



Transforming BIM data interaction: A user-centric framework leveraging lightweight ontology and large language model integration

Zhikun Ding^{1,2,3}, Yongchang Li², Zhiwei Liu², Hongping Yuan^{4,*}

¹Key Laboratory for Resilient Infrastructures of Coastal Cities, Ministry of Education, Shenzhen University, Shenzhen 518060, Guangdong, China.

²Sino-Australia Joint Research Center in BIM and Smart Construction, Shenzhen University, Shenzhen 518060, Guangdong, China.

³Shenzhen Key Laboratory of Green, Efficient and Intelligent Construction of Underground Metro Station, Shenzhen University, Shenzhen 518060, Guangdong, China.

⁴School of Management, Guangzhou University, Guangzhou 510006, Guangdong, China.

***Correspondence to:** Hongping Yuan, School of Management, Guangzhou University, Guangzhou 510006, Guangdong, China. E-mail: hpyuan@gzhu.edu.cn

Received: November 05, 2025 **Accepted:** March 09, 2026 **Published:** March 12, 2026

Cite this article: Ding Z, Li Y, Liu Z, Yuan H. Transforming BIM data interaction: A user-centric framework leveraging lightweight ontology and large language model integration. J Build Des Environ. 2026;4:202594. <https://doi.org/10.70401/jbde.2026.0034>

Abstract

This study addresses the interoperability of building information modeling (BIM) across different systems and platforms. Because the semantics of the industry foundation classes (IFC) standard are large and complex, traditional full semantic conversion methods are general-purpose, but they often lead to data expansion, which reduces system responsiveness and usability. In addition, the strong dependence of BIM models on specialized software further limits flexibility in practical use. Our novel framework combines a streamlined ontology with IFC, significantly improving memory efficiency and operational effectiveness compared to existing systems. Additionally, we integrate large language models for enhanced natural language processing in BIM data interactions. This harmonization of technologies not only simplifies system extension but also makes BIM data services more user-friendly and adaptable to various industry needs. By streamlining BIM data management and enriching data services, our approach broadens BIM's applicability and improves data integration and extraction, establishing a more interactive and user-centric paradigm.

Keywords: Building information modeling, ontology interaction, large language models, intelligence semantic interaction, data management

1. Introduction

Building information modeling (BIM) plays a crucial role in the digital transformation of the construction industry, creating digital replicas of physical structures and compiling comprehensive lifecycle data of building components^[1,2]. However, integrating BIM data across various phases of engineering projects presents considerable challenges, mainly due to inconsistent data standards among different BIM software, leading to significant interoperability issues^[3]. Industry foundation classes (IFC), as a universal BIM data standard, alleviate some interoperability issues but have limitations. The complexity of IFC's structure and its extensive learning curve hinder its widespread adoption, while the text-heavy nature of IFC files reduces efficiency^[4].

Currently, academics have tried to combine IFC standards with database or ontology technologies to enhance BIM data management^[5]. The core of these studies involves converting the IFC Schema into more advanced data models, such as database schemas and Industry Foundation Classes Web Ontology Language (IFCOWL) schemas^[6]. In this conversion mode, users can efficiently query data using Structure Query Language (SQL) or SPARQL Protocol and RDF Query Language (SPARQL) commands and simplify data querying operations through the design of customized views. These methods are difficult to achieve with the traditional IFC schema. Current research in this area has two main shortcomings. First, the complex structure of the IFC schema leads to system redundancy^[6-7]. The second key issue is human-computer data interaction (HDI), with existing methods mainly relying on static HDI (SHDI) with fixed datasets. However, this method's drawback is that it requires direct user intervention whenever data requirements change. Additionally, this method is not only inefficient but also demands high levels of user expertise and operational skills^[8]. Therefore, a



lightweight system and a Dynamic Human-Data Interaction (DHDl) approach are essential.

Based on the analysis above, this study aims to address issues in two primary areas. Firstly, we focus on mitigating data redundancy in IFCOWL. To address redundancy, a simplified IFC data management method is proposed, centered on developing a concise ontology dedicated to representing building topology, intentionally excluding the conversion of detailed entities such as geometry and materials in the IFC schema. This ontology captures building component instances and their spatial relationships. Subsequently, users can retrieve entity instances using SPARQL queries. This allows for precise retrieval of components at the spatial level. Finally, using the GUID, it is possible to extract geometric, material, and other detailed attribute values of each building element instance from the text-based IFC dataset^[9]. This approach enhances the overall efficiency and streamlining of the system.

To address the second issue, we focus on constructing a DHDl framework. Unlike SHDI, DHDl enables computers to directly participate in the workflow. In this model, computers act as intelligent agents, replacing the user's role^[10]. The key challenge in implementing DHDl lies in accurately parsing the semantic information of natural language and converting it into precise structured data instructions. Traditional keyword recognition and instance matching technologies only work for sentences with clear structures and struggle to handle complex or implied semantic expressions^[11]. The rapid development of large language models (LLMs) has proven effective in addressing this issue, thereby opening up new possibilities for intelligent agents and DHDl^[12]. The application of LLMs in professional domains simplifies the processing of traditional tasks, allowing originally complex tasks to be completed efficiently through natural language dialogue, significantly enhancing the naturalness, intuitiveness, and efficiency of human-computer interaction^[13].

Therefore, this study introduces LLMs as the core intelligent agents. LLMs receive and analyze data requirements expressed by users in natural language and, through semantic analysis, convert these into SPARQL commands usable by subsequent modules. Throughout the system's operational cycle, LLMs manage data validation and transfer tasks between different submodules. This integration significantly reduces the need for users to deeply engage in the DHDl process, simplifies the workflow, and lowers the learning curve. In our proposed framework, we synergistically combine IFC-expressed BIM data, ontology technology, and LLM technology to provide users with enhanced BIM data services. This integration allows for intuitive, natural language interactions with BIM data, accommodating diverse industrial needs and use cases. It significantly improves the efficiency of data integration and extraction, effectively addressing the current shortcomings in BIM data management.

The introduction section primarily provides an overview of the research content and its significance. In the second section, we conduct a comprehensive literature review to summarize the common methods of BIM data processing and sequentially analyze their advantages and disadvantages. Based on related work, we outline the objectives that this study aims to achieve. In the third section, we design a specific technical framework aimed at achieving the objectives set out above and detail how each technology and method collaborates within this framework. In the fourth section, we test the effectiveness of the proposed methods through a case study. Finally, we analyze the effectiveness and limitations of our research, providing insights for future scholars to build upon this work.

2. Review of Related Literature

In this section, we systematically review the literature related to our study, focusing on two main aspects. The first aspect concerns BIM data processing methods related to the IFC. The second aspect involves HDI driven by natural language within the BIM domain.

2.1 Processing methods of BIM data

2.1.1 Direct processing methods

The proprietary format-based approach to BIM data processing relies on using API interfaces provided by BIM software vendors to parse unique formats, such as .rvt and .dgn. This method, which is relatively common for processing BIM data, involves creating data import/export tools that facilitate the movement and use of BIM data across various software platforms^[14-16]. This method, tailored for individual software formats, constitutes a 'point-to-point' BIM data processing strategy. Its main limitation lies in its specificity to certain software platforms.

The IFC functions as a 3D building product data standard based on the object-oriented EXPRESS data specification language, essential for articulating BIM. As an open data standard, IFC seeks to overcome data interoperability challenges among diverse BIM software platforms by offering an intermediary format. To address different data application requirements across various project phases, buildingSMART has developed the information delivery manual (IDM)^[17] and model view definition (MVD)^[18] based on IFC. IDM delineates data requirements for distinct business phases, and MVD enables the extraction of specific segments of BIM model data. This method requires precise definitions and standardized methodologies to establish IDM/MVD, targeting specialized domain-specific data needs. Based on standardized IDM/MVDs, tools are developed by domain experts to extract data from IFC datasets. These tools support specific domain requirements and enhance data exchange processes in areas such as engineering measurement, cost estimation^[19], energy consumption analysis^[20], and construction schedule generation^[21]. Beyond standardized development processes, some researchers have explored more streamlined operations and development workflows to support the

implementation of customized data extraction requirements. Won *et al.*^[22] introduced a method for extracting partial model data without requiring MVD definitions, thereby streamlining the operational process. This innovative approach depends solely on the relationships between data instances within the IFC instance model file to extract partial models. Further enhancing this concept, Mingjuan^[23] advanced the IFC partial model extraction algorithm, focusing on the identification and removal of invalid instances.

However, the intricate data structure of BIM and the required advanced programming skills pose significant challenges and inefficiencies for practitioners. The lack of uniformity in proprietary BIM data formats exacerbates interoperability issues across different BIM platforms. While the IFC offers a universal standard for BIM data storage and mitigate interoperability challenges to a degree, several critical issues remain, including semantic limitations, complex data structures, difficulties in data processing and analysis, and constrained extensibility^[24]. As the IFC schema is subject to alternative definitions and interpretations of domain knowledge, its flexible data representation can be an obstacle to defining strict rules and robust validation processes regarding object attributes and references^[25]. This flexibility aims to provide a comprehensive representation of various objects in the construction domain, despite the inherent incompleteness of IFC. The detailed data representation complicates processing and makes analysis challenging, requiring specialized knowledge and advanced techniques for resolution. Finally, the overly detailed data representation results in BIM datasets based on the IFC standard being excessively large. So, efficient algorithms and tools are often required to extract, transform, and analyze IFC data to meet the demands of various application scenarios.

These challenges underscore the need for ongoing development and enhancement of BIM data standards to enable more effective and comprehensive data integration across various BIM systems. Current ontology methods fail to effectively balance expressive completeness and system efficiency, highlighting the need for a selective transformation strategy that retains only high-level semantic structures to support rapid querying.

2.1.2 Indirect processing methods

Researchers implement the conversion from IFC schema to database schema, so that, BIM data can be stored and managed via databases. Compared with extending IFC schema, extending at the database schema level is clearly easier and more efficient. Similarly, the conversion between IFC schema and Ontology Web Language (OWL) schema is also due to this reason. The difference lies in that OWL offers has advantages in semantic extension and the expression, storage, and processing of unstructured data.

A notable example is BIMserver^[26], an open-source platform based on the IFC standard, providing centralized storage, querying, and sharing of BIM data. Building on BIMserver, Mazairac *et al.*^[27] developed BIMQL, a query language for BIM models, enhancing data management efficiency. Furthermore, researchers including Cheng *et al.*^[28], Solihin *et al.*^[29], and Guo *et al.*^[30] have explored converting IFC models to relational database models, using relational databases for IFC data management. Wyszomirski^[31] applied NoSQL databases for IFC data processing, aiming for BIM-GIS data integration, while Yang *et al.*^[32] and Gradišar *et al.*^[33] examined the combination of IFC data with other data sources using graph databases. These studies demonstrate the exploration of various database systems integrated with IFC to address different research challenges. However, it is important to note that constructing such systems is often complex, relying on specialized databases and resulting in a sophisticated ecosystem. Although these systems mainly focus on associating, storing, and querying structured data, they still face challenges in handling unstructured knowledge and data integration.

Semantic web technologies have been developed to provide a structured context for unstructured data. These technologies enable data to be understood not only by humans but also by machines, thereby supporting more intelligent information retrieval, data mining, and knowledge management. Their integration with BIM promises substantial benefits for the architecture, engineering, and construction industry^[34]. During the design phase, an ontology model of building codes can be developed using the Semantic Web, enabling the automation of various tasks. For instance, the system can automatically verify whether the fire-resistance rating of walls complies with relevant code requirements or generate building energy performance reports, thereby eliminating the need for manual, item-by-item checks^[35,36]. In practical engineering applications, semantic reasoning engines can perform complex queries similar to those in databases. For example, one can query all walls that are constructed with sustainable materials and meet specific seismic performance standards, a capability that is difficult to achieve with traditional BIM approaches^[37]. Furthermore, by leveraging semantic web technologies, such as the BIM-to-GEO standard, semantic mapping between GIS and BIM can be achieved, supporting the unified management of 3D city models^[38].

Ontologies, fundamental to the semantic web, offer a structured conceptual model for domain knowledge, enabling effective machine understanding and interpretation of data. OWL is designed for representing these ontologies, providing an array of tools and syntax to define and elaborate on classes, properties, relationships, and other complex semantic elements. In the realm of BIM, integrating BIM with the semantic web chiefly involves translating IFC schema into OWL schema. Pauwels' pioneering work in converting IFC into an ontology format led to the creation of the IFCOWL ontology, now recognized as a standard by buildingSMART^[39]. The IFCOWL offers a comprehensive transformation of IFC schema, creating a corresponding individual in the IFCOWL ontology for each IFC instance.

Notably, IFCOWL files tend to occupy considerably more memory than their IFC counterparts, potentially obstructing efficient

ontology-based reasoning and querying processes. As indicated in Table 1, examinations of five distinct models demonstrate that the memory usage of IFCOWL files is approximately 3–10 times greater than that of IFC files, with the variance contingent on the model’s complexity. This becomes particularly burdensome in large-scale projects and significantly limits the performance of ontology-based reasoning and querying. To address this issue, researchers have proposed semantics-oriented simplification strategies. For example, Wu *et al.* introduced a geometry simplification method based on semantic constraints: the level of geometric detail retained for each building component is dynamically determined according to its semantic attributes, such as type and function^[40]. This approach achieves selective geometric optimization while preserving, or even enhancing, the semantic completeness of the model, offering a practical way to balance semantic expressiveness with computational efficiency.

Table 1. Comparison of data volume between IFC and IFCOWL.

	Model1	Model2	Model3	Model4	Model5
IFC(.ifc)	12kb	3.09MB	25.8MB	49.1MB	110MB
IFCOWL(.ttl)	98kb	9.50MB	259MB	427MB	1.09GB

IFC: industry foundation classes; IFCOWL: Industry Foundation Classes Web Ontology Language.

Indirect BIM data processing methods have indeed somewhat reduced usage costs and improved efficiency, particularly in terms of multi-source data integration and semantic expansion, showing promise and potential value. However, these approaches are primarily still in the research phase and are rarely implemented in actual engineering projects.

2.2 HDI method driven by natural language

The continual advancement of natural language processing (NLP) technology has ushered in new possibilities for HDI. This is particularly evident in the realm of BIM data management systems. BIM models are typically characterized by large data volumes, highly complex structures, and rich semantic information. However, the efficiency of utilizing this information has long been limited by the complexity of specialized BIM software and the high technical barrier to data querying^[41]. In recent years, researchers have begun integrating natural language analysis modules into traditional BIM systems to enhance the intelligence of HDI and streamline the complexity of BIM data processing. For instance, Lin^[42] proposed a method for cloud BIM data retrieval and representation using natural language processing. They introduced “keywords” and “constraints” to express user requirements, mapping them to IFC entities or properties through the International Framework for Dictionaries (IFD). Then, they established relationships between user needs and IFC-based data models through pathfinding on IFC schema-generated graphs. Moreover, Guo^[43] developed a method for translating natural language queries into SPARQL queries, facilitating direct interaction between natural language and ontological data. It is essential to recognize that much of this research is reliant on keyword recognition and instance matching, and is less effective in processing semantically complex information.

LLMs have emerged as a significant development in the field of artificial intelligence, representing a major leap forward in NLP technology. Built upon deep learning and neural network architectures, these models have the capability to comprehend, process, and generate natural language text. Pretraining on vast datasets allows LLMs to grasp the fundamental rules and structures of language. With increasing model size, such as the 175 billion parameters in GPT-3^[44], LLMs become more adept at capturing subtle and complex linguistic patterns. In the research domain of BIM, Zheng *et al.*^[45] propose “BIMS-GPT”, a framework utilizing generative pre-trained transformer (GPT) technologies for efficient natural language-based search in BIM. The framework supports NL-based BIM search by providing dynamic prompt-based virtual assistant functionalities.

Currently, a significant body of research focuses on developing BIM data query systems based on LLMs. These systems can translate users’ natural language questions into executable queries over BIM data^[46]. Since LLMs cannot directly read the complex 3D geometry and semantic data in BIM models, the system must first convert BIM models, such as IFC or Revit files, into formats that LLMs can understand.

Several prototype systems have already achieved integration with mainstream BIM software (e.g., Tekla Structures, Revit), enabling users to interact with models directly through a chat interface within the software environment. For example, a prototype called “AI-Assisted BIM Query System” leverages an LLM, REST APIs, and function calling to allow users to query data from Tekla Structures models and receive human-readable responses^[47].

In addition, some studies explore the use of natural language generation for editing BIM models. This line of research aims to enable users to create or modify BIM models directly through natural language descriptions, thereby lowering the barrier to design. Text2BIM^[48] proposes a multi-agent framework based on a large language model that translates user instructions in natural language into sequences of API calls for BIM software (e.g., Vectorworks), automatically generating 3D building models. By decomposing complex design tasks among specialized agents, such as a design agent, a geometric computation agent, a code generation agent, and a rule-checking agent, the framework improves both task success rates and model quality. The current LLM-BIM integration faces two primary shortcomings: the absence of a reliable semantic consistency mechanism and a standardized accuracy evaluation

system. Moreover, no research has successfully created an end-to-end dynamic loop by integrating lightweight ontologies, LLMs, and GUID extraction.

2.3 Research aim

Based on the organization of existing research, this paper aims to address two main objectives. Firstly, this study seeks to resolve the redundancy issue with the existing IFCOWL. BIM data management via the OWL schema not only facilitates BIM data interchange but also maintains its advantages in integrating more source data and semantic fusion. This is valuable for the future development of BIM. Secondly, this study aims to develop a DHDl framework through LLMs, employing dynamic prompts to process BIM data. This approach is designed to achieve a more natural and intelligent interaction experience.

3. Methodology

Figure 1 presents an overview of our innovative dynamic BIM data interaction framework, utilizing LLMs and ontology. The methodology in this framework is delineated through several distinct phases. The initial phase involves preliminary processing of BIM data by converting the hierarchical relationships of building components defined in the IFC standard into ontological representations.

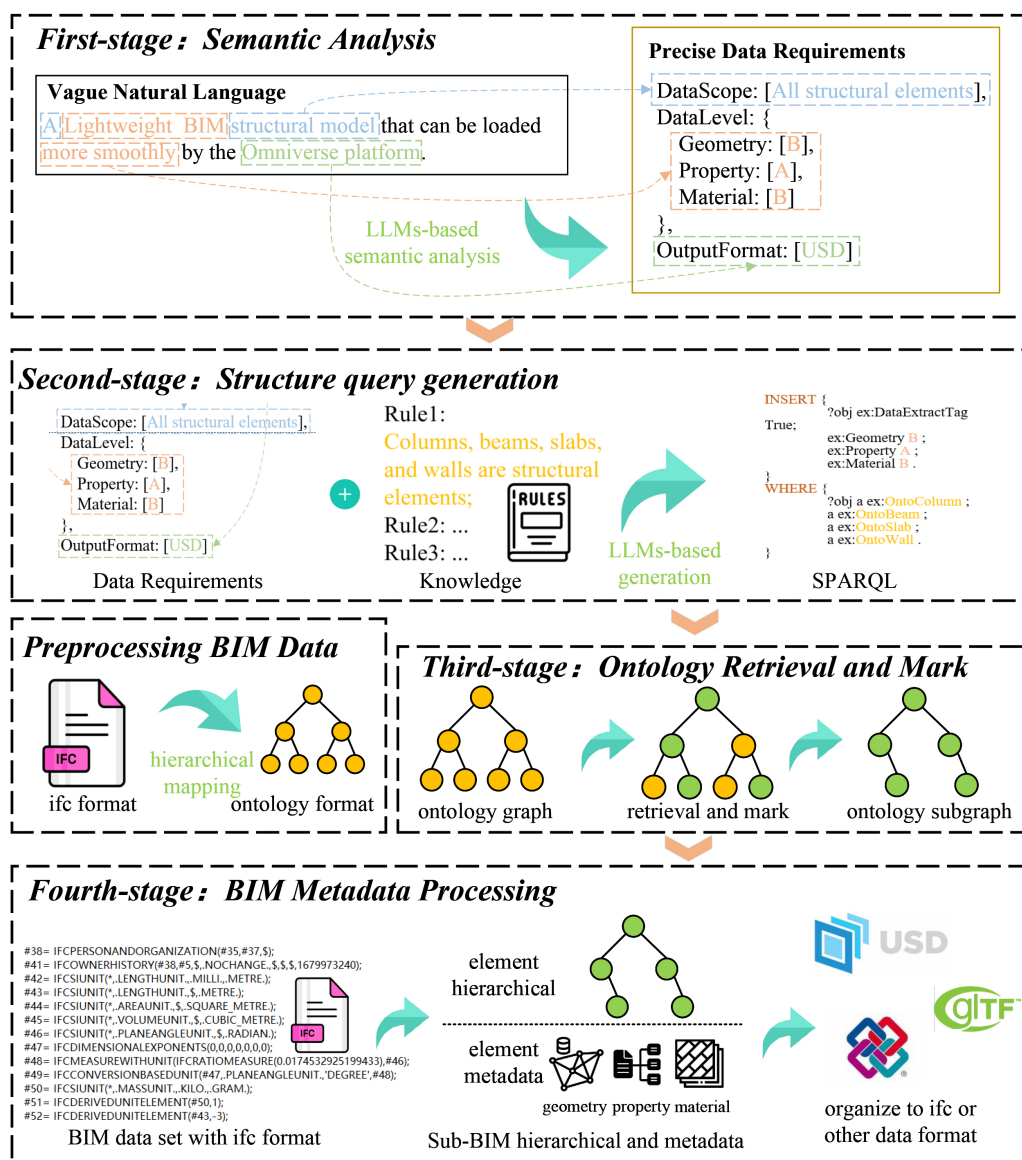


Figure 1. Overview of the dynamic BIM data interaction via LLMs and ontology. BIM: building information modeling; LLMs: large language models.

The first stage encompasses semantic analysis, where user requirements, articulated in natural language, are transformed into computationally executable data queries.

The second stage is the generation of structured SPARQL query statements, tailored for ontology-based contextual inquiries. Distinguished from the initial stage, this process leverages LLMs alongside an external knowledge base, enabling the creation of contextually relevant SPARQL queries pertinent to the specific project and process at hand.

In the third stage, data retrieval and tagging are executed. With BIM data hierarchically organized through ontologies during preprocessing, we conduct direct instance retrieval and tagging using the SPARQL queries. A key part of this stage is the use of the 'DataExtractTag' attribute to filter and construct ontology subgraphs.

The final stage entails extracting and processing BIM metadata from the refined subgraph. Extraction and processing are guided by the precision markers assigned to each object. For instance, objects with a 'Geometry' attribute marked 'C' undergo a process to reduce surface detail for more efficient loading. The processed metadata is then organized into various formats as required for specific user applications, ensuring seamless integration across different platforms.

Figure 2 highlights the different modules of our developed system, including the data source, user interface, semantic analysis, and task processing units. These modules synergize to offer a user-friendly and intelligent BIM data interaction experience.

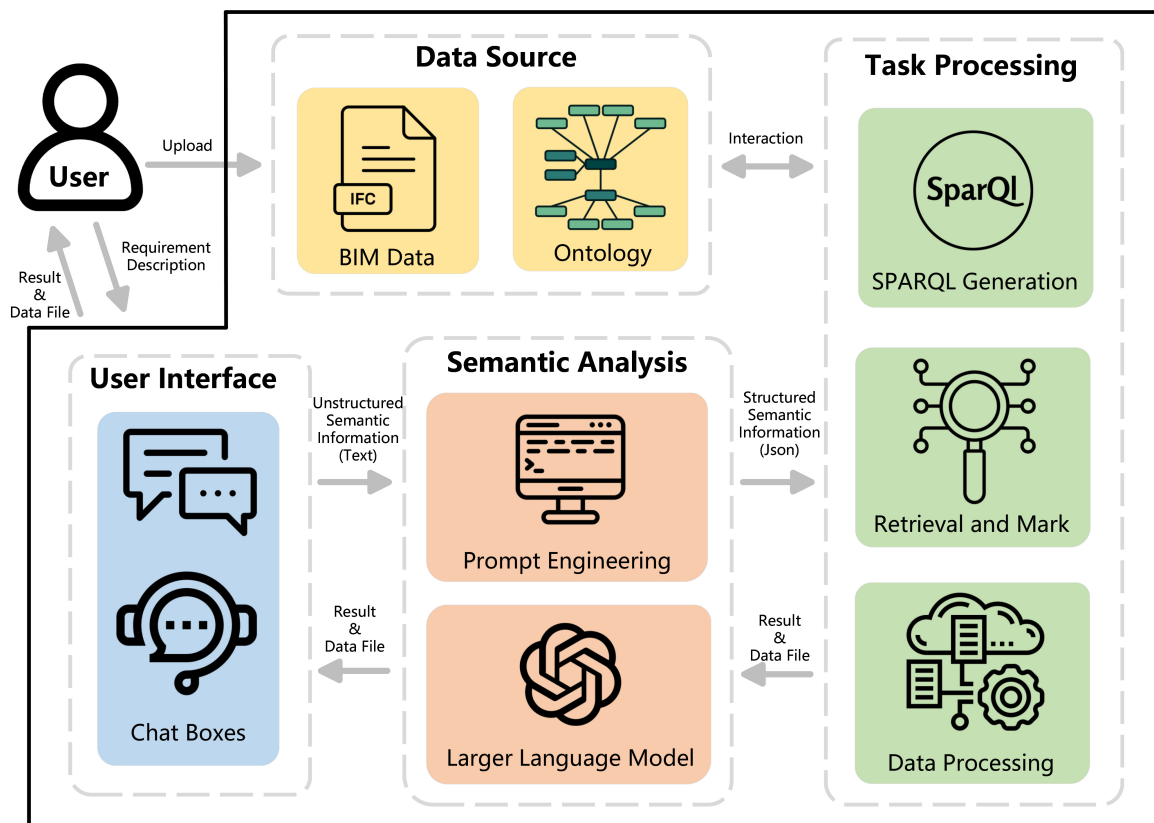


Figure 2. Framework for intelligent semantic interaction system oriented towards BIM. BIM: building information modeling.

3.1 BIM data initialization

The process of BIM data initialization encompasses the parsing of IFC files uploaded by the user, followed by the hierarchical representation of this data through ontology. The ontology modeling adopts a two-layer TBox-ABox architecture: the Terminological Box (TBox) serves as the core conceptual model of the system, strictly following the IFC standard to define classes, attributes, relationships, and constraint rules for building components. Its modular design allows users to extend subclasses and refine attributes according to specific needs. The assertional box (ABox), on the other hand, is dynamically generated at runtime by instantiating concepts from the TBox, transforming the uploaded BIM data into a hierarchical instance database while also serving as an indexing mechanism. This architecture combines the static constraints of the TBox with the dynamic instantiation of the ABox, ensuring semantic consistency while enabling flexible adaptation and efficient retrieval of heterogeneous BIM data.

3.1.1 Ontology TBox development

(1) Class and Hierarchy Definition

To streamline the ontology and enhance the efficiency of querying and reasoning processes, this research selectively excludes data

related to the expression and conversion of geometric materials in IFC. Instead, the focus is placed on instances that represent building elements and their spatial hierarchical relationships. As a result, three main ontology classes have been established: *OntoProject*, *OntoBuildingElement*, and *OntoSpatialStructure*. Within these classes, *OntoBuildingElement* and *OntoSpatialStructure* function as parent categories, each encompassing a variety of specific subcategories. These specific subcategories delineate various spatial aspects of a building project. The detailed definitions of these ontology classes and their hierarchical structures are systematically outlined in [Table 2](#).

Table 2. Class and hierarchy definition.

Class	SubClass
OntoProject	
OntoBuildingElement	OntoColumn, OntoBeam, OntoSlab, OntoWallOntoDoor, OntoWindow, OntoStair, OntoChimney, OntoCovering, OntoCurtainWall, OntoFooting, OntoMember, OntoPile, OntoRailing, OntoRamp, OntoRampFlight, OntoRoof, OntoShadingDevice, OntoStairFlight, OntoBuildingElementProxy
OntoSpatialStructure	OntoSite, OntoBuilding, OntoBuildingStorey, OntoSpace

(2) Object Property Definition

Object properties primarily represent the relationships between instances, such as the containment link between a storey and a building element. In IFC, building elements (*IfcBuildingElement*) are organized using spatial structure elements (*IfcSpatialStructureElement*), with hierarchical relationships established between them via *IfcRelAggregates*. The definitions of object properties in the lightweight ontology, adhering to these IFC rules, are systematically detailed in [Table 3](#).

Table 3. Object property definition.

Object Property	Sub-Property	Domains	Ranges
hasSpatialStructure	hasSite	OntoProject	OntoSite
	hasBuilding	OntoSite	OntoBuilding
	hasBuildingStorey	OntoBuilding	OntoBuildingStorey
	hasSpace	OntoBuildingOntoBuildingStorey	OntoSpace
hasBuildingElement		OntoSpatialStructure	OntoBuildingElement

(3) Data Property Definition

Data attributes fundamentally denote the intrinsic characteristics of instances, with their definitions provided in [Table 4](#). This study classifies them into three distinct types: (i) Unique identifiers for instance objects, which serve for subsequent metadata tracing; (ii) Data extraction indicators, determined using ontological logic, which ascertain whether the data of the current instance is required, considering extraction only for instances flagged as True; (iii) Data precision indicators, wherein each building element instance is categorized by three types of metadata: geometry, material, and property. These indicators define the level of precision for data extraction, with detailed explanations provided in [Table 5](#).

Table 4. Data property definition.

Data Property	Sub-Property	Data Type	Usage
InstanceID	Name	String	UniqueID
	ID		
DataExtractTag		Bool	Data extraction marker
DataPrecisionTag	Geometry	Enumerate (A, B, C)	Data precision marker
	Property		
	Material		

Table 5. Data precision definition.

	Geometry	Material	Property
A	Fine	Photorealistic Rendering	All Properties
B	Medium	Base Color	Partial Properties
C	Simple	White Model	None Properties

3.1.2 Ontology ABox Auto-generation

This study leverages Python-IfcOpenShell for parsing IFC data, focusing on extracting instance objects that epitomize spatial structures and building elements within the IFC framework. Subsequent to this, the automatic generation of the ABox, based on the pre-established TBox, is carried out using Python-RdfLib. The algorithmic procedure of this conversion is illustrated in Figure 3. To ensure the thoroughness and precision of the conversion, a meticulous level-by-level traversal is employed. This ‘Top-Down and Depth-First’ traversal method commences from the *IfcProject* object, methodically progressing to encompass all associated objects. The key attributes, such as Name, ID, and the interrelationships of these IFC instances, are then transformed into the ontological ABox.

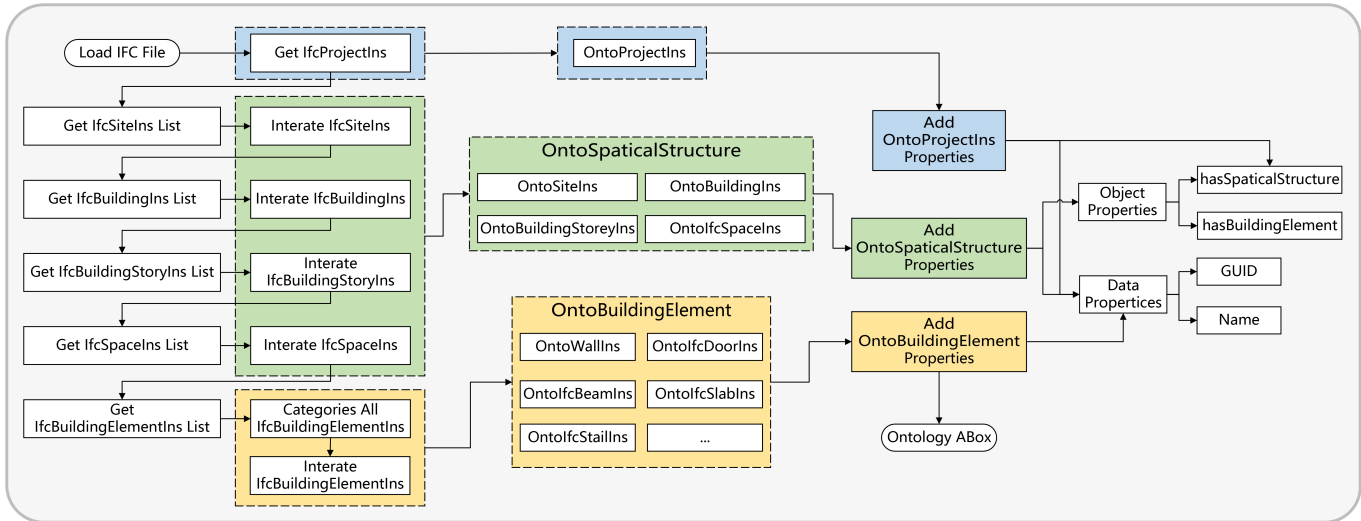


Figure 3. Mapping algorithm of IFC to ontology ABox. IFC: industry foundation classes; ABox: assertional box.

The implementation method is as follows:

Step 1: Initially, retrieve *IfcProjectIns* from the IFC dataset and obtain its attribute values: Name and ID.

Step 2: Create *OntoProjectIns*. *OntoProjectIns* has the same data attributes as *IfcProjectIns* to ensure consistency among instance objects.

Step 3: Retrieve the list of *IfcSite* instance (*IfcProjectInsList*) associated with *IfcProject*, iterate through the *IfcProjectInsList*, and create corresponding *OntoSiteIns*.

Step 4: Add an object attribute (*hasSite*) to *OntoProjectIns*, assigning all *OntoSiteIns*.

By deeply traversing the IFC instance reference network, this method enables accurate mapping from non-linear hierarchical structures to the ontology ABox. It is pivotal to recognize that, for entities depicting spatial structures (like *IfcSite*, *IfcBuilding*, *IfcBuildingStorey*, *IfcSpace*), the process involves not just the extraction of relationships among these spatial elements, but also a keen understanding of their linkages with the corresponding *BuildingElement*. Notably, in the transformation from *IfcProjectIns* to *OntoProjectIns*, the step for associating with building elements is not required since *IfcProject* does not classify as a spatial element.

3.2 Semantic analysis based on large language model

This chapter focuses on the methodologies to utilize LLMs for interpreting unstructured semantic information provided by users, with the aim of improving their applicability in downstream data processing tasks. Figure 4 exemplifies this process, showcasing the workflow for refining ambiguous semantic information and the expected results of such processing. Our discussion will revolve around two main aspects: the development of effective prompting strategies and the intricacies of interacting with LLMs.

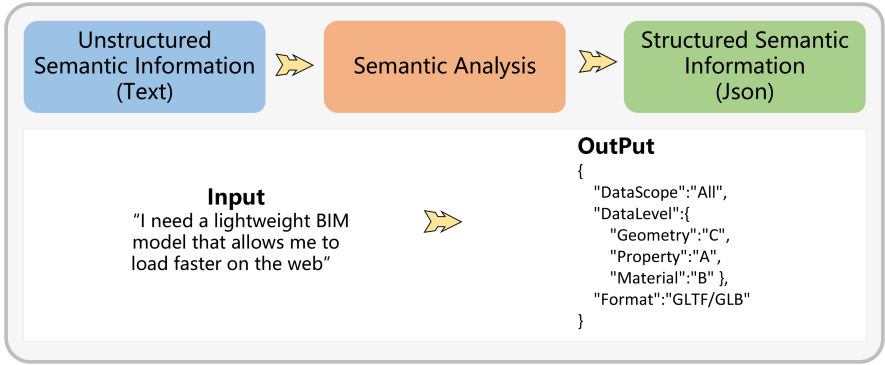


Figure 4. An example of semantic analysis.

3.2.1 Prompt management

Prompt engineering (PE) is a technique where a specific prompting function, referred to as $f_{\text{prompt}}(X)$, is created to improve outcomes in subsequent tasks^[49]. This approach reduces the discrepancies between the data used in pre-training and fine-tuning stages, thereby increasing the LLM’s effectiveness in downstream tasks.

There are two main types of prompting templates: cloze and prefix prompts. Cloze prompts follow a fill-in-the-blank style, like in the example “The capital of China is [Z]”, where the model fills “[Z]” with “Beijing”. This type is widely used in knowledge extraction and question-answering systems. Prefix prompts, on the other hand, provide a starting phrase and prompt the model to continue generating content, that is useful across a broader range of language understanding and generation tasks. Our research aims to enable machines to interpret unstructured semantic input from users and produce structured information that initiates downstream tasks, combining both cloze and prefix approaches. For instance, as shown in Example 4, the semantic interaction module needs to identify: 1) the data segment for extraction; 2) the precision level of different data types, categorized into three levels as shown in Table 4; and 3) the desired data output format. To meet these requirements, we design specific Prompt templates that guide the LLM in accurately retrieving and inferring information.

In this study, we have developed a Prompt template comprising five key elements, as illustrated in Figure 5. The ‘Task’ component communicates the task details to the LLM, including the text to be analyzed ([X]), predictive values ([Z1-Z5]), and the desired output format (Json). The ‘Knowledge’ module is designed to provide universally relevant, intuitively deducible knowledge to improve the LLM’s understanding of the task. It also allows for the incorporation of source knowledge, such as an ontology file, enabling the LLM to integrate this information during semantic analysis and thus merge with the local knowledge base. The ‘Instruction’ module offers specific directives to guide the LLM in producing the intended outcomes, acting as a logical framework that breaks down complex tasks into simpler steps for more efficient completion by the LLM. ‘Answer Mapping’ is the process of aligning the LLM’s outputs with a predefined list of answers, crucial for transitioning from vague semantic information to concrete instructions and facilitating subsequent task processing. Finally, the ‘Response Format’ provides a template for the LLM’s formatted content output, aiding in the generation of more nuanced and accurate responses.

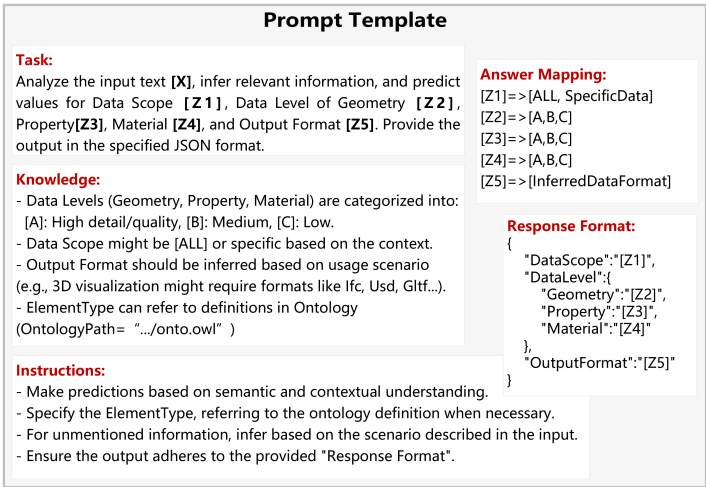


Figure 5. Prompt template.

3.2.2 Semantic analysis

After creating the prompts, the next step is to integrate user input [X] with the prompt to create a modified input [X'], which is then fed into the LLM. The LLM processes [X'] using the context and directives from the prompt to generate outputs at specified locations [Z1]–[Z5]. However, it's important to recognize that even with the LLM's advanced semantic understanding and the detailed construction of the prompt, there is no guarantee that the outputs will always meet user expectations. Therefore, an essential feature of our system is a user feedback mechanism, as depicted in Figure 6. Instead of directly passing the LLM's response to the task processing module, it first goes through a user validation step. If there's a discrepancy between the LLM's output and the user's expectations, the user can provide additional guidance to correct the divergence. The LLM then adjusts its response based on this feedback, working towards an answer that better aligns with the user's needs. Only after the user confirms the accuracy of the response does the semantic processing model forward the information to the task processing module for subsequent actions. Implementing this layered interaction method significantly improves the precision of subsequent task execution. It should be noted that this study used the Qwen2.5-14B model (locally deployed via Ollama; inference parameters: temperature = 0.1, top-p = 0.9, max tokens = 2048). This model was selected for three reasons: (i) its superior performance on Chinese natural language understanding benchmarks, which is critical given that the domain experts in this study operate primarily in Chinese; (ii) its feasibility for local deployment under limited computational resources (a single NVIDIA RTX 4090 GPU), which avoids data privacy concerns associated with cloud-based APIs; and (iii) its strong instruction-following capability that aligns well with our structured prompt templates. While models such as GPT-4 may offer higher general-purpose capability, Qwen2.5-14B achieved satisfactory task accuracy in our workflow at substantially lower operational cost.

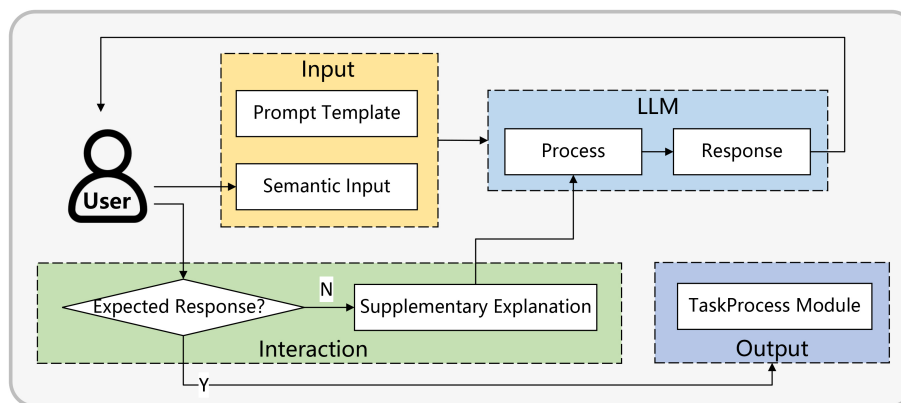


Figure 6. Interaction logic of semantic analysis. LLM: large language model.

3.3 Task processing

The task processing module processes structured semantic information from the semantic analysis module. It initially formulates SPARQL query statements aligned with data requirements to enable ontology queries. It then retrieves data objects corresponding to user needs from the ontology and labels them per data precision specifications. Finally, utilizing markers within the ontology, metadata from the IFC file is uniformly processed and integrated, leading to the generation of a data file customized to user specifications.

3.3.1 SPARQL generation

For ontology retrieval, SPARQL queries are essential, requiring the integration of class and property names from the ontology. The semantic analysis module's final output, "DataScope", provides a formal semantic description but is not ready for direct retrieval as a SPARQL statement. Therefore, converting this formalized semantic expression into a SPARQL query is crucial for enabling the retrieval of data expressed in the ontology.

Converting semantic expressions into SPARQL queries is a key research area in NLP and knowledge graphs. Prior studies^[43] have used entity recognition and linking, along with relation recognition, to process natural language into SPARQL statements for data retrieval. This method typically involves keyword mapping and steps like synonym substitution, which can be cumbersome and limited to simple sentence structures. It falls short in handling complex sentences or those with implicit information, such as extracting BIM models containing structural elements. Since predefined ontologies or IFC specifications cannot directly match a building element category to "structure", this process relies on deduction to identify and extract elements like columns, beams, slabs, and walls to form a structural model.

Therefore, a more advanced method is necessary to handle sentences with implicit information requiring deeper reasoning and analysis. This study continues to employ the LLMs to assist in achieving these goals. The specific process is illustrated in Figure 7.

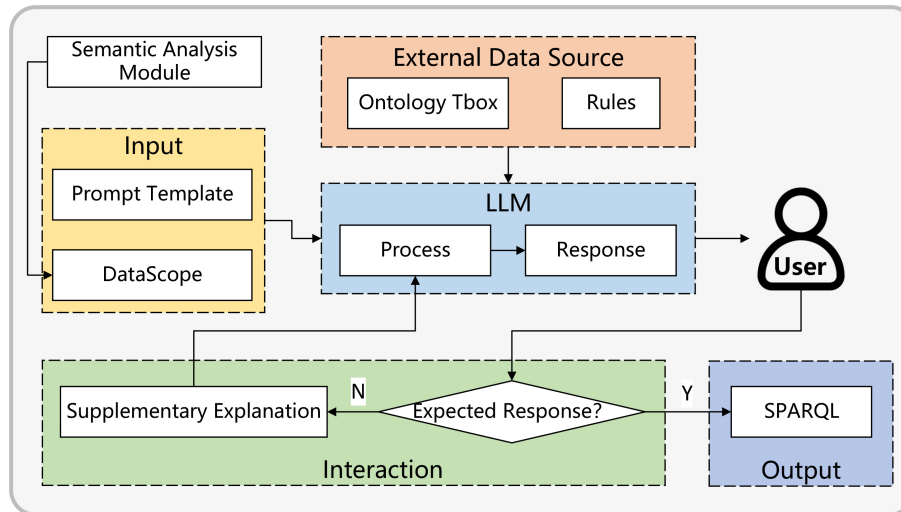


Figure 7. Interaction logic of SPARQL generation. LLM: large language model; SPARQL: SPARQL Protocol and RDF Query Language.

In this phase of interacting with the LLMs, relying solely on the LLMs for interpreting statements with implicit information can lead to imprecision. To improve the inference accuracy, the LLMs first access a subset of external knowledge, allowing for more efficient reasoning based on this acquired knowledge. This external data is not dependent on prompt templates but is instead created and imported based on user requirements, significantly enhancing the system's adaptability. The external knowledge consists of two components: the first is the ontology TBox, which includes the definitions within the ontology; the second is user-defined rules that guide the LLMs in translating complex semantics into SPARQL queries. In our study, external knowledge, including the predefined rules of the proposed graph structure and user-defined rules, is recorded in textual form. This method ensures that all necessary information is accessible and clearly defined, facilitating its integration into the LLM framework. For example, the following rule was injected into the LLM context: "Columns (OntoColumn), beams (OntoBeam), slabs (OntoSlab), and walls (OntoWall) are defined as structural elements in this project". This guided the LLM to map the implicit request "structural model" to a SPARQL UNION query targeting these four classes.

To mitigate syntax errors, our prototype system framework incorporates an automatic exception capture mechanism. When an LLM generates a SPARQL statement, it is immediately executed. If a syntax error is detected, the system captures this exception and automatically re-submits the erroneous statement along with the exception details back to the LLM for correction. This process continues iteratively until the SPARQL query executes successfully without errors. For semantic misunderstandings, which can be more complex to address, we employ a manual intervention strategy. As depicted in Figure 7, potential semantic errors identified in the outputs from the LLM are corrected through continuous human-machine interaction. This interaction involves domain experts who review and adjust the queries to ensure their semantic accuracy. We acknowledge that while LLMs offer significant capabilities, they do not always produce perfectly aligned responses; accordingly, our framework incorporates domain expert oversight as a deliberate validation layer, shifting expert involvement from routine execution to targeted semantic review (discussed further in Section 5.1). This combination of automated error correction for syntax and expert-driven adjustment for semantics provides a comprehensive approach to maintaining the accuracy of SPARQL statements generated by LLMs.

3.3.2 Retrieval and mark

After formulating the SPARQL queries, the subsequent step involves data retrieval and annotation within the ontology. Data annotation is achieved by incorporating data properties. Following the initial processing of BIM data, the Ontology ABox is generated. Each instance in the Ontology ABox is attributed with three data properties: *name*, *guid*, and *DataExtractTag*, where the initial value of *DataExtractTag* is set to False.

During this stage, the SPARQL queries identify instances in the Ontology ABox and update the *DataExtractTag* value to True, thereby marking these instances. Concurrently, the data properties of *DataPrecisionTag* are assigned, guided by the *DataPrecision* output from the semantic analysis module.

Figure 8 illustrates the process of data annotation. In Figure 8 (left), instances are shown where the *DataExtractTag* is set to True, indicating that they are marked for extraction. Conversely, Figure 8 (right) shows instances with the *DataExtractTag* set to False, indicating that they are not selected for extraction. Ontology instances marked with the *DataExtractTag* property as True are further distinguished by having three additional properties that define the precision of data extraction, unlike instances marked as False.

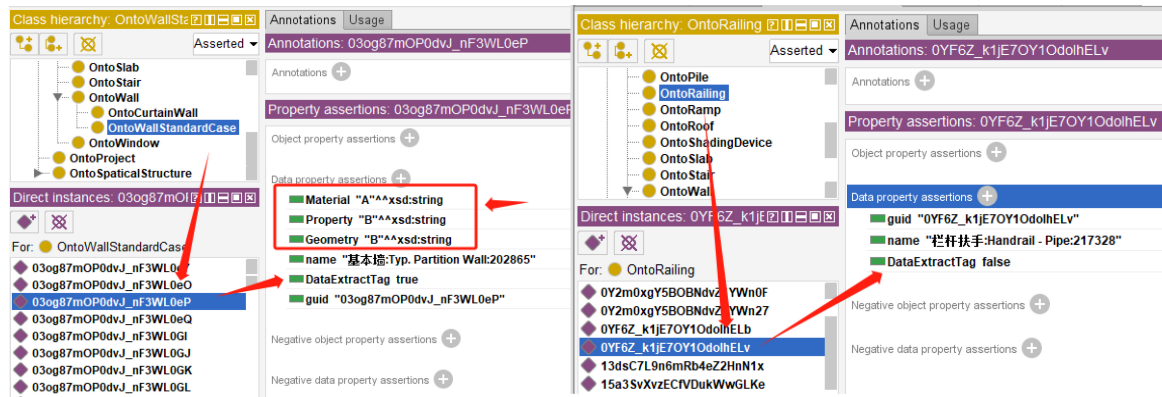


Figure 8. Example of data annotation.

3.3.3 Data processing

In this research, we have implemented an automated process within the ontology to identify and mark instances that require extraction, thereby defining the data extraction requirements and specifying the precision level for each instance.

The next critical phase is the extraction and processing of metadata based on the annotated instances. This step hinges on the original IFC dataset provided by the user, as the ontology in our framework delineates only the instances of building elements and not their associated metadata, which encompass geometry, materials, attributes, location, and more. The choice to focus on instance definition over metadata in the ontology was influenced by several factors. Primarily, metadata accounts for a significant portion of storage space, exceeding 70%. Furthermore, incorporating metadata introduces considerable complexity and redundancy in data parsing. Most crucially, metadata often fails to yield meaningful insights into user-specific data requirements. This approach maintains the integrity of the IFC data file's structure while leveraging the advantages of ontology technology to facilitate convenient and adaptable task processing.

Interaction between IFC and data expressed in the ontology is facilitated through GUID, with the specific steps as follows:

Step 1: Select *OntoBuildingElement* instances to be extracted in the ontology and organize them according to their corresponding hierarchical relationships, ultimately forming a structural tree of a sub-model.

Step 2: Traverse the subtree, retrieve the ID of the sub-node instance element, and obtain metadata related to that instance in the IFC dataset.

Step 3: Process these metadata according to the data precision marker in the ontology, ensuring they adequately meet user requirements.

Step 4: Organize and output these metadata according to user requirement formats.

In our research, we handle three distinct types of metadata, geometry, material, and property, each offering three processing schemas, detailed in Table 4. This approach results in 27 unique combinations, effectively catering to a wide range of user requirements. Such versatility not only ensures compliance with user specifications but also optimizes the volume of data, thereby enhancing loading efficiency.

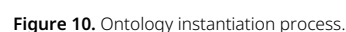
However, it's crucial to acknowledge that this methodology primarily addresses the processing of BIM metadata for computer graphics rendering, which remains abstract for end-users. They necessitate a more accessible, standardized data file format, such as IFC, Graphics Library Transmission Format (GLTF), or Universal Scene Description (USD), which can be readily utilized in various applications. Consequently, further data conversion is required to align with these diverse format specifications. Given the growing prominence of USD in facilitating data sharing and collaboration, our study extends to exploring the transformation of IFC metadata into USD format files. This endeavor aligns with NVIDIA's ongoing efforts to amalgamate USD with AI technologies, thereby amplifying its utility in digital twin applications.

4. Case Study

In this study, we illustrate the effectiveness of our approach through a detailed case study. We selected a library building as our focus, encompassing an area of approximately 12,600 square meters, featuring a basement and six above-ground floors, and comprising 20,933 architectural elements. The BIM model of this building, originally created using Autodesk Revit 2020, is showcased in Figure 9 a. For our research purposes, we converted this model to the IFC format using Revit's built-in features, as shown in Figure 9b. This conversion facilitates effortless data transfer and compatibility for the subsequent stages of our methodology.



Users can upload their IFC data directly to the system's initialization module. Initially, it involves parsing the IFC data to identify architectural elements and spatial hierarchies, including their interconnections. Subsequently, these elements are methodically aligned with a pre-established ontology framework, known as the TBox, in a process called ontology instantiation. [Figure 10](#) illustrates this transformation process.



After completing the preprocessing of BIM data, we established a foundational dataset crucial for the semantic interaction system. This phase is essential in our methodology. Unlike other research that utilizes the IFC2RDF tool for full conversion from IFC to ontology, our method selectively converts specific elements, omitting geometric data expressions typically used for graphic rendering. This strategy addresses the issues of excessive system size and data redundancy effectively. In the case illustrated in [Figure 11](#), we employed the BIM data of a library building to compare the computer resource usage between two data conversion methods: full conversion via IFCtoRDF and our proposed partial conversion. The results highlight the efficiency of our approach.

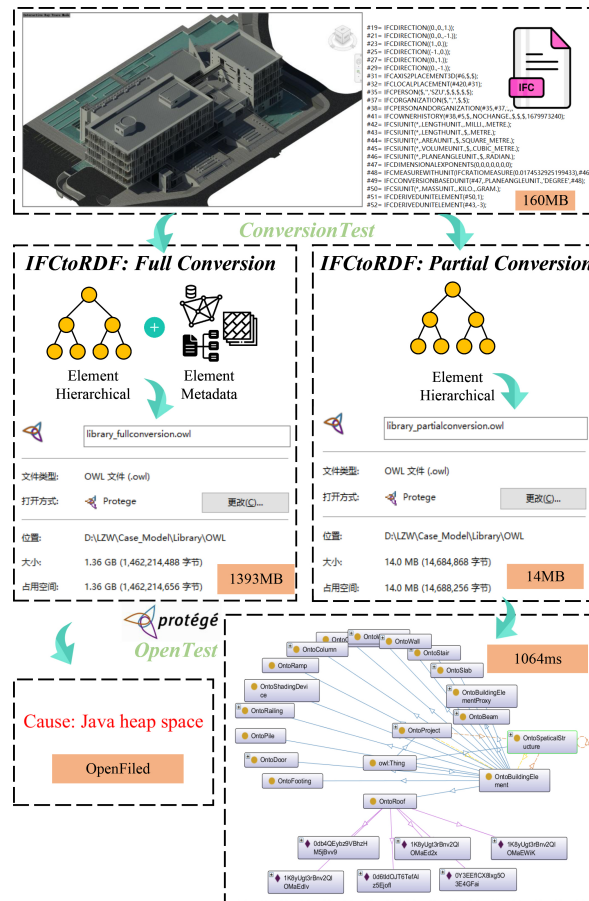


Figure 11. Comparison the methods of conversion and partial conversion.

Our system's performance was evaluated based on time complexity (the duration required to complete a task) and space complexity (the computer memory used in task completion). Although quantifying the superiority of the methods based on time complexity was challenging due to its dependency on various factors, we focused primarily on space complexity for assessment. Figure 11 shows that the initial IFC data file of 160 MB expanded to 1,393 MB, an increase of about 8.7 times, following full RDF conversion. Conversely, our partial conversion method resulted in a data file that was only 8.75% of the original size. In this study, Protégé version 3.5.1 was used to open the OWL files. The file-opening tests showed that the fully converted file could not be opened due to memory overflow, whereas the partially converted file could be opened in only 1 second. This underscores the practicality and effectiveness of our lightweight data conversion approach, which not only resolves system redundancy but also significantly improves the efficiency of ontology-based data retrieval. These results are consolidated in Table 6 for an integrated performance overview.

Table 6. System performance comparison.

Performance Dimension	Metric	Full IFCtoRDF Conversion	Proposed Lightweight Method
Space Complexity	Converted file size	1,393 MB	14 MB
	Size ratio to original IFC	870.6%	8.75%
Loadability	Protégé loading result	Memory overflow (failed)	Successfully loaded in 1s
Rendering Efficiency	Geometric primitives	876,131	148,876 (↓ 83.0%)
	GPU memory usage	98.59 MB	66.44 MB (↓ 32.6%)
	Primitive variable storage	18.8 MB	3.58 MB (↓ 81.0%)
LLM Interaction	Avg. rounds per task	—	2.3 rounds*
	Expert intervention rate	—	1 out of 3 rounds (33.3%)*

*: Based on the case study interaction log (3 interaction rounds recorded in Section 4.2); ↓ : decline.

4.2 Intelligence semantic interaction and ontology-based data retrieval

After obtaining the foundational dataset, users can engage with the system via natural language, facilitating the extraction and processing of BIM data through dialogue-based interactions. As illustrated in [Figure 12](#), we sought to extract the structural components of the model. This task was initiated by inputting our requirements into the LLMs, which then generate outputs based on predefined templates. Initially, the LLMs' responses did not fully meet our expectations, prompting us to provide additional clarifications in a subsequent interaction. In this second round, the LLMs produced a more accurate response. The process then progresses to converting these specific data requirements into structured ontology query statements (SPARQL), again utilizing the capabilities of LLMs.

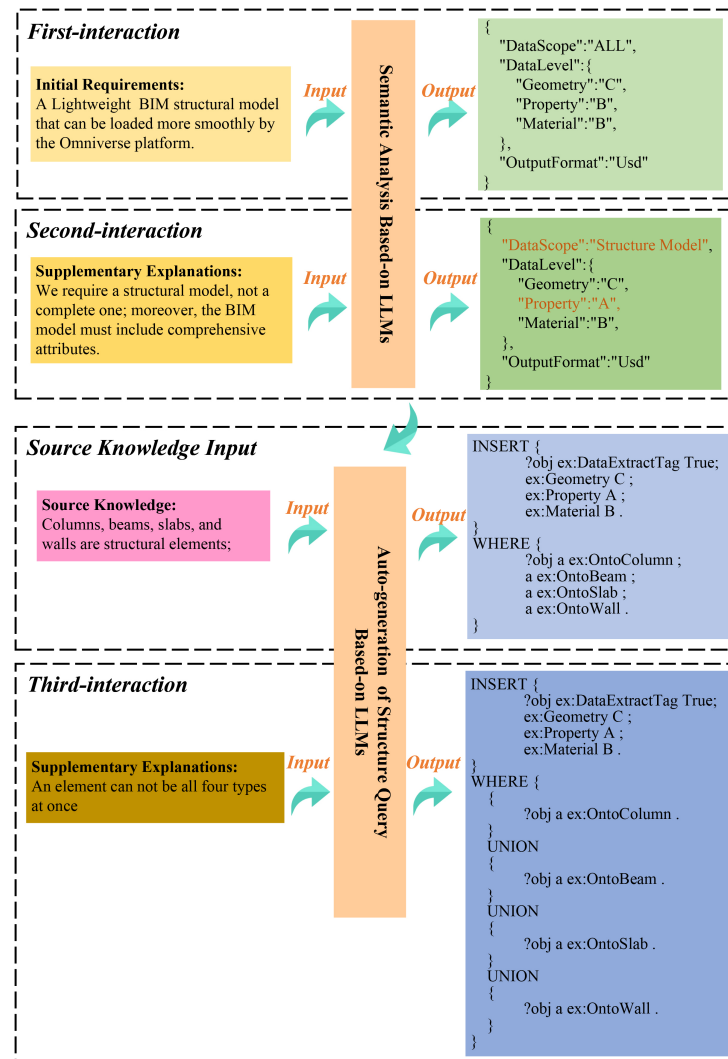


Figure 12. Processing of semantic analysis and data retrieval.

In our case study, we established a rule stating that “columns, beams, slabs, and walls are considered structural elements in this project and are essential for the structural model extraction”. These rules may be broadly applicable or project-specific. To enhance the system’s versatility, we crafted these rules through a localized knowledge base. During interactions with LLMs, we selectively integrated this knowledge base as required, enabling the LLMs to generate responses that more accurately satisfy the user’s specific needs. As seen in our case, after inputting our rules, the LLMs, in conjunction with the data requirements, generated SPARQL queries suitable for ontology retrieval and tagging. Applying these SPARQL queries to the instantiated ontology enabled the marking of structural elements designated for extraction.

4.3 Metadata extract and processing

The process of metadata extraction and processing commences once data tagging within the ontology is complete. [Figure 13](#) delineates the workflow for this process. It starts with identifying the GUID of marked ontology objects, which is then utilized to extract all corresponding metadata for each architectural object from the IFC dataset. The metadata types are then processed

according to their respective data tags. In our example, the geometry data level was marked as ‘C’, indicating a need for the simplification of the geometric data of each object to enhance loading efficiency. A comparison of geometric rendering parameters before and after this optimization is presented in Table 7. Additionally, Figure 13 includes visual representations under ‘geometry’ and ‘Simple geometry’ to illustrate these modifications. The final step involves organizing and formatting the processed data into the USD format.

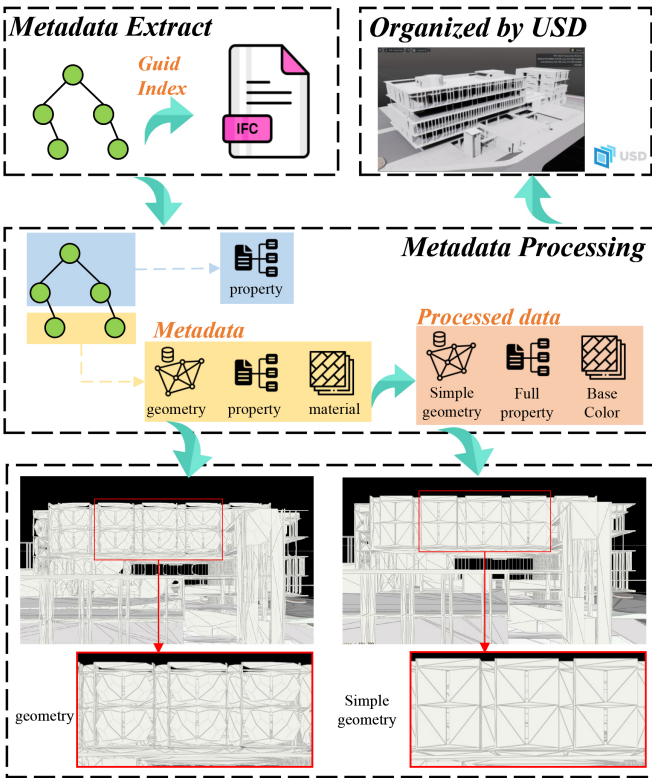


Figure 13. Process of metadata extract and processing.

Table 7. Render parameters of geometry.

	Primitives	GPU Memory	Primitive Variable	Topology
Geometry	876,131	98.59 MB	18.8 MB	21.37 MB
Simple Geometry	148,876	66.44 MB	3.58 MB	21.37 MB

Primitives are basic shapes like points, lines, and triangles, used to build 3D models; GPU Memory is where the graphics card stores data for rendering, affecting how fast and well it can render images; Primitive Variables are details like color and texture that define how each primitive looks; Topology is about how points are connected to form shapes. It defines the structure of the graphics.

5. Discussion

5.1 Effectiveness of dynamic BIM data interaction

This study introduces an innovative approach to BIM data interaction by leveraging LLMs to enhance both the efficiency and intelligence of BIM processes. Compared with existing work, the contributions of this study lie not only in introducing LLMs to enable natural language interaction, but more importantly in restructuring the organization of IFC data and the semantic query pathways. This design supports scalable and low-threshold access to BIM data. The key highlights of the study are:

Advanced Semantic Interaction : LLMs are used to understand users’ natural language requests and to iteratively refine these requests during interaction. The finalized requirements are then translated into executable SPARQL queries and applied to the instantiated ontology, enabling the tagging and retrieval of target components. Unlike BIMS-GPT, which mainly focuses on generating answers or explanations, this study operationalizes LLMs’ output as structured queries. As a result, the retrieval pipeline has explicit execution semantics and can be reviewed and verified. It should be noted that “lowering the expertise barrier” does not imply eliminating domain expert oversight entirely. Specifically, our system removes the routine operational burden from typical BIM users, such as learning IFC schema structures, writing SPARQL from scratch, or manually navigating ontology graphs. A semantic

validation layer by domain experts remains available for mission-critical applications to ensure the correctness of SPARQL-mediated retrieval. This design is analogous to code review in software engineering: automation shifts expert focus from execution to validation rather than replacing it altogether. In the case study, expert intervention was required in only one of three interaction rounds, specifically in Round 3, where the LLM generated an incorrect UNION clause syntax, suggesting that the system substantially reduces, though does not fully eliminate, the need for expert involvement.

Efficient Data Processing: Introducing a method that integrates IFC with a lightweight ontology to streamline BIM data extraction. The key strategy proposed in this study is selective extraction and lightweight semantic mapping. Specifically, only the BIM hierarchical structure and the core entities/relations required for high-frequency queries are instantiated in the ontology, while geometric representations that mainly serve graphical rendering are omitted. This design differentiates our approach from typical full IFCtoRDF/IFCOWL conversions. This approach significantly reduces irrelevant data conversion, simplifies operations, and enhances the efficiency of ontology querying and reasoning.

Comparative Analysis: Using a library building as a case study, this paper compares the proposed method with a traditional BIM data extraction approach in terms of space and time complexity. For space complexity, the original IFC file size was 160 MB. After full conversion using a conventional IFCtoRDF method, the resulting RDF/OWL file size increased to 1,393 MB, representing an increase of approximately 770.6% ($((1393-160)/160 \approx 7.706)$). In contrast, the partially converted data produced by our method (i.e., lightweight mapping) was about 14 MB, which is only 8.75% of the original IFC file. For time complexity, the fully converted file could not be opened due to memory overflow, whereas the lightweight-mapped file could be opened in 1 s. Overall, the proposed method demonstrates superior performance. A consolidated summary of these metrics is provided in Table 6. Regarding time-related benchmarking, we acknowledge this as a current limitation and outline a systematic future benchmarking plan in Section 5.2.

Integration of Specialized Knowledge: This study defines and introduces a set of rules, and injects them into LLMs on demand through a local knowledge base. This design guides the model to generate retrieval intents and SPARQL queries that better match the project context. By aligning the LLMs with the domain using a “local rule base + ontology vocabulary/relations”, this study’s method improves semantic consistency and retrieval accuracy for a specific project. This also further differentiates our approach from BIMS-GPT: rather than relying entirely on the model’s intrinsic capability for correctness, we emphasize controllable knowledge injection and structured execution.

In summary, this approach effectively bridges the gap between natural language input and structured BIM output, advancing BIM processes toward greater intuitiveness, efficiency, and user-friendliness. In terms of methodological innovation, this paper proposes a task-oriented IFC semantic mapping approach and a lightweight ontology representation paradigm. It further realizes semantic interaction through a pipeline of “LLM semantic analysis→ontology constraints→SPARQL execution”, rather than relying on the generated answers themselves. In terms of technical innovation, this study implements automated IFC parsing, ABox instantiation, domain alignment based on rules, a local knowledge base, and an end-to-end data pipeline of “ontology tagging→GUID backtracking→geometry simplification→format output”. This integrated pipeline allows semantic retrieval results to directly support downstream engineering processing and project delivery.

5.2 Limitations and future research directions

While the implementation of LLMs in BIM data interaction marks a significant step forward in terms of efficiency and user experience, it is crucial to acknowledge the inherent limitations and challenges of this innovative approach. The primary areas for further development in this study are as follows:

Firstly, our current prototype system, though innovative, does not yet possess the full range of functionalities needed to meet the diverse needs of users comprehensively. Future work should focus on expanding the system’s capabilities to address a broader range of BIM tasks, thereby enhancing its practicality and applicability.

Secondly, the effectiveness of intelligent semantic interaction heavily depends on the selection and application of LLMs. While manually designed prompt templates improve output accuracy, they have limited generalization capability when adapting to diverse user inputs. Future efforts should focus on developing automated prompt generation algorithms that leverage machine learning to understand user intent and context, thereby enhancing the LLMs’ responsiveness. Additionally, incorporating adaptive learning mechanisms will enable the system to perform more professional and precise reasoning when integrating with local knowledge bases, especially in handling complex and context-dependent queries.

In addition, although this study has demonstrated that lightweight mapping significantly outperforms full IFCtoRDF conversion in terms of data size and loadability, it still lacks a systematic performance benchmark under a unified hardware environment, particularly for LLM query parsing latency, SPARQL execution time, and ontology loading time across models of varying complexity. In future work, we will establish a rigorous benchmarking framework comprising: (i) a standardized test dataset covering IFC models of varying scales (small: < 10 MB; medium: 10–100 MB; large: > 100 MB) to assess how performance scales with model complexity; (ii) a representative query set spanning different levels of semantic complexity, from simple instance retrieval to multi-hop relational

queries, enabling systematic measurement of SPARQL execution time and LLM response latency; and (iii) a controlled hardware environment with fixed configurations for CPU, GPU, memory, and triple store settings, ensuring reproducible and comparable results. This benchmarking plan will enable quantitative reporting of metrics such as end-to-end task completion time, query response time, and system latency under concurrent access conditions.

Lastly, interdisciplinary collaboration will play a key role in advancing the system's development. By integrating achievements from fields such as artificial intelligence, architecture, and engineering, it is possible to build a comprehensive platform capable of not only processing BIM data but also incorporating knowledge from related domains such as structural engineering and environmental sustainability. This would significantly enhance the system's professional value. This study has taken initial steps toward exploring this direction; however, constructing a more systematic knowledge base remains a key focus for future work.

6. Conclusion

This research, driven by the objectives of minimizing operational costs, elevating system efficiency, and focusing on user-centric design, introduces an innovative method for BIM data extraction. This method ingeniously interplays with IFC and a lightweight ontology to effectively counter the common issue of data redundancy prevalent in full IFC-to-ontology conversions. By refining the data extraction process, the study markedly decreases superfluous computational demands, thus boosting the overall efficiency of BIM applications.

Building on this groundwork, the study establishes an advanced semantic interaction system utilizing LLMs. This system transforms user interaction with BIM data, offering a more intuitive, natural language dialogue interface. Employing LLMs leverages their formidable semantic comprehension capabilities, empowering the system to precisely discern and fulfill diverse user requests. This not only elevates the user experience by streamlining interactions but also ensures that the system thoroughly understands user needs.

The system can convert parsed user commands into structured queries, enabling seamless integration from input to execution, thereby making data querying, extraction, and interaction more efficient. This bridges the gap between complex BIM data management and user-friendly interfaces.

Furthermore, the approach demonstrates potential for cross-industry applications. In facility management, the lightweight ontology and SPARQL-based retrieval mechanism are expected to support operation and maintenance scenarios with lower data and computational costs. Meanwhile, natural language interaction can reduce the learning barrier for management personnel. In digital twin applications, this study provides a verifiable data access interface through executable queries, which enhances interoperability and integration efficiency across platforms. Moreover, by adjusting responses and system behavior according to user input, the system can deliver a customized experience that appropriately meets the diverse needs commonly found in BIM projects.

Declarations

Acknowledgements

Qwen2.5-14B was used as a research tool within the proposed framework for natural language understanding and SPARQL generation. All generated outputs were reviewed and validated by the authors before use in the study. The authors take full responsibility for the accuracy of the results and conclusions.

Authors contribution

Ding Z: Conceptualization, funding acquisition, project administration, resources, writing-review & editing.

Li Y: Formal analysis, data curation, visualization, writing-original draft, writing-review & editing.

Liu Z: Investigation, methodology, software, writing-original draft, data curation.

Yuan H: Validation, supervision, writing-review & editing.

Conflicts of interest

Hongping Yuan is an Associate Editor of *Journal of Building Design and Environment*. The other authors declare no conflicts of interest.

Ethical approval

Not applicable.

Consent to participate

Not applicable.

Consent for publication

Not applicable.

Availability of data and materials

All data and models generated and used to support the findings of this study are available from the corresponding author upon reasonable request.

Funding

This research was conducted with the support of the Shenzhen Natural Science Foundation (the Stable Support Plan Program No. 20220810160221001).

Copyright

@ The Author(s) 2026.

References

1. Cao H, Huang M. Building information modeling technology capabilities: Operationalizing the multidimensional construct. *Sustainability*. 2023;15(20):14755. [DOI]
2. Adepoju O. Building information modelling. In: *Re-skilling human resources for construction 4.0*. Cham: Springer; 2021. p. 43-64. [DOI]
3. Jaskula K, Papadonikolaki E, Rovas D. Comparison of current common data environment tools in the construction industry. In: 2023 European Conference on Computing in Construction and the 40th International CIB W78 Conference; 2023 Jul 10-12; Heraklion, Greece. Greece: European Conference on Computing in Construction; 2023. [DOI]
4. Laakso M, Kiviniemi A. The IFC standard - A review of history, development, and standardization. *J Inf Technol Constr*. 2012;17:134-161. Available from: <https://www.itcon.org/2012/9>
5. Kone V, Mahesh G. An ontology-driven bi-directional workflow for integrating project management data into the IFC standard. *J Inf Technol Constr*. 2025;30(73):1768-1795. [DOI]
6. Pauwels P, Krijnen T, Terkaj W, Beetz J. Enhancing the ifcOWL ontology with an alternative representation for geometric data. *Autom Constr*. 2017;80:77-94. [DOI]
7. Venugopal M, Eastman CM, Sacks R, Teizer J. Semantics of model views for information exchanges using the industry foundation class schema. *Adv Eng Inform*. 2012;26(2):411-428. [DOI]
8. Filardo MM, Debus P, Melzner J, Bargstädt HJ. XML-based Automated Information Requirement Import to a Modelling Environment. In: *Proceedings of the 30th International Conference on Intelligent Computing in Engineering (EG-ICE) 2023*; 2023 Jul 4-7; London, United Kingdom. Germany: European Group for Intelligent Computing in Engineering; 2023. Available from: https://www.ucl.ac.uk/bartlett/sites/bartlett/files/xml-based_automated_information_requirement_import_to_a_modelling_environment.pdf
9. Sun Z, Wang C, Wu J. Industry foundation class-based building information modeling lightweight visualization method for steel structures. *Appl Sci*. 2024;14(13):5507. [DOI]
10. Sarkar BB, Chaki N. High level net model for analyzing agent base distributed decision support system. In: *2009 International Association of Computer Science and Information Technology - Spring Conference*; 2009 Apr 17-20; Singapore. Piscataway: IEEE; 2009. p. 351-358. [DOI]
11. Jiang K, Jin G, Zhang Z, Cui R, Zhao Y. Incorporating external knowledge for text matching model. *Comput Speech Lang*. 2024;87:101638. [DOI]
12. Sun Y, Zhang Q, Bao J, Lu Y, Liu S. Empowering digital twins with large language models for global temporal feature learning. *J Manuf Syst*. 2024;74:83-99. [DOI]
13. Chen H, Liu Z, Sun M. The social opportunities and challenges in the era of large language models. *J Comput Res Dev*. 2024;61(5):1094-1103. Chinese. [DOI]
14. Yang T, Xu J, Nie X. Analysis of data exchange between BIM design software and performance analysis software based on revit. *J Phys Conf Ser*. 2022;2185(1):012070. [DOI]
15. Yakhou N, Thompson P, Siddiqui A, Abualdenien J, Ronchi E. The integration of building information modelling and fire evacuation models. *J Build Eng*. 2023;63:105557. [DOI]
16. Huo J, Liu J, Pei G, Wang T. Research on LOD lightweight method of railway four electric BIM model. In: *Proceedings of the 2022 6th International Conference on Electronic Information Technology and Computer Engineering*; 2022 Oct 21-23; Xiamen, China. The Netherlands: Atlantis Press; 2022. p. 492-497. [DOI]
17. buildingSMART International [Internet]. *Information Delivery Manual (IDM)*. Available from: <https://technical.buildingsmart.org/standards/information-delivery-manual/>
18. buildingSMART International [Internet]. *Model View Definitions (MVD)*. [cited 2026 Mar 11]. Available from: <https://technical.buildingsmart.org/standards/ifc/mvd/>
19. Ma Z, Wei Z, Zhang X. Semi-automatic and specification-compliant cost estimation for tendering of building projects based on IFC data of

- design model. *Autom Constr.* 2013;30:126-135. [DOI]
20. Choi J, Shin J, Kim M, Kim I. Development of openBIM-based energy analysis software to improve the interoperability of energy performance assessment. *Autom Constr.* 2016;72:52-64. [DOI]
 21. Kim H, Anderson K, Lee S, Hildreth J. Generating construction schedules through automatic data extraction using open BIM (building information modeling) technology. *Autom Constr.* 2013;35:285-295. [DOI]
 22. Won J, Lee G, Cho C. No-schema algorithm for extracting a partial model from an IFC instance model. *J Comput Civ Eng.* 2013;27(6):585-592. [DOI]
 23. Zhu M. Studies on partial model extraction and model merging for BIM based on file instances [dissertation]. Beijing: Tsinghua University; 2016. Chinese. Available from: <http://59.75.36.213/KCMS/detail/detail.aspx?filename=1017817927.nh&dbcode=CMFD&dbname=CMFD2018>
 24. Jia J, Ma H, Zhang Z. Integration of industry foundation classes and ontology: Data, applications, modes, challenges, and opportunities. *Buildings.* 2024;14(4):911. [DOI]
 25. Sobhkhiz S, Zhou YC, Lin JR, El-Diraby TE. Framing and evaluating the best practices of IFC-based automated rule checking: A case study. *Buildings.* 2021;11(10):456. [DOI]
 26. bimserver.org [Internet]. BIMserver.org – open source building information server. c2008-present. Available from: <https://bimserver.org/>
 27. Mazairac W, Beetz J. BIMQL–An open query language for building information models. *Adv Eng Inform.* 2013;27(4):444-456. [DOI]
 28. Cheng YS, Li QM, Cheng H. On implementing IFC data in relational database and its application. *Comput Appl Softw.* 2014;31(11):34-44. Chinese. Available from: https://kns.cnki.net/kcms2/article/abstract?v=BkbJkO_np9N-YmjV5RR8lyZxQ8HygILIZ4yuNkPtRycDfTMpIycFFZeNrtKaIhR2zRTqTqC_M22hQPCmcAXzWKSgLTvSAdIUyAiPIWbrSEDO9anbnsYjcN82is4YIGXoigGbFnrydehM7uDsrbh0HLdW2wt_I_EFEBimPX5SvuiiIkZ_WL_Py7CFV15J0FW&uniplatform=NZKPT&language=CHS%20%W%20CNKI
 29. Solihin W, Eastman C, Lee YC, Yang DH. A simplified relational database schema for transformation of BIM data into a query-efficient and spatially enabled database. *Autom Constr.* 2017;84:367-383. [DOI]
 30. Guo HL, Zhou Y, Ye XT, Luo ZB, Xue F. Automated mapping from an IFC data model to a relational database model. *J Tsinghua Univ Sci Technol.* 2021;61(2):152-160. Chinese. [DOI]
 31. Wyszomirski M. Analysis of the possibility of using key-value store NoSQL databases for IFC data processing in the BIM-GIS integration process. *Pol Cartogr Rev.* 2022;54(1):11-22. [DOI]
 32. Yang B, Dong M, Wang C, Liu B, Wang Z, Zhang B. IFC-based 4D construction management information model of prefabricated buildings and its application in graph database. *Appl Sci.* 2021;11(16):7270. [DOI]
 33. Gradišar L, Dolenc M. IFC and monitoring database system based on graph data models. *Adv Civ Eng.* 2021;2021:4913394. [DOI]
 34. Austern G, Bloch T, Abulafia Y. Incorporating context into BIM-derived data: Leveraging graph neural networks for building element classification. *Buildings.* 2024;14(2):527. [DOI]
 35. Chen Z, Chen L, Zhou X, Huang L, Sandanayake M, Yap PS. Recent technological advancements in BIM and LCA integration for sustainable construction: A review. *Sustainability.* 2024;16(3):1340. [DOI]
 36. Biagini C, Bongini A, Marzi L. From BIM to digital twin. IOT data integration in asset management platform. *J Inf Technol Constr.* 2024;29:1103-1127. [DOI]
 37. Yin M, Tang L, Webster C, Xu S, Li X, Ying H. An ontology-aided, natural language-based approach for multi-constraint BIM model querying. *J Build Eng.* 2023;76:107066. [DOI]
 38. Korkmaz Ö, Basaraner M. Ontology based integration of BIM and GIS for the representation of architectural, structural and functional elements of buildings. *Int Arch Photogramm Remote Sens Spatial Inf Sci.* 2024;48:49-55. [DOI]
 39. Pauwels P, Terkaj W. EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology. *Autom Constr.* 2016;63:100-133. [DOI]
 40. Wu J, Shen T, Huo L. Research on lightweighting methods for 3D building models based on semantic constraints. *Int Arch Photogramm Remote Sens Spatial Inf Sci.* 2025;48:1565-1571. [DOI]
 41. Huang PH, Song SY, Xu Z, Hu ZZ, Lin JR. Intelligent BIM searching via deep embedding of geometric, semantic, and topological features. *Buildings.* 2025;15(6):951. [DOI]
 42. Lin JR, Hu ZZ, Zhang JP, Yu FQ. A natural-language-based approach to intelligent data retrieval and representation for cloud BIM. *Computer Aided Civil Eng.* 2016;31(1):18-33. [DOI]
 43. Guo D, Onstein E, La Rosa AD. An approach of automatic SPARQL generation for BIM data extraction. *Appl Sci.* 2020;10(24):8794. [DOI]
 44. Brown T, Mann B, Ryder N, Subbiah M, Kaplan JD, Dhariwal P, et al. Language models are few-shot learners. *arXiv:2005.14165 [Preprint].* 2020. [DOI]
 45. Zheng J, Fischer M. Dynamic prompt-based virtual assistant framework for BIM information search. *Autom Constr.* 2023;155:105067. [DOI]
 46. Hellin S, Nousias S, Borrmann A. Natural Language Information Retrieval from BIM Models: An LLM-Based Agentic Workflow Approach. In: Petrova E, Srećković M, Meda P, Soman RK, Beetz J, McArthur J, Hall D, editors. *Proceedings of the 2025 European Conference on Computing in Construction & 42nd International CIB W78 Conference on Information Technology in Construction; 2025 Jul 14-17; Porto, Portugal. Portugal: European Council on Computing in Construction (EC3); 2025. Available from: https://mediatum.ub.tum.de/doc/1781947/66amsnnaqbygipuftj8b88oqv.2025_HELLIN_EC3.pdf*

47. Perov I. AI assisted BIM query system [dissertation]. Finland: Tampere University; 2025. Available from: <https://trepo.tuni.fi/bitstream/handle/10024/229025/PerovIvan.pdf?sequence=2>
48. Du C, Esser S, Nousias S, Borrmann A. Text2BIM: Generating building models using a large language model-based multiagent framework. J Comput Civ Eng. 2026;40(2):04025142. [DOI]
49. Liu P, Yuan W, Fu J, Jiang Z, Hayashi H, Neubig G. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. ACM Comput Surv. 2023;55(9):1-35. [DOI]

Publisher's Note

Science Exploration remains a neutral stance on jurisdictional claims in published maps and institutional affiliations. The views expressed in this article are solely those of the author(s) and do not reflect the opinions of the Editors or the publisher.