article 102324, 2024. Available: https://doi.org/10.1016/j.datak.2024.102324

[24] G. F. Schneider and M. H. Rasmussen, “TechnicalBuildingSystems/OpenSmartHomeData: First release of Open Smart Home Data Set,” 2018. Available: https://doi.org/10.5281/zenodo.1244602

[25] G. Schneider, M. Rasmussen, P. Bonsma, J. Oraskari, and P. Pauwels, “Linked building data for modular building information modelling of a smart home,” in *eWork and eBusiness in Architecture, Engineering and Construction*, 2018, pp. 407–414.

[26] N. Noy and D. Mcguinness, “Ontology development 101: A guide to creating your first ontology,” *Knowledge Systems Laboratory*, vol. 32, 01 2001.

[27] P. Dhameniya and R. Yadav, “12 – knowledge graph-based question answering (KG-QA) using natural language processing,” in *Knowledge Graph-Based Methods for Automated Driving*, 2025, pp. 227–249. Available: https://doi.org/10.1016/B978-0-443-30040-0.00012-6

[28] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, “Retrieval-augmented generation for knowledge-intensive NLP tasks,” in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 9459–9474. Available: https://proceedings.neurips.cc/paper_files/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf

[29] A. Gupta, S. Tyagi, N. Panwar, S. Sachdeva, and U. Saxena, “NoSQL databases: Critical analysis and comparison,” in *2017 International Conference on Computing and Communication Technologies for Smart Nation (IC3TSN)*. IEEE, 2017, pp. 293–299. Available: https://doi.org/10.1109/IC3TSN.2017.8284494

[30] S. Tang, D. R. Shelden, C. M. Eastman, P. Pishdad-Bozorgi, and X. Gao, “A review of building information modeling (BIM) and the internet of things (IoT) devices integration: Present status and future trends,” *Automation in Construction*, vol. 101, pp. 127–139, 2019. Available: https://doi.org/10.1016/j.autcon.2019.01.020

[31] B. Shah, P. M. Jat, and K. Sashidhar, “Performance study of time series databases,” *arXiv preprint arXiv:2208.13982*, 2022. Available: https://arxiv.org/abs/2208.13982

[32] S. Munir and A. Aldini, “Towards evaluating large language models for graph query generation,” *arXiv preprint arXiv:2411.08449*, 2024. Available: https://arxiv.org/abs/2411.08449

[33] P. Sahoo, A. K. Singh, S. Saha, V. Jain, S. Mondal, and A. Chadha, “A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications,” 2025. Available: https://arxiv.org/abs/2402.07927

[34] K. Rabuzin, M. Cerjan, and S. Križanić, “Supporting Data Types in Neo4j,” in *New Trends in Database and Information Systems. ADBIS 2022. Communications in Computer and Information Science,*, vol. 1652, 2022, pp. 459–466. Available: https://doi.org/10.1007/978-3-031-15743-1_42

[35] M. Callisto De Donato, F. Corradini, F. Fornari, B. Re, and M. Romagnoli, “Enabling 3D Simulation in ThingsBoard: A First Step Towards A Digital Twin Platform,” in *Enterprise Design, Operations, and Computing. EDOC 2023 Workshops. EDOC 2023. Lecture Notes in Business Information Processing*, vol. 498, 2024, pp. 325–330. Available: https://doi.org/10.1007/978-3-031-54712-6_20

[36] M. Callisto De Donato, F. Corradini, F. Fornari, and B. Re, “SAFE: An ICT platform for supporting monitoring, localization and rescue operations in case of earthquake,” *Internet of Things*, vol. 27, article 101273, 2024. Available: https://doi.org/10.1016/j.iot.2024.101273