Building Information Model Analysis Through Large Language Models and Knowledge Graphs

Angelo Massafra[1], Ugo Maria Coraglia[2], Giorgia Predari[3], Riccardo Gulli[4]
[1,2,3,4] Alma Mater Studiorum University of Bologna
[1,2,3,4]{angelo.massafra2|ugomaria.coraglia|giorgia.predari|riccardo.gulli}@unibo.it

The advent of Large Language Models (LLMs) seems to mark a break between past and present in the methods of structuring knowledge, making it possible today to transfer this capability to machines even in a sector like AECO, always been information-intensive but resistant to technological transition. In terms of knowledge, the most established paradigm has been Building Information Modelling (BIM), with IFC functioning as the main schema for standardizing the industry's information. Added to this are knowledge graphs that, emerging with semantic web technologies, allow storing knowledge in structures consisting of nodes and edges with semantic meanings. Nevertheless, a barrier to the widespread adoption of BIM is its accessibility. Querying BIM models is often limited for stakeholders without digital skills, who may struggle to access the vast amount of information stored in these complex informative models. In an attempt to outline one of the possible uses of LLMs in BIM, this research proposes a method for querying BIM models through textual prompts aimed at analyzing a selected case study. In the workflow, a BIM model is first realized. Then, data is integrated into a knowledge graph. Next, ChatGPT's LLMs are used to activate query functions for the analysis of the graph. The results of the queries are displayed in a user-friendly graphical user interface. The study's outcomes offer insights for researchers and industry professionals, highlighting emerging research potentials for LLMs in the field.

Keywords: *Building Information Modeling, Large Language Models, Natural Language Processing, Knowledge Graphs.*

INTRODUCTION

Since the introduction of ChatGPT by openAI, Large Language Models (LLMs) have started to play a significant role in the advancement of Natural Language Processing (NLP) technologies. These cutting-edge systems signify a pivotal shift in the way we understand and structure knowledge, enabling the transfer of complex cognitive processes to machines. The availability of low-code application programming interfaces (APIs) for these LLMs has democratized the development of NLP-based applications, making it feasible even in information-intensive but technologically conservative sectors such as Architecture, Engineering, Construction, and Operations (AECO).

In recent decades, the cornerstone of information management in the AECO sector has been Building Information Modelling (BIM) (Eastman et al., 2018), whose standardization has been supported by the Industry Foundation Classes (IFC) schema. Complementing BIM are knowledge graphs (KGs), a byproduct of semantic web technologies, recently spread because of their capability to organize knowledge in flexible interconnected networks of nodes and relationships enriched with descriptive attributes and semantic contexts (Lygerakis et al., 2022). Nonetheless, the adoption of BIM and KG still faces challenges, particularly in terms of accessibility. For AECO's stakeholders lacking digital skills, navigating BIM models to extract valuable information can be complex.

Within this context, this research aims to explore the capabilities of LLMs in querying and analyzing BIM models. In particular, the study aims to: update the state of the art of the applications of LLMs in the construction industry, which are still limited; evaluate some GPT models for deployment of BIM-based in the industry; discuss opportunities and emerging challenges; and validate a use case of GPT with BIM and KG for data selection. To achieve these objectives, this research proposes a novel approach for extracting data from BIM by taking as only input textual prompts. The methodology, demonstrated within a prototype web application developed in Python, employs the Topologic software (Jabi et al., 2018) to construct the BIM model of a selected case study, which is then integrated into a KG using the NetworkX library. Subsequently, the ChatGPT APIs are invoked to transform the textual prompts given by the app's users into effective queries for extracting and aggregating data from the KG and visualizing it in user-friendly dashboards.

By evaluating some performance metrics and the overall efficacy of the approach in the case study, the study offers insights for academics and industry practitioners exploring the possibilities for applying LLMs within the AECO domain.

STATE OF THE ART

The rising need to digitalize the construction industry in recent years has propelled research towards systematizing various methods for the organization of building knowledge within digital models and applications. This journey, beginning with BIM, subsequently evolved into newer paradigms like digital twins (DTs) (Deng et al., 2021).

Building Information Modeling

BIM not only represents the building object geometrically but also includes physical attributes, functional specifications, and related semantic information, becoming itself a data-sharing environment. This sharing enables better collaboration between the professionals involved in architectural processes, improving the management of the entire building lifecycle (Fonseca Arenas and Shafique, 2023).

Going beyond detailed modelling in 3D, through BIM, it is possible to simulate and analyze times and costs or simulate thermal and energy aspects useful for managing, adapting, or improving building performance. For these reasons, BIM has several applications in the AECO sector, as it can cover different critical aspects of design and construction.

Knowledge graphs

Many methods and techniques have been developed to manage the flow of information within digital information systems. A particular focus has been given to building data integration within KGs, with many studies demonstrating their suitability in managing technologies like BIM, building management systems, and the Internet of Things (IoT), but also Business Intelligence (BI), leading to the rise of the DT concept (Boje et al., 2020).

These graph structures allow for identifying specific objects and labelling their relationships to explain and/or share a problem, a domain, or a concept, with the essential purpose of transferring knowledge. In particular, property graphs are composed of nodes (or vertices), used to represent entities and subjects, and edges (or links), used for their connections, i.e., predicates. Both semantics and property descriptions can be attributed to the nodes and edges that make up graphs. This kind of data organization allows, on the one hand, to assign an ontological meaning to each element that composes a knowledge domain and, on the other hand, to use attributes to describe every element. Once encapsulated within a KG, information can be conveyed as valuable knowledge via processes such as extraction, aggregation, and analysis, regardless of the complexity of the underlying cognitive structures. Beyond these capabilities, KGs are highly efficient in managing and integrating data from different environments or formats and analyzing Big Data. Coupled with the semantic web, they enable excellent ease of data extraction and accessibility on the web. All these features have allowed KGs to play a crucial role in digital applications, including in the AECO sector (Lygerakis et al., 2022).

A fundamental aspect within KGs is the use of ontologies to grasp the definition and term meanings and, therefore, enable effective data exchange and communication. In the development of LLMs, ontologies acquire particular value since they can provide valuable support to enrich textual inputs with semantic attribution.

Large Language Models

Accessing the knowledge encapsulated in BIM/graph systems largely remains in the domain of digital experts/developers responsible for their creation. However, with the advent of LLM-based tools, the prospect of making this information accessible to non-expert users (like building managers, design coordinators or construction contractors) is emerging.

LLMs are characterized by their extensive number of parameters and are trained using self-supervised and semi-supervised learning methods on large datasets. This shift has significantly propelled the field of NLP forward, emphasizing the utility of unlabeled data over the traditional reliance on labelled data for specific tasks.

In particular, OpenAI's Generative Pre-trained Transformer (GPT) has attracted significant attention. In the transition from GPT-2 to GPT-4, unveiled in 2023, these models have shown significant performance improvements. A key strength of GPT models lies in their ability to generate coherent and fluent language, closely mimicking human-written text, thus proving to be highly effective in several applications.

Despite their success in overcoming various sectors and their potential to enhance productivity, research on the application of GPT models in the AECO sector remains scarce. Saka has initiated a review in this area (Saka et al., 2024), yet much ground is needed to cover.

In this context, the article proposes a practical approach to a) identify opportunities for applying GPT models in the AECO sector, b) evaluate their integration with BIM models and KGs, c) assess some limitations of applying GPT models in this context, and d) validate a use case for GPT models within the AECO industry.

METHODOLOGY

The research workflow, shown in Figure 1, is divided into three main phases: data modelling, data extraction, and data visualization.

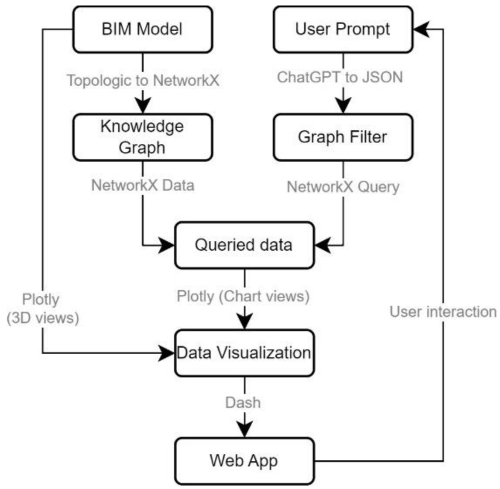


Figure 1
Conceptual diagram illustrating the research workflow.

In the first phase, the BIM representation of the building in question is developed. This model becomes the basis for the development of the KG, within which all the information and components represented in the BIM framework are stored. In the second phase, a specialized algorithm, supported by ChatGPT APIs, interprets the textual requests to generate filters useful for querying the KG. These filters are designed with specific syntax and semantics, allowing effective application to the KG in the data retrieval process. The extracted data is then processed and visualized in the third phase via a web application that provides a 3D representation of the information in graphs and tables. The application also provides simplified explanations of the data, making it accessible even to users with limited digital experience.

The entire process is realized using BTwin, an open-source Python library currently being developed by the authors, which structures building knowledge in graphical formats. Based on an established set of Python libraries such as Topologicpy, Pandas, Dash, Plotly, ChatGPT, and NetworkX, BTwin offers a low-code solution for prototyping digital decision support systems for building management.

The Faculty of Engineering building, a historically significant building managed by the University of Bologna (Italy), is selected as the subject of the workflow. This building, selected as a case study, becomes a representative model of the wide range of buildings managed by local public administrations in Italy. In particular, the study focuses on a particular building zone comprising approximately 65 rooms.

Data modeling

BIM model realization. The BIM model of the building was created using Autodesk Revit and Topologicpy, following a semi-automated method previously developed in another research. This model mainly consists of Topologic "Cells", which represent the building's spaces, Topologic "Faces" that define the elements bounding the spaces (including walls, roofs, and floors), and Topologic "Apertures" (representing doors, windows, and holes), embedded in the Faces to allow the passage of people, light, or air between different areas (Figure 2).

The advantage of using Topologicpy for setting up the workflow lies in its lightweight and straightforward data structure, which is based on the JSON format. This tool enables the easy analysis of both topological characteristics and informational attributes of BIM models, highly simplifying the process of examining relationships and data embedded within BIM formats. Moreover, Topologicpy relies on solid graph analysis functions, that make it an interoperable tool with many graph analysis software, like NetworkX and Neo4J.

For the purposes of the presented application, the following example properties of the BIM elements are considered. The spaces are characterized by geometric information (gross area and gross volume) and occupancy data (occupancy type and people count at peak hours). The faces are detailed with geometric data (gross area) and information about their orientation (i.e., West, East, South, North). Lastly, the apertures are distinguished by their dimensional attributes (width, height, and area) and material properties (frame material, U-value, and glazing type). However, the research approach can be extended equally to other properties.

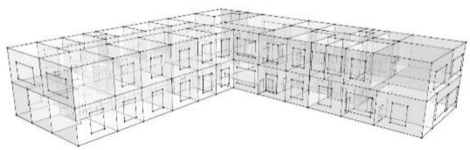


Figure 2
The Topologic BIM model of the selected case study.

Knowledge graph development. The transformation of the BIM model into a knowledge graph is made by BTwin, which converts the Topologic BIM representation into a NetworkX graph, leveraging on the semantic principles defined in a supporting ontology. This ontology federates meanings and concepts from established ontologies in the field, i.e. IFC, BOT (Building Topology Ontology) (W3C, 2021), and Brick (Balaji et al., 2018), aiming to simplify, albeit within a limited scope, the representation of spatial, construction, and HVAC systems aspects of the case study building.

Utilizing a federated ontology, BTwin enables the encapsulation of building knowledge within knowledge graphs, underpinning the notion that a building can be conceptualized as a graph comprising nodes (representing entities or elements) and edges (depicting relationships). These nodes are semantically categorized based on their alignment with specific classes and subclasses within the ontology, while the edges denote the relationships between them, whose types are also codified by the ontology.

For the purpose of this paper, entities are classified into:

- Zones: These are spatial components, as defined in BOT, existing either physically or abstractly, defined by their location within a 3D space.
- Interfaces: abstract or physical partitioning elements, as defined in BOT, that bind and connect zones, including walls, roofs, and floors, as well as doors and windows.

Zones encompass classes like 'Site', 'Building', 'Storey', 'Zone', and 'Space', forming BTwin's spatial hierarchy, which adheres to BOT's and IFC's spatial structure. Interfaces, whose concept is also established in BOT, can be divided into 'Face' and 'Aperture' (as in Topologic), with Faces including elements like walls and roofs, and Apertures referring to openings such as doors and windows.

Based on their class and subclass affiliations, nodes are interconnected through various relationship types. These are:

- 'HasLocation/IsLocationOf' for indicating spatial containment relationships between spatial elements (e.g. a storey is the location of a space).
- 'IsInterface/HasInterface' for indicating the adjacency relationship between spaces and interfaces (e.g. a wall is the interface of a space).

The 'HasLocation' relationship is defined in Brick's ontology, while the "IsInterface" is in BOT's.

Beyond the semantic and categorization of nodes, BTwin facilitates the enrichment of modeled entities with data and metadata that offer qualitative and quantitative insights into the entities. This enrichment involves associating nodes with property sets (PSets), collections of related properties organized by theme and type, similar to IfcPropertyStes in the IFC's schema. For instance, PSets can detail the thermal setpoints of a space, delineating its heating and cooling requirements, or the thermal characteristic of a wall, illustrating its U-Value and thermal inertia. These PSets apply to any entity type within the graph, linked via the 'HasPropertySets' relationship, signifying the association between properties and their respective entities.

For constructing the KG of the presented case study, Topologic elements (spaces, faces, and apertures) were transformed into NetworkX nodes, along with their properties, while their interrelations were modeled as edges, resulting in the KG illustrated in Figure 3.

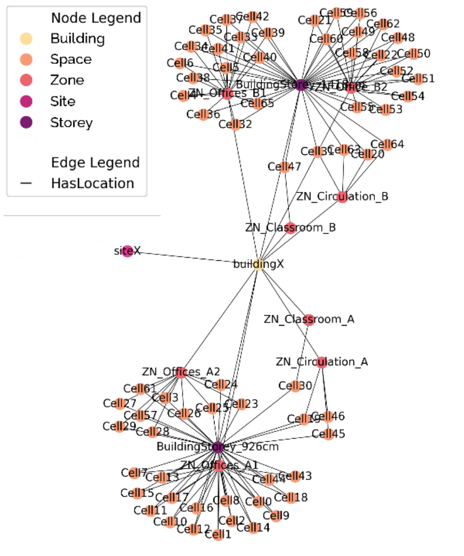


Figure 3
Knowledge graph representing the case study's spatial hierarchy

Data extraction

The data extraction process unfolds in three distinct steps. Initially, the app's user inputs a textual prompt for querying data from the KG, e.g. «Sum the area of all the spaces with office function.» This input is then preprocessed and augmented with additional contextual information, all in textual format, that details the request before sending it to ChatGPT, adding relevant words that might improve the accuracy of the results. Following this enrichment, the improved prompt is given as input to ChatGPT. A GPT model interprets the request and generates a JSON object, representing a filter for querying the NetworkX graph. This object is encoded with a specific syntax tailored to navigate the graph's structure, targeting nodes and edges relevant to the query based on the semantic and hierarchical information encoded within the knowledge graph.

User prompt. In the initial data extraction step, the user prompts the app to execute a querying task. In this study, permissible tasks include aggregations such as count, sum, minimum, maximum, and average of properties related to spaces, faces, or apertures, as well as basic query and filtering operations based on properties using conditional rules.

Example prompts working in the app include «*Display all spaces used as offices*», «*Sum the area of walls facing West*», or «*Calculate the average U-Value of windows with wooden frame*». These requests are relayed to ChatGPT via Python APIs, with the prompt enriched with contextual instructions to guide the LLM towards more accurate responses.

Prompt contextualization. Initially, textual indications about the context where the task is performed are provided to GPT: «*The following command is given by the user of a BIM-NLP application to query a knowledge graph.*»

Then, the prompt is concatenated to such indications and basic information about the ontology is added to the textual request: «W*hich class does the prompt refer to among these: 'building', 'storey', 'space', 'wall', 'roof', 'floor', 'window', 'door'? Return the class name*».

Following this, instructions on the properties the application should examine are given: «*Which properties do the user want to inspect among the following {props}? Return only the prop names you think are mentioned in the user's prompt*." Only properties relevant to the class identified by GPT are passed to this text within the '{props}' variable. For instance, if the class found is 'space', property names such as gross area, gross volume, occupancy type, and occupancy number peak are passed to the request.

ChatGPT is then asked to identify, within the user prompt, any conditional rules for filtering elements: «*What are the property filter values? These are the eligible operators: equalTo, notEqualTo, minorOf, majorOf*».

Additionally, for applying conditional rules on string properties (e.g., occupancy type), guidance is also provided to GPT by illustrating what are the property values present within the BIM to enable better string matching queries: «*Please, among these property values {propValues}, find the string that may match with the filter value. Return them as a list in this format: ['propValue1','propValue2',etc.]*». This includes detailing the specific strings or terms used within the model to describe various attributes, ensuring that the query mechanism can accurately identify and filter entities based on textual descriptions. For example, if the model uses specific terminology to designate room functions, these terms are communicated to the query system to facilitate precise matching and retrieval of data based on the user's textual prompts. This approach enhances the tool's ability to interpret and execute more complex queries, leveraging the rich semantic information encoded within the BIM model's property values.

ChatGPT is then informed about the tasks the tool can perform: «*These are the task types that the application can perform: 'Sum', 'Min', 'Max', 'Mean', 'Query'*».

A final crucial request concerns the JSON filter format to be provided as output: «Answer in the following format: *{'Task': {'TaskType' : 'myTaskType', 'PropName' : propNameToSumMinMaxMeanQuery ', 'Class' : myClass, 'Filter1' : {'PropName' : myPropName1, 'FilterValue' : myFilterValue1, 'FilterOperator' : myFilter*

Operator1} , 'Filter2' : {'PropName' : myPropName2 , 'FilterValue' : myFilterValue2 , 'FilterOperator' : myFilterOperator2} , 'Filtern...'.»

The complete request request is finally passed to ChatGPT as a concatenation of the mentioned texts.

Graph filter. ChatGPT's output is a JSON object representing a graph filter using the required syntax. Let's imagine the user's prompt is «*Sum the area of all the spaces with office function and more than 25 sqm area*». The output is the JSON object provided in Figure 4.

Figure 4
Example of graph filter JSON object


```
{
    "Task": {
        "TaskType": "Sum",
        "PropName": "pr_GrossArea"
    },
    "Class": "Space",
    "Filter1": {
        "PropName": "pr_OccupancyType",
        "FilterValue": [
            "Office:AdministrationOffice",
            "Office:MeetingRoom",
            "Office:ResearchOffice"
        ],
        "FilterOperator": "="
    },
    "Filter2": {
        "PropName": "pr_GrossArea",
        "FilterValue": 25,
        "FilterOperator": ">"
    }
}
```


Within this object, the key 'Task' indicates the operation to be performed on the filtered data. In the example, the task type 'Sum' suggests that the operation involves summing up the values of the 'pr_GrossArea' property of the elements to filter.

The key' Class 'specifies instead the node class the filter applies to, which in this case is 'Space '. This implies that the data to be queried only concerns space elements.

At the 'Filter1 'key, a filter object narrows down the data based on the 'pr_OccupancyType' property. The filter value here is a list containing 'Office:AdministrationOffice', 'Office:MeetingRoom', and 'Office:ResearchOffice' property values retrieved from the graph. This indicates that the filter is looking for spaces designated as either administration offices, meeting rooms, or research offices, encompassing a broader range of office types compared to the generic 'office' category requested by the user prompt. The filter operator '=' suggests that the property value must match one of the items in the list.

Similarly, the 'Filter2' applies another filtering based on the 'pr_GrossArea' property. It selects spaces with a gross area greater than 25 sqm, using the '>' operator. This filter sets a minimum size threshold for the spaces of interest to focus on spaces of significant size for the analysis.

KG query. The graph filter is then applied to the KG to extract the required data. To achieve this, all nodes belonging to the class specified by the filter are selected first. Then, the properties attached to these nodes are read and compared with those in the filter. If all the conditional rules are satisfied, the node is extracted from the graph, and its properties are used for display in the application and to perform the operations/tasks requested by the user (e.g., sum). To make the query operations efficient, functions from NetworkX, a library specifically developed for these purposes, are utilized.

Data visualization

In the data visualization stage, a sample web application, designed to process and visualize data related to BIM through user inputs, is used to interactively display the results of the data extraction step. The app integrates various technologies and open-source Python frameworks, with Dash and Plotly at its core, facilitating the creation of an interactive web-based application.

The application is structured to allow users to interactively engage with the data through a simple, intuitive interface.

At the heart of the app is a text input field where users can specify their queries. Accompanying this input field is a 'Submit' button, which, once clicked, triggers the app's data processing mechanisms. The underlying layout hosts visual elements such as 3D view and chart placeholders, which are set to dynamically update visualization based on the user's input, providing a graphical representation of the query results. Additionally, there is a designated area for displaying results in textual form, offering users a clear and concise summary of the processed data (Figure 5).

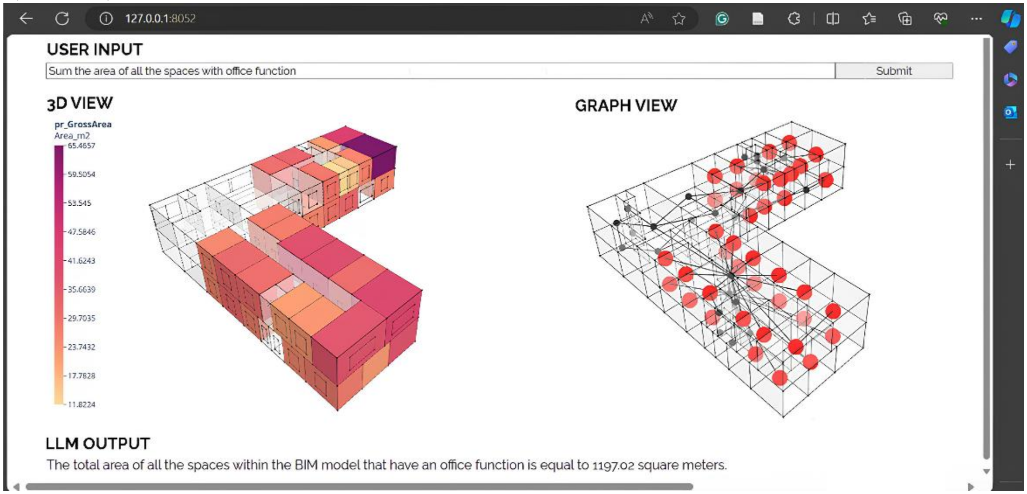


Figure 5
GUI of the developed web application

TESTS AND RESULTS

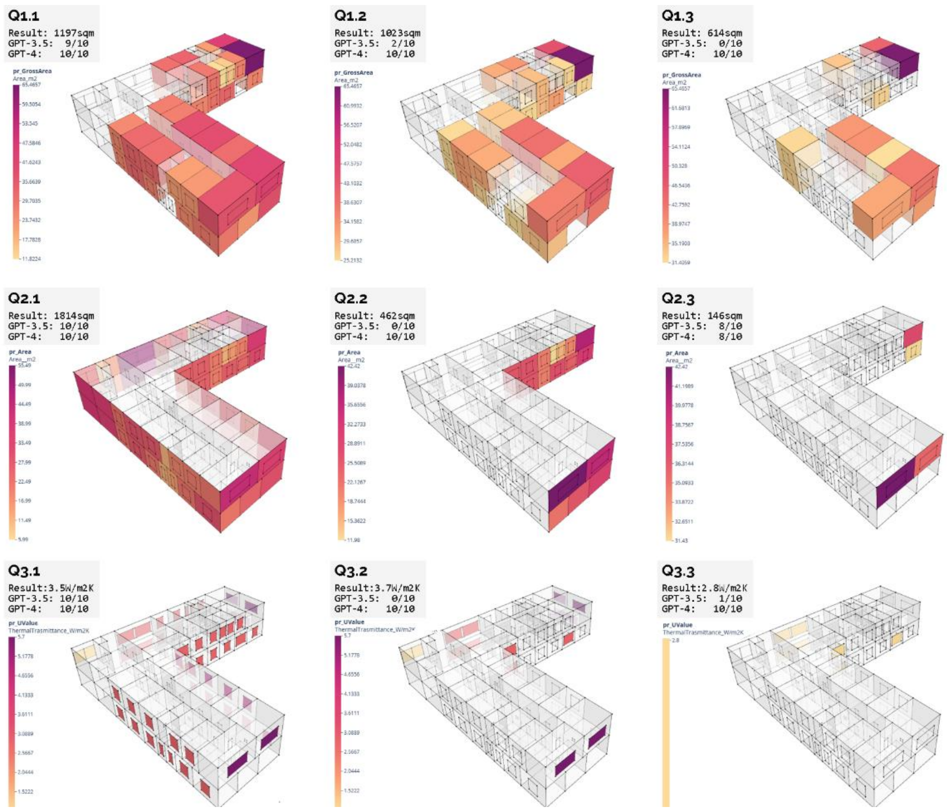
This section introduces a series of tests conducted using the developed application. These tests serve two primary purposes: firstly, to demonstrate practical examples of the app's usage, and secondly, to assess its performance based on specific metrics. A notable aspect of the evaluation includes comparing the performance of different LLMs, specifically GPT-4 and GPT-3.5-turbo. Additionally, the tests explore the responses to prompts from various users, which are designed to achieve progressively more challenging goals.

The tool was tested on nine questions: three relating to spaces, three to walls, and three to windows. For each of these categories, three progressively more complex questions were asked.

For spaces: «*Sum the area of all the spaces with office function*» (Q1.1), «*Sum the area of all the spaces with office function and more than 25 sqm area*» (Q1.2), and «*Sum the area of all the spaces with office function, more than 25 sqm area and hosting more than three people*» (Q1.3). For walls: «*Sum the area of all the external walls*» (Q2.1), «*Sum the area of all the external walls exposed to south*» (Q2.2), and «*Sum the area of all the external walls exposed to south and area more than 30 sqm*» (Q2.3). For windows: «*Mean the U-Value of all the windows*» (Q3.1), «*Mean the U-Value of all the windows wider than 200 cm*» (Q3.2), and «*Mean the U-Value of all the windows wider than 200 cm and with wooden frame*» (Q3.3). Note how the grammar of the questions is not deliberately perfect in order to simulate a user who inputs their questions quickly and without accurate grammatical control.

The results of these tests are provided in Figure 6. ChatGPT-4 demonstrated perfect accuracy across all tasks in evaluating office space areas (Q.1.1), while ChatGPT-3.5-turbo showed unreliable performance especially in more complex queries (Q1.2 and Q1.3). For the summation of external walls' areas, both models achieved full marks when considering all walls (Q2.1), but ChatGPT-3.5-turbo faltered in identifying walls with specific orientations or sizes (Q2.2 and Q2.3). With apertures, both models perfectly calculated the mean U-Value for all windows (Q3.1), but again, ChatGPT-3.5-turbo encountered errors with more detailed specifications (Q3.2 and Q3.3), unlike ChatGPT-4, which remained accurate. ChatGPT-4 proved to be a highly reliable model for these architectural computations, significantly outperforming ChatGPT-3.5-turbo.

Figure 6
Tests conducted through the web application



CONCLUSION

This article presents an example of how LLMs combined with BIM can be used to improve accessibility to data contained within such digital systems by non-expert users. The improvement of accessibility is promoted through three key elements: the use of simple BIM formats (Topologic and JSON-based KGs) to model building data, the use of only textual inputs resembling human language to extract the data, and the visualization of data in a user-friendly dashboard.

The goal of the article is to conduct an exploration into the integration between BIM, KG, and LLM, demonstrating it in a prototypical form within a web application. The novelty of the paper compared to the initial state-of-the-art on the interaction between BIM and GPT models, presented by Saka in 2024, lies in the fact that this research combines continuous forms of knowledge (LLMs) with discrete forms of knowledge (KG), leveraging the strengths of both.

Despite the simplified approach, the results found are noteworthy, thus similar applications in the AECO sector can be expected in the coming years. On one hand, LLM models have proven to be quite useful in performing data extraction tasks. On the other hand, the application has emphasized the importance of organizing information in the BIM field semantically, as without contextual information (such as on the KG ontology), LLM models fail to complete their tasks efficiently.

Future developments will involve extending the approach within a digital twin system. Particular attention will be paid to the development of LLM-based algorithms for extracting time-series data (real-time and historical) related to the energy performance and comfort of a real case study.

ACKNOWLEDGEMENT

This research was funded by the BeTwin project, under the AlmaValue 2023 program of the University of Bologna. The publication was realized by a researcher co-funded by the European Union - PON Research and Innovation 2014-2020. The authors thank Prof. Wassim Jabi who supervised the realization of the BIM model presented in the paper.

REFERENCES

Balaji, B., Bhattacharya, A., Fierro, G., Gao, J., Gluck, J., Hong, D., Johansen, A., Koh, J., Ploennigs, J., Agarwal, Y., Bergés, M., Culler, D., Gupta, R.K., Kjærgaard, M.B., Srivastava, M., Whitehouse, K., 2018. Brick : Metadata schema for portable smart building applications. Applied Energy 226, 1273–1292. https://doi.org/10.1016/j.apenergy.2018.02.091

Boje, C., Guerriero, A., Kubicki, S., Rezgui, Y., 2020. Towards a semantic Construction Digital Twin: Directions for future research. Automation in Construction 114, 103179. https://doi.org/10.1016/j.autcon.2020.103179

Deng, M., Menassa, C.C., Kamat, V.R., 2021. From BIM to digital twins: a systematic review of the evolution of intelligent building representations in the AEC-FM industry. ITcon 26, 58–83. https://doi.org/10.36680/j.itcon.2021.005

Eastman, C.M., Teicholz, P.M., Sacks, R., Lee, G., 2018. BIM handbook: a guide to building information modeling for owners, managers, designers, engineers and contractors, Third edition. ed. Wiley, Hoboken, New Jersey.

Fonseca Arenas, N., Shafique, M., 2023. Recent progress on BIM-based sustainable buildings: State of the art review. Developments in the Built Environment 15, 100176. https://doi.org/10.1016/j.dibe.2023.100176

Jabi, W., Aish, R., Lannon, S., Chatzivasileiadi, A., Wardhana, N.M., 2018. Topologic - A toolkit for spatial and topological modelling. Presented at the eCAADe 2018: Computing for a better tomorrow, Łódź, Poland, pp. 449–458. https://doi.org/10.52842/conf.ecaade.2018.2.449

Lygerakis, F., Kampelis, N., Kolokotsa, D., 2022. Knowledge Graphs' Ontologies and Applications for Energy Efficiency in Buildings: A Review. Energies 15, 7520. https://doi.org/10.3390/en15207520

Saka, A., Taiwo, R., Saka, N., Salami, B.A., Ajayi, S., Akande, K., Kazemi, H., 2024. GPT models in construction industry: Opportunities, limitations, and a use case validation. Developments in the Built Environment 17, 100300. https://doi.org/10.1016/j.dibe.2023.100300

W3C, 2021. Building Topology Ontology [Online]. Available at: https://w3c-lbd-cg.github.io/bot/