Enabling Natural Language Access to BIM Models with AI and Knowledge Graphs

Andrea Ibba[1,2,†], Rubén Alonso[2,3,†] and Diego Reforgiato Recupero[1,2,*,†]

[1]Department of Mathematics and Computer Science, Università degli Studi di Cagliari, Via Ospedale 72, 09124 Cagliari, Italy
[2]ICT Division, R2M Solution s.r.l., 42, 27100 Pavia, Italy
[3]Programa de Doctorado, Centro de Automática y Robótica, Universidad Politécnica de Madrid-CSIC, Madrid, Spain

Abstract

Building Information Modeling (BIM) centralizes project data within a unified digital framework, enhancing collaboration across the Architecture, Engineering, Construction, and Operation (AECO) sector stakeholders. However, querying BIM data remains challenging due to the complexity of formats such as Industry Foundation Classes (IFC), which require specialized expertise. Existing tools provide limited functionality when attempting to extract information through natural language interactions, while Large Language Models (LLMs) struggle with IFC data due to its scale and complex relationships. The proposed approach addresses these limitations by integrating LLMs and knowledge graphs (KGs) to facilitate natural language queries. By structuring BIM data as a KG prior to LLM processing, we are able to enhance the extraction of knowledge while preserving semantic integrity. Evaluated on a multi-storey building, our approach demonstrates the potential of graph-based AI for BIM analysis.

Keywords

Building Information Modeling, Large Language Models, Artificial Intelligence, Knowledge Graphs, Semantic Web, Retrieval-Augmented Generation

1. Introduction

Building Information Modeling (BIM) has significantly advanced the Architecture, Engineering, Construction, and Operation (AECO) industry by consolidating project data into a structured digital representation. It enhances collaboration across all project phases, from design to facility management, by incorporating diverse types of information, such as geometry, scheduling, costs, and sustainability aspects [1]. However, despite its advantages, extracting meaningful insights from BIM models remains a challenge due to their complexity, high data volume, and reliance on structured formats like Industry Foundation Classes (IFC) [2]. Existing tools, including Autodesk Revit[1], Navisworks[2], or Solibri[3], facilitate querying and analysis but require specialized expertise, making them less accessible to non-experts.

Advancements in artificial intelligence, particularly in Large Language Models (LLMs), offer promising approaches to improve data retrieval from BIM models. However, directly applying LLMs to IFC files has proven ineffective due to BIM data's extensive size and non-sequential nature, which limits their ability to establish semantic connections between elements. While these models can extract elementary numerical information, they struggle with reasoning over complex interdependencies between building components. To address this, we propose an approach that combines AI-driven reasoning with structured data representations to enhance the accessibility of BIM information through natural language queries. By transforming BIM data into a structured graph before applying AI-based processing, this approach

Third International Workshop on Semantic Technologies and Deep Learning Models for Scientific, Technical and Legal Data SemTech4STLD June 1st or 2nd, 2025 / Portoroz, Slovenia

*Corresponding author.
†These authors contributed equally.
andrea.ibba.uni@gmail.com (A. Ibba); ruben.alonso@r2msolution.com (R. Alonso); diego.reforgiato@unica.it (D. R. Recupero)

[1]https://www.autodesk.com/products/revit/
[2]https://www.autodesk.com/products/navisworks/
[3]https://www.solibri.com/

ensures that essential relationships between elements are maintained, facilitating more effective query resolution. The primary contributions of this research are:

- A system framework that processes natural language questions and retrieves relevant BIM-related information from IFC files.
- A two-step LLM-driven approach: first, by employing prompt engineering techniques we translate user queries into structured queries, and second, reasoning techniques are applied to enhance response accuracy.
- An experimental evaluation conducted on a real-world multi-storey building, demonstrating the system's effectiveness in retrieving structured BIM data from a natural language question.

This research is conducted within the scope of Digital Building Logbooks (DBLs) under the CHRONICLE[4] and LEGOFIT[5] projects, both of which focus on digitalization and sustainability in the built environment [3]. The remainder of this paper is structured as follows: Section 2 reviews related research, Section 3 provides background on BIM data representation and graph-based transformations, and Section 4 details the proposed approach. The experimental setup and the obtained results are illustrated in Section 5, whereas conclusions are drawn in Section 6.

2. Related Works

Researchers have explored a variety of strategies to enhance Natural Language Processing (NLP) applications within BIM, with a particular focus on semantic enrichment and comprehensive data retrieval. A wide-ranging review provided in [4] surveys the literature on augmenting BIM models with meaningful semantics, examining existing IFC-based methods and web-ontology approaches.

A systematic study in [5] assesses the use of NLP under Industry 4.0 principles, identifying key datasets, technologies, and methodological gaps by reviewing 91 articles. The authors highlight the persistent issue of data isolation in research and recommend cross-disciplinary methods, such as unified frameworks and pretrained neural networks, to bridge these gaps.

Further integrating NLP into construction workflows, the authors in [6] developed a virtual assistant for handling textual queries related to BIM and IFC models, achieving notable performance over multiple test queries and diverse datasets. Another framework, introduced in [7], employs machine learning classification of question types, combined with syntactic and semantic analyses, to extract user-relevant details from BIM models through a specialized Navisworks application.

A graph-based approach is presented in [8], which formulates constraints and keywords for user requirements, mapping them to IFC properties via the International Framework for Dictionaries (IFD). This technique leverages pathfinding to link user queries with the IFC schema. Additionally, the work reported in [9] leverages a modular ontology and an ontology-supported parser to translate multi-constraint textual questions into SPARQL statements, obtaining a high accuracy rate on natural language queries for BIM data. Likewise, [10] proposes an intelligent dialogue system that combines BERT-based language models and natural language generation to retrieve specific attribute information from IFC models.

A recent review [11] further supports the motivation of this study, highlighting the intermediate level of data readiness in BIM environments for AI integration and identifying graph-based data preparation as a promising method for elucidating relationships within IFC models.

The approaches presented in the cited works collectively indicate that making BIM models more accessible through various techniques, ranging from virtual assistants to ontology-driven interfaces, is a scientifically promising direction. These studies served as a foundation and inspiration, confirming the relevance of enhancing human, BIM interaction through NLP and semantic methods. In particular, the recurring use of graph-based representations suggested their potential for capturing complex relationships inherent in IFC models. Building on these insights, our work adopts similar principles

[4]https://doi.org/10.3030/101069722
[5]https://doi.org/10.3030/101104058

but extends them through a dedicated pipeline that integrates graph-based BIM transformation with LLM-driven prompt engineering.

3. Background

BIM models can rely on open or closed data formats. Open formats and open standards (such as IFC) foster interoperability and efficient data exchange. In this research, we adopt an Open BIM[6] approach to leverage these advantages. Open BIM has gained increasing prominence in the AECO industry because it promotes data exchange based on shared standards and non-proprietary formats [12]. One of the most recognized formats within Open BIM is the IFC, which provides a rich data model for describing building components and their attributes [13]. The decision to work with IFC stems from its interoperability: it allows a wide array of design, analysis, and management software tools to exchange information without locking stakeholders into specific proprietary ecosystems. Consequently, open standards accelerate collaboration by lowering technical barriers and enhancing transparency, making IFC an appealing choice for creating comprehensive digital representations of buildings. In this work, we employed an IFC model containing both geometric and spatial information about a multi-storey office building. Models like the one used in this work are typically created through CAD-to-BIM processes, where existing computer-aided design files are manually converted into BIM representations. Models derived from photogrammetry and LiDAR-based Scan2BIM workflows are also frequently used, by capturing real-world environments and converting them into precise digital representations. These approaches yield a robust and detailed source of building data, suitable for subsequent semantic processing. To streamline access, we transformed IFC data into a KG using the IFC-to-LBD converter [14]. This conversion preserves critical building elements such as walls, floors, and openings as well as their interrelationships, while casting the model into a more semantic format that can be queried via SPARQL.

The IFC model is publicly available[7], along with its conversion to a KG representation[8].

4. The Proposed Approach

The input of our approach consists of an IFC file representing a construction asset alongside a user's natural language question. The goal is to convert the building data into a knowledge graph (KG), apply prompt engineering techniques to formulate SPARQL queries for data retrieval, and then generate a concise, contextually relevant answer derived from these results.

In our approach, we use a variety of prompt techniques to support complex reasoning and data extraction tasks from the KG. These include instruction prompting, chain-of-thought prompting, and decomposed prompting. Each technique is tailored to a specific stage in the pipeline to improve clarity, accuracy, and semantic alignment with the KG. For executing these prompts, we utilized the open-source model GPT-4o Mini, which offered a suitable trade-off between reasoning capability and computational efficiency.

In the first stage, the IFC file undergoes conversion into a KG using the IFC-to-LBD converter mentioned in Section 3. This transformation maps building components (such as walls, doors, or floors) to graph entities, while relevant attributes and relationships are encoded as RDF triples. The resulting KG is then stored in GraphDB[9], a database optimized for managing and querying semantic data.

Once the KG is available, the user's query is processed through an LLM to generate SPARQL statements. Specifically, decomposed prompting engineering approaches ensure that the question is first analyzed for complexity, potentially dividing it into smaller sub-questions if multiple entities are involved[10]. This decomposition is important because complex queries often involve multiple relationships and

[6]https://www.buildingsmart.org/about/openbim/
[7]https://github.com/aibba19/ASK-BIM/tree/main/IFC-to-LBD%20Conversion/IFC%20model
[8]https://github.com/aibba19/ASK-BIM/tree/main/IFC-to-LBD%20Conversion/LBD%20Model
[9]https://graphdb.ontotext.com/
[10]https://github.com/aibba19/ASK-BIM/blob/main/prompts/simplify_question.py

properties that need to be retrieved separately before reasoning over them. The system ensures that the question is broken down into the smallest number of sub-questions necessary to retrieve all relevant data efficiently. This structured approach leads to the generation of a minimal set of SPARQL queries, each targeting specific data points. For example, if a query involves comparing the dimensions of walls and windows, it would be split into two sub-questions: one retrieving wall dimensions and another retrieving window dimensions. This ensures that each sub-question focuses on a distinct entity.

This approach accommodates both simple questions, which focus on a single type of building element or property, and more complex queries requiring retrieval and comparison of different types of data (for example, dimensions of windows relative to walls).

Each sub-question is then mapped to the relevant KG classes and relationships, providing the LLM with the contextual elements needed to construct accurate SPARQL queries.

In this prompt, we use a matching technique that leverages instruction prompting to guide the model in selecting the most relevant building element classes from a predefined list of available classes extracted from the KG. It then instructs the model to match key terms or implied concepts in the question (e.g., "height of doors") with related classes (e.g., Door) based on their relevance.[11]

Thanks to this matching process, we can retrieve all properties and values associated with the key entities present in the question, ensuring that the necessary contextual information is available for further processing.

In the next step, a separate prompt is used to construct the actual SPARQL queries. The previously extracted classes, properties and values from the KG are provided as contextual input to this prompt, allowing the LLM to generate meaningful and well-structured queries.

The SPARQL generation prompt[12] instructs the model to generate SPARQL queries based on the provided classes, properties, and prefixes. It emphasizes simplicity, the inclusion of specific identifiers, and adherence to given constraints. By clearly delineating the requirements and structure of the desired output, the model is guided to produce accurate and relevant queries.

Once the LLM has created suitable SPARQL statements, they are executed against the SPARQL endpoint to obtain the necessary data. The retrieval process is designed to yield a broad set of candidate elements and property values, especially in more involved questions where calculations or comparisons are required.

After retrieving the raw results, our approach employs the LLM once more to interpret these outcomes and formulate a concise, contextually relevant answer in natural language to the original user's question. This step involves synthesizing the retrieved data and applying reasoning over multiple results when necessary. The model is prompted with both the original user question and the structured output from the SPARQL queries[13], allowing it to refine the final response based on the retrieved information. This not only enhances the clarity of the answer but also ensures that relationships between different queried elements are properly understood.

5. Experimental Settings

We defined 28 questions by consulting architects, engineers, and BIM modelers, aiming to cover both basic data retrieval and more advanced inference needs from an IFC model. The questions range from straightforward element counts to queries requiring additional reasoning, ensuring a balanced evaluation of our approach's capabilities.

The defined questions are designed to be broadly applicable across various types of buildings, without relying on prior knowledge of a specific structure. Their formulation targets common architectural elements and relationships typically present in most BIM models, making them suitable for general-purpose evaluation. Nonetheless, despite their intended generality, not all questions are applicable to every building instance. For example, queries referring to upper floors, such as those involving a

[11]https://github.com/aibba19/ASK-BIM/blob/main/prompts/identify_classes.py
[12]https://github.com/aibba19/ASK-BIM/blob/main/prompts/generate_SPARQL.py
[13]https://github.com/aibba19/ASK-BIM/blob/main/prompts/final_answer.py

"second floor", cannot be meaningfully applied to single-storey buildings. Still, the overall design of the questions prioritizes generalizability and avoids dependency on unique features or naming conventions specific to the building used in our evaluation.

We categorized these questions by complexity, based on the number of IFC entities involved, the relationships between them, the properties required, and the need for inference by the LLM. Higher-complexity queries often demand recognizing multiple entities, retrieving a wider array of properties, and interpreting results through logical connections. To assess the approach thoroughly, we divided the validation process into three main steps. First, we examined how effectively the system breaks down each user query into smaller, entity-focused sub-questions. Next, we evaluated the accuracy of the generated SPARQL queries, ensuring they were both syntactically and logically valid. Finally, we assessed the LLM's ability to integrate and reason over the retrieved data, producing coherent, user-friendly answers aligned with the original questions.

To assess the performance of our approach, we categorized the 28 test questions based on two key dimensions. The first dimension defines the complexity of each question in relation to the number of entities it references, the depth of relationships between them, and the extent of data processing required. This classification includes:

- **Single Entity, Single Property**: Questions that focus on retrieving a single attribute associated with one entity without requiring additional reasoning.
- **Single Entity, Multiple Properties**: Queries that involve multiple attributes of a single entity, requiring retrieval and potential comparison or aggregation.
- **Based on Reasoning**: More complex queries that require relationships between multiple entities to be analyzed, as well as inference or reasoning over retrieved data.

The second dimension distinguishes between questions based on how the requested information is stored and whether additional inference is needed:

- **Direct Questions**: Queries that rely on explicitly modeled entities and attributes in the KG, where retrieving the answer requires only structured queries.
- **Indirect Questions**: Queries requiring additional inference, either because the target information is not directly represented in the graph or because multiple relationships must be combined to extract relevant insights.

To evaluate system accuracy, one BIM expert analyzed the responses and classified them into three groups: correct, if the provided answer fully addressed the question; partially correct, if the response contained minor errors or missing details; and incorrect, if the system failed to generate a valid or meaningful answer. The final assessment determined that 16 responses were classified as correct, 3 as partially correct, and 9 as incorrect. The list of questions and their split and evaluation results is publicly available[14].

To qualitatively assess the tool's performance, we present representative examples for each outcome label. For the **Correct** category, the tool successfully answered the direct question *"How many doors are there on the second floor of the building?"*, leveraging the explicitly represented `beo:Door` entity and floor-level property. In the case of a **Partially Correct** answer, the system addressed the indirect query *"What is the average ceiling height of all hallways in the building?"*, but failed to accurately isolate hallway-specific elements, resulting in a generalized approximation. For the **Incorrect** outcome, the indirect question *"How many rooms are located on the second floor?"* could not be answered due to the absence of a direct room representation in the KG, highlighting current limitations in reasoning over inferred spatial constructs.

Performance varied across the different categories of questions. In the single entity with a single property category, our approach demonstrated strong performance, correctly answering five out of seven questions. One response was deemed partially correct due to an omission in the SPARQL query,

[14]https://github.com/aibba19/ASK-BIM/tree/main/Results

which resulted in retrieving an incomplete dataset. Another response was incorrect because the system attempted to infer room counts, a concept not explicitly stored in the KG, leading to query failure.

For the single entity with multiple properties category, our method performed well, with six of seven questions answered correctly. The single partially correct case involved retrieving the glazed area of windows, where the system retrieved the total surface area instead due to the absence of a dedicated property for glazing in the KG. This limitation highlights the challenge of interpreting missing or implied attributes when explicit representations are unavailable.

The reasoning-based category, which posed the most significant challenges, required inference across multiple entities and relationships. Out of fourteen queries, five were successfully answered, one was partially correct, and eight failed. The correct responses illustrate the ability of our approach to manage multi-step reasoning when all required attributes are explicitly represented. For example, when asked about the largest window that could fit within the smallest wall, the system accurately identified relevant entities, generated sub-queries, retrieved dimensions, and computed the correct response.

Many incorrect responses resulted from errors in SPARQL query construction. In cases such as calculating the enclosed area between walls, the system retrieved individual wall areas instead of reasoning over their spatial relationships. Similarly, when asked to identify corridors with doors exceeding a specified width, the failure arose because corridors were not explicitly represented as entities in the KG, requiring reasoning capabilities that were beyond the system's current capacities.

Despite these limitations, our approach successfully handled some indirect questions, demonstrating its potential to extract meaningful insights beyond explicitly modeled attributes. One notable success involved identifying fire-resistant walls, where the system inferred fire resistance based on textual labels in the KG. By recognizing standard notation such as "(1-hr)"[15] in wall descriptions, it was able to deduce the fire rating, showcasing its ability to process structured text elements for reasoning.

The evaluation underscores the strengths of our method in answering direct queries and performing structured reasoning when the required attributes are explicitly defined in the KG. However, the results also highlight challenges in handling complex inference tasks that require reasoning over implicit relationships or missing properties. The most frequent failure points were associated with SPARQL query generation, where queries were either too broad, leading to irrelevant results, or too restrictive, returning no useful data. Another common point of failure is the loss of information when converting the IFC model to the KG.

The system shows promise in bridging the gap between natural language queries and structured BIM data, but further refinement is needed to improve query generation and reasoning over inferred relationships.

6. Conclusions and Future Work

In this study, we explored how artificial intelligence and NLP could enhance access to BIM data, allowing users to query complex models through intuitive interactions. We developed an approach that converts BIM models into a structured KG, enabling semantic querying through a combination of SPARQL queries and LLM-based reasoning.

The system follows a structured process: first, converting an IFC model into a KG; second, analyzing user queries to identify relevant entities, relationships, and properties; and third, executing SPARQL queries to extract data, which is then processed by an LLM to generate a final response. We carried out the evaluation on a BIM model of a multi-storey office building in Barcelona, testing its performance against a range of predefined questions classified by complexity and information representation.

The evaluation showed that our approach performs well in retrieving information for direct queries, where relevant data is explicitly represented in the KG. However, indirect queries, especially those requiring inference or spatial reasoning, proved more challenging. A key limitation of the system lies in SPARQL query generation, where incorrect or incomplete queries sometimes led to inaccurate results.

[15]The "1-hr" notation indicates that the wall has a fire resistance rating of one hour, meaning it can withstand fire exposure for 60 minutes before losing its structural integrity.

Additionally, due to prompt size constraints, the system lacks full contextual awareness of the entire ontology, limiting its ability to infer implicit relationships between BIM entities.

Despite these challenges, the results demonstrate that integrating KGs with LLMs provides a viable method for natural language querying of BIM data. Future research will focus on improving SPARQL query accuracy, enhancing LLM-based reasoning for complex queries, and leveraging AI-driven techniques to interpret spatial relationships more effectively. Further exploration of ontology integration and optimized prompt engineering will also be key to improving the reliability and contextual accuracy of the responses of the presented approach.

7. Acknowledgments

This research has been partially funded by the European Union's Horizon research and innovation programme, as part of CHRONICLE project - Building Performance Digitalisation and Dynamic Logbooks for Future Value-Driven Services (Grant agreement ID 101069722). This work has been partially funded by the European Union's Horizon project LEGOFIT - Adaptable technological solutions based on early design actions for the construction and renovation of Energy Positive Homes (Grant agreement ID: 101104058).

Declaration on Generative AI

During the preparation of this work, the author(s) used ChatGPT, Grammarly in order to: Grammar and spelling check, Paraphrase and reword. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] R. Charef, H. Alaka, S. Emmitt, Beyond the third dimension of bim: A systematic review of literature and assessment of professional views, Journal of Building Engineering 19 (2018) 242–257. URL: https://www.sciencedirect.com/science/article/pii/S2352710217306320. doi:https://doi.org/10.1016/j.jobe.2018.04.028.

[2] R. Kebede, A. Moscati, H. Tan, P. Johansson, Integration of manufacturers' product data in bim platforms using semantic web technologies, Automation in Construction 144 (2022) 104630. URL: https://www.sciencedirect.com/science/article/pii/S0926580522005003. doi:https://doi.org/10.1016/j.autcon.2022.104630.

[3] R. Alonso, R. Olivadese, A. Ibba, D. R. Recupero, Towards the definition of a european digital building logbook: A survey, Heliyon (2023).

[4] T. Bloch, Connecting research on semantic enrichment of bim - review of approaches, methods and possible applications, Journal of Information Technology in Construction 27 (2022) 416 – 440. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85133341225&doi=10.36680%2fj.itcon.2022.020&partnerID=40&md5=bdfc6f37b9e7773519e48cc003041c64. doi:10.36680/j.itcon.2022.020.

[5] Y. Ding, J. Ma, X. Luo, Applications of natural language processing in construction, Automation in Construction 136 (2022) 104169. URL: https://www.sciencedirect.com/science/article/pii/S0926580522000425. doi:https://doi.org/10.1016/j.autcon.2022.104169.

[6] N. Wang, R. Issa, C. Anumba, Nlp-based query-answering system for information extraction from building information models, Journal of Computing in Civil Engineering 36 (2022) 04022004. doi:10.1061/(ASCE)CP.1943-5487.0001019.

[7] A. Nabavi, I. J. Ramaji, N. Sadeghi, A. Anderson, Leveraging natural language processing for automated information inquiry from building information models, Journal of Information Technology in Construction 28 (2023) 266–285. doi:10.36680/j.itcon.2023.013.

[8] J.-R. Lin, Z.-Z. Hu, J.-P. Zhang, F.-Q. Yu, A natural-language-based approach to intelligent data retrieval and representation for cloud bim, Computer-Aided Civil and Infrastructure Engineering 31 (2016) 18–33. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/mice.12151. doi:https://doi.org/10.1111/mice.12151. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/mice.12151.

[9] M. Yin, L. Tang, C. Webster, S. Xu, X. Li, H. Ying, An ontology-aided, natural language-based approach for multi-constraint bim model querying, Journal of Building Engineering 76 (2023) 107066. URL: https://www.sciencedirect.com/science/article/pii/S2352710223012457. doi:https://doi.org/10.1016/j.jobe.2023.107066.

[10] N. Wang, R. R. A. Issa, C. J. Anumba, Query Answering System for Building Information Modeling Using BERT NN Algorithm and NLG, ????, pp. 425–432. URL: https://ascelibrary.org/doi/abs/10.1061/9780784483893.053. doi:10.1061/9780784483893.053. arXiv:https://ascelibrary.org/doi/pdf/10.1061/9780784483893.053.

[11] S. Du, L. Hou, G. Zhang, Y. Tan, P. Mao, Bim and ifc data readiness for ai integration in the construction industry: A review approach, Buildings 14 (2024). URL: https://www.mdpi.com/2075-5309/14/10/3305. doi:10.3390/buildings14103305.

[12] S. Jiang, L. Jiang, Y. Han, Z. Wu, N. Wang, Openbim: An enabling solution for information interoperability, Applied Sciences 9 (2019) 5358.

[13] buildingSMART, Ifc schema specifications, https://technical.buildingsmart.org/standards/ifc/ifc-schema-specifications/, 2024. Accessed: February 2025.

[14] M. Bonduel, J. Oraskari, P. Pauwels, M. Vergauwen, R. Klein, The ifc to linked building data converter-current status, in: The IFC to Linked Building Data Converter-Current Status, 2018.