A Graph-RAG Framework for Natural Language Interaction with Dynamic Construction Project Graphs

Mohammad Saeed Heidary[1], Xuesong Shen[1], Milad Mousavi[1], Khalegh Barati[1], and James Linke[2]

[1] School of Civil and Environmental Engineering, the University of New South Wales, Sydney, NSW 2052, Australia
[2] GeoAI, Sydney, NSW 2019, Australia.
m.heidary@unsw.edu.au, x.shen@unsw.edu.au, milad.mousavi@unsw.edu.au, khalegh.barati@unsw.edu.au, j.linke@llsurveys.com.au

Abstract –

Construction projects generate heterogeneous data from schedules, design models, and site observations. Graph-based data models have recently been adopted to integrate these diverse data sources, but interacting with such models typically requires specialized technical knowledge, which limits their usability in construction practice. This study proposes a schema-aware Graph Retrieval-Augmented Generation (Graph-RAG) framework that enables natural language interaction with construction project graph data models. The framework integrates a large language model with a Neo4j graph database to interpret user queries, retrieve project data through Cypher queries, and generate evidence-grounded responses. A case study integrating schedule data, an IFC design model, and computer-vision-derived site observations produced a project graph with 218 nodes and 377 relationships. Evaluation using six project queries showed that the proposed framework achieved 100% accuracy, outperforming a Text-to-Cypher approach (33.3% accuracy), while direct LLM interaction failed to answer any questions (0% accuracy). These results demonstrate the effectiveness of schema-aware Graph-RAG for reliable natural language access to construction project data.

Keywords –

Construction Projects; Knowledge Graph; Graph-RAG; LLMs; Construction Data Interaction; Natural Language Interaction

1 Introduction

Construction projects generate large volumes of heterogeneous data throughout their lifecycle, including schedules, design models, site observations, and as-built records [1, 2]. Integrating these diverse data sources in a meaningful and consistent manner remains a significant challenge for project management and decision-making [3]. In recent years, graph-based data models and knowledge graphs have emerged as a promising solution for construction data integration [4]. By representing project entities such as activities, infrastructure elements, resources, and observations as interconnected nodes and relationships, graph models enable heterogeneous data sources to be linked in a semantically meaningful way, supporting richer representation of project knowledge and improved data integration across different project systems [5, 6].

Despite their strong capability for integrating heterogeneous construction data, interacting with graph-based data models often requires specialized technical knowledge. Accessing and querying graph databases typically require graph query languages and database technologies that are more familiar to computer scientists than to construction practitioners [7]. As a result, many project stakeholders may find it difficult to directly interact with graph-based project data models, limiting their usability in everyday construction management. This creates a gap between the availability of integrated project data and the ability of practitioners to effectively access and interpret that information.

Recent advances in large language models (LLMs) offer a promising opportunity to bridge this gap by enabling natural language interaction with complex data systems [8]. Because graph data models organize project information through semantically connected entities and relationships [9], LLMs can potentially serve as an interface that translates user questions into meaningful interactions with the project graph. Motivated by this potential, this study proposes a framework that leverages LLMs to enable natural language interaction with construction project graph data models. By allowing users to query project information using natural language while grounding responses in graph-based data, the proposed approach aims to improve the usability of graph-based construction information systems and make integrated project knowledge more accessible to construction stakeholders.

2 Related Work

This section reviews prior research on construction information modelling, LLMs, and retrieval-augmented generation, highlighting current practices and limitations in construction-phase applications.

2.1 Construction Information Models and Graph-based Representations

Construction projects have traditionally relied on Building Information Modelling (BIM) platforms and relational databases to manage project information [10]. BIM has been widely adopted to support design coordination, quantity take-off, and visualization[11], while relational databases have been used to store schedules, cost records, and inspection data [12, 13]. For example, Heidary et al. [14] used a 4D BIM environment to link 3D design models with construction schedules, enabling improved visualization of planned activities over time and supporting hazard identification and safety management during construction, whereas Palomar et al. [15] implemented a relational database–based platform using SQL Server to centrally store and synchronize structured project data across multiple disciplines. Although these approaches have improved information organization, they typically operate in isolated silos and rely on predefined schemas that limit flexibility when project conditions change [3]. As construction progresses and new data sources emerge, such as site observations and as-built records, maintaining consistency and traceability across systems remains challenging [1, 2].

To address these limitations, recent research has explored graph-based data models and knowledge graphs for construction information management [4]. Graph representations allow entities such as activities, resources, building elements, and observations to be explicitly linked, supporting richer semantics and more flexible querying than traditional tabular models [5]. In this context, Sun et al. [16] reviewed text-based automatic knowledge graph construction for road infrastructure operations, demonstrating how document-centric sources can be transformed into structured knowledge graphs to support querying and reasoning across lifecycle activities. Knowledge graphs have been applied to integrate BIM data, schedules, safety information, and regulatory knowledge, enabling reasoning over relationships rather than isolated attributes [6, 17]. These approaches have demonstrated advantages in representing complex dependencies and enabling inference across heterogeneous data sources [18, 19].

Despite these advances, interaction with graph-based construction data models typically relies on predefined dashboards or fixed query templates [7]. Such interfaces are designed around anticipated information needs and require backend development when new questions arise [9]. As a result, responding to emerging or unexpected information demands during the construction phase remains difficult, limiting the practical usability of graph-based systems for day-to-day decision-making by project teams.

2.2 Large Language Models in Construction Projects

LLMs are generative AI systems trained to predict and produce natural language, enabling tasks such as question answering, summarization, and drafting based on learned linguistic patterns [8]. Recent evaluations show that ChatGPT-style models can perform competitively on domain problem-solving tasks, motivating their exploration as decision-support tools in construction management [20]. In the construction context, LLMs are increasingly positioned as conversational interfaces that can translate complex technical content into more accessible, actionable narratives for practitioners [21].

In construction applications, Prieto et al. [22] investigated ChatGPT for construction scheduling, examining how well it can generate schedule-related outputs and support planners through natural-language interaction. For project documentation, Xiao et al. [23] combined computer vision with ChatGPT to convert construction videos into automated daily reports, aiming to reduce manual reporting effort while preserving site progress details. For compliance, Yang and Zhang [24] proposed a prompt-based pipeline to transform building code text into structured representations usable for automated checking.

However, the direct use of LLMs also introduces important limitations. LLMs generate responses based on statistical patterns learned from training data rather than verified, project-specific information [24], which can lead to hallucinations, incorrect assumptions, or missing contextual constraints [25]. In construction projects, where decisions are often time-sensitive and safety-critical, such behavior poses significant risks. These limitations highlight the need for complementary mechanisms that can ground LLM outputs in authoritative, traceable, and project-specific information sources.

2.3 Retrieval-augmented Generation in Construction

Retrieval-augmented generation (RAG) has recently been adopted in construction research to improve the reliability of large language model (LLM) outputs by grounding responses in external knowledge sources. Lee et al. [26] compared a RAG-based approach with a fine-tuned LLM for construction safety knowledge retrieval by integrating GPT-4 with a safety knowledge graph derived from guidelines, demonstrating improved consistency in answering safety queries. Similarly, Baek et al. [27] developed a RAG-enhanced framework that retrieves activity- and equipment-specific cases from a construction accident repository to generate tailored safety risk management guidance, showing that domain-adapted embeddings improve retrieval quality. Beyond safety applications, Zheng et al. [28] integrated LLMs with a nested contract knowledge graph to improve the accuracy and interpretability of automated contract risk identification.

Recent studies have further extended RAG and LLM-grounded approaches to different construction tasks and data modalities. Guo et al. [29] proposed a multimodal framework combining a visual language model with a two-stage RAG pipeline to support automated safety compliance monitoring from site images. Zhou et al. [30] integrated an external construction engineering management knowledge base with general-purpose LLMs to enable question answering in construction management contexts when domain training data are limited. Related work has explored alternative grounding strategies, such as long-context LLMs for building code interpretation [31], RAG-based systems for generating personalized feedback in construction management training [32], and multimodal multi-agent frameworks for geotechnical design tasks involving both text and images [33]. In addition, recent work by Pan et al. [34] proposed an LLM-enabled multi-agent framework that enables natural language interaction with graph-based digital twins of existing buildings, where an agent pipeline interprets user queries, generates graph database commands, and retrieves information from the digital twin graph to produce grounded responses for facility management scenarios.

Nevertheless, existing applications of RAG in construction largely focus on static knowledge sources, such as codes, regulations, historical documents, or predefined training materials. The use of RAG to interact with live, evolving project data during the construction phase has received limited attention. In particular, the potential of Graph-RAG to enable natural language access to project knowledge graphs that represent the current state of construction activities and site conditions remains underexplored. Addressing this gap requires frameworks that combine graph-based project data with retrieval-grounded language models to support reliable natural language interaction with construction-phase project information.

3 Research Methodology

This study proposes a Graph Retrieval-Augmented Generation (Graph-RAG) framework to enable natural language interaction with a construction project graph data model. The system integrates a large language model (LLM) with a graph database so that users can query project information using natural language while responses remain grounded in structured graph data. When a user submits a question, the framework first retrieves the schema and ontology from the project graph to provide structural context. Using this information, the LLM decomposes the user query into simpler sub-questions aligned with the graph structure. For each sub-question, the LLM generates a Cypher query to retrieve relevant information from the graph database. The retrieved graph data are then used to generate grounded answers for each sub-question, which are finally synthesized by the LLM to produce a unified response to the original user query. The overall workflow of the proposed Graph-RAG framework is illustrated in Figure 1.

The LLM in the proposed framework performs three key tasks: (1) question decomposition, (2) graph query generation for retrieval, and (3) response generation and synthesis. These components enable the system to bridge natural language user queries and structured project graph data, allowing complex project information to be accessed through intuitive natural language interaction.

3.1 Schema-aware Question Decomposition

When a user submits a natural language query, the system first retrieves the schema and ontology information from the project graph database. This information describes the available node types, relationships, properties, and semantic structures within the project data model. The schema is provided as contextual input to the LLM to ensure that the query interpretation is consistent with the structure of the graph.

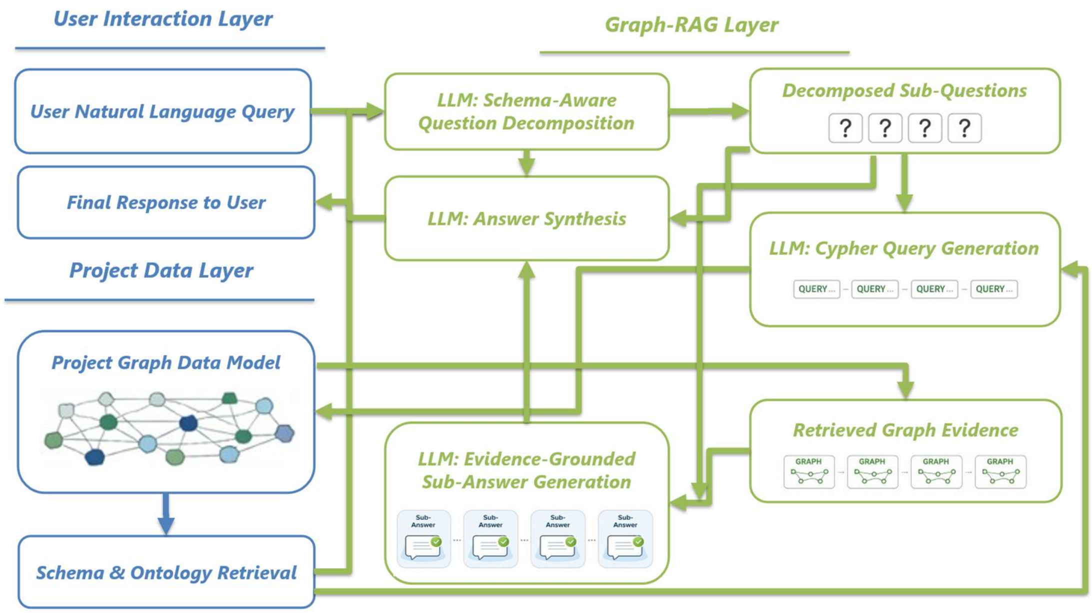

Figure 1. Overview of the proposed Graph-RAG framework for natural language interaction with construction project graph data.

Using this schema-aware context, the LLM refines the user query and decomposes it into a set of simpler sub-questions. Each sub-question represents a specific information retrieval task that can be mapped to graph entities and relationships. This decomposition step helps transform complex user queries into structured information retrieval tasks that align with the project graph schema.

3.2 Graph Query Generation and Retrieval

For each decomposed sub-question, the LLM generates a Cypher query that retrieves the relevant information from the graph database. The generated queries follow the structure of the project graph schema and are designed to retrieve entities, relationships, and contextual attributes required to answer the sub-question.

The Cypher queries are executed against the Neo4j graph database, and the retrieved results are returned as structured graph evidence. This retrieval step ensures that the responses generated by the LLM are grounded in actual project data rather than relying solely on language model inference.

3.3 Evidence-Grounded Response Generation and Synthesis

Once the relevant graph data are retrieved, the LLM generates a response for each sub-question using the retrieved evidence. The model is instructed to base its reasoning only on the retrieved graph information to ensure factual consistency and transparency.

Finally, the LLM synthesizes the responses to the individual sub-questions together with the original user query to generate a unified answer. This synthesis step enables the system to provide coherent explanations that integrate multiple pieces of graph-retrieved information into a final response for the user.

4 Case Study

To evaluate the proposed framework in a real-world scenario, a case study was conducted on an ongoing road intersection upgrade project in New South Wales, Australia. Construction of the project commenced in late 2023 and is scheduled for completion in 2027. The project involves multiple construction activities, resources, and observational data collected during the construction phase. These heterogeneous data sources provide a suitable environment for examining how a graph-based data model combined with large language models can support natural language interaction with project information.

4.1 Case Study Project Graph Data Model Preparation

In this case study, three types of project data were integrated to construct the project graph data model: the project schedule (structured data), an IFC design model (semi-structured data), and construction site images (unstructured data). These heterogeneous data sources were mapped into a graph structure using ten top-level entities representing key concepts of the construction domain. Through predefined semantic rules, information from the schedule, design model, and site observations was linked to form a unified project graph.

Construction site images were processed using the YOLOv8 computer vision model to detect relevant construction elements. The detected elements were then matched with corresponding elements in the IFC design model based on spatial location and linked to the related construction activities defined in the project schedule. By connecting activities, design elements, and observed site conditions, the different data sources were semantically integrated into a single project graph data model. After processing one project schedule, one IFC model, and 32 construction site images (two images per observation date across 16 different dates), the resulting graph contained 218 nodes and 377 relationships representing the project state (Figure 2).

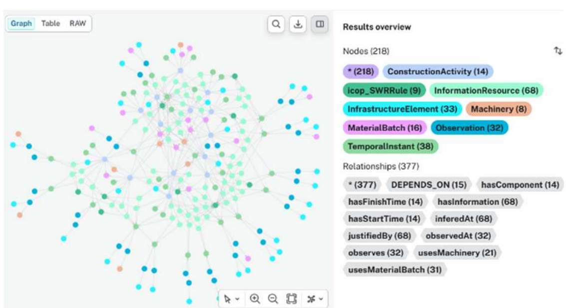

Figure 2. Snapshot of the dynamic construction project graph in Neo4j on 5 September 2025.

4.2 Developing the Graph-RAG Pipeline

The Graph-RAG pipeline was developed and implemented in Python by integrating the Neo4j graph database with the OpenAI GPT-4.1 model through the OpenAI API. The developed pipeline first retrieves the schema of the project graph and converts it into a structured text summary that can be provided to the LLM as contextual input. Based on this schema-aware context, the framework processes the user question through several LLM-driven stages. First, the LLM decomposes the user question into simpler sub-questions while considering the schema and ontology of the project graph. Second, for each sub-question, the LLM generates a Cypher query to retrieve relevant data from the Neo4j database. Finally, the retrieved graph evidence is used by the LLM to generate grounded responses for each sub-question, and these are then synthesized into the final answer returned to the user. Prompt engineering was used throughout these stages to control the LLM behavior and keep the outputs aligned with the graph structure and project semantics.

In the implemented prompts, each stage was guided by a specific instruction logic. For question decomposition, the LLM was instructed to act as an ontology-aware planning assistant that refines the user question conservatively, decides whether decomposition is needed, and creates schema-consistent sub-questions without introducing unsupported graph elements. For Cypher generation, the LLM was instructed to produce read-only Cypher queries, follow project-specific semantic rules, use only valid retrieval patterns, and avoid any write or administrative operations. For evidence-grounded answering, the LLM was instructed to answer each sub-question only from the retrieved graph evidence, explicitly mention uncertainty when evidence is incomplete, and avoid inventing unsupported facts. In the final synthesis stage, the LLM was instructed to answer the original user question directly by combining the sub-question answers while clearly preserving the distinction between graph-supported findings and uncertain information.

4.3 Testing Setup

To evaluate the effectiveness of the proposed framework, six questions were designed to test the system's ability to retrieve project information under different reasoning scenarios. These questions include temporal status queries, progress tracking queries, resource matching queries, and delay detection queries.

- Q1 — What was the construction status of the activity "8 Columns A – FW" on 2 June 2025?
- Q2 — What was the construction status of the activity "8 Columns A – FW" on 21 May 2025?
- Q3 — What was the construction status of the activity "8 Piles A" on 23 April 2025?
- Q4 — What was the construction status of the activity "8 Piles A" on 7 May 2025?
- Q5 — Which observed machinery types match the machinery planned for the associated construction activities?
- Q6 — Based on the latest available observations, which construction activities are currently identified as delayed?

Four different question-answering methods were evaluated:

- Direct LLM – only the user question is given to the LLM.
- LLM + Full Graph Data – the LLM receives the question and the entire project graph data model as input.
- Text-to-Cypher Retrieval – the LLM generates a Cypher query directly from the question and retrieves graph data in a single step.
- Proposed Graph-RAG Method – the schema-aware decomposition and retrieval framework proposed in this study.

A Python-based testing interface was developed to allow a user to submit a question and automatically generate responses from all four evaluation methods simultaneously for comparison, as illustrated in Figure 3.

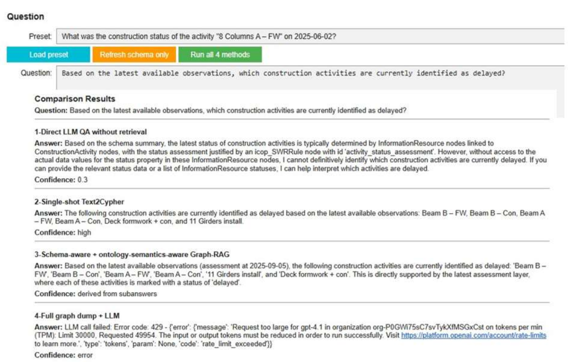

Figure 3. Screenshot of the Python testing environment showing simultaneous outputs from the four evaluation methods.

4.4 Results

The experimental results demonstrate clear differences between the evaluated methods. When only the question was provided to the LLM, the model was unable to produce correct answers for any of the questions because it had no knowledge of the specific project data. This outcome is expected since general-purpose LLMs are trained on broad datasets and do not contain information about a particular construction project.

When the entire project graph data model (218 nodes and 377 relationships) was provided as input to the LLM, the API request exceeded the tokens-per-minute (TPM) rate limit of GPT-4.1, producing the following error:

Error: request exceeds the allowed token limit (429 rate limit error).

This indicates that processing the entire project graph as input exceeded the TPM rate limit of the GPT-4.1 API, making this approach impractical for real-world applications due to computational and processing constraints.

The remaining two approaches were able to produce responses. The one-step Text-to-Cypher method correctly answered only the last two questions (2/6), while it failed to correctly interpret the temporal context required for the first four questions. In contrast, the proposed Graph-RAG framework successfully answered all six questions while also providing evidence retrieved from the project graph. A summary of the performance of the four evaluated methods is presented in Table 1, demonstrating the improved accuracy and reliability of the proposed Graph-RAG framework in retrieving project-specific information.

Table 1. Accuracy comparison of the evaluated question-answering methods.

Method	Accuracy
Direct LLM	0/6 (0 %)
LLM + Full Graph Data	Not executable under standard API rate limits
Text-to-Cypher	2/6 (33.3 %)
Proposed Graph-RAG	6/6 (100 %)

5 Discussion and Conclusions

This study proposed a Graph Retrieval-Augmented Generation (Graph-RAG) framework to enable natural language interaction with construction project graph data models. By integrating a large language model with a graph database, the framework allows users to query project information in natural language while ensuring that responses are grounded in structured project data. The case study shows that the approach can effectively retrieve project-specific information from heterogeneous data sources integrated into a unified project graph. Compared with alternative approaches such as direct LLM interaction and one-step Text-to-Cypher retrieval, the proposed framework achieved higher accuracy and produced responses supported by explicit graph evidence.

Nevertheless, this study has some limitations. The evaluation was conducted on a single case study with a relatively small project graph, which may not fully represent the scale and complexity of larger construction projects. Future work will extend the framework to support multi-query conversations, causal and predictive reasoning, and broader evaluation across different project types. Further research will also examine the performance, scalability, and user adoption of the proposed framework in larger construction project environments.

Acknowledgments

This research is supported by funding from the Australian Research Council (ARC) Research Hub for Resilient and Intelligent Infrastructure Systems (IH210100048), and the ARC Research Hub for Human-Robot Teaming for Sustainable and Resilient Construction (IH240100016). We also extend our sincere appreciation to GeoAI, in particular Mr. Alastair Linke, Chief Executive Officer, Dr. Youheng Guo, and Mr. Barber Cameron, for providing access to the case study data and for offering valuable technical feedback throughout the project.

References

[1] W. Teclaw, J. O'Donnel, V. Kukkonen, P. Pauwels, N. Labonnote, and E. Hjelseth, "Federating cross-domain BIM-based knowledge graph," *Advanced Engineering Informatics,* vol. 62, 2024, doi: 10.1016/j.aei.2024.102770.

[2] C. Du, Z. Deng, S. Nousias, and A. Borrmann, "Predictive modeling: BIM command recommendation based on large-scale usage logs," *Advanced Engineering Informatics,* vol. 68, 2025, doi: 10.1016/j.aei.2025.103574.

[3] D. F. Bucher, J. J. Hunhevicz, R. K. Soman, P. Pauwels, and D. M. Hall, "From BIM to Web3: A critical interpretive synthesis of present and emerging data management approaches in construction informatics," *Advanced Engineering Informatics,* vol. 62, 2024, doi: 10.1016/j.aei.2024.102884.

[4] M. H. Rasmussen, M. Lefrançois, P. Pauwels, C. A. Hviid, and J. Karlshøj, "Managing interrelated project information in AEC Knowledge Graphs," *Automation in Construction,* vol. 108, 2019, doi: 10.1016/j.autcon.2019.102956.

[5] X. Wang and N. El-Gohary, "Deep learning-based relation extraction and knowledge graph-based representation of construction safety requirements," *Automation in Construction,* vol. 147, 2023, doi: 10.1016/j.autcon.2022.104696.

[6] H. Wang, S. Xu, D. Cui, H. Xu, and H. Luo, "Information Integration of Regulation Texts and Tables for Automated Construction Safety Knowledge Mapping," *Journal of Construction Engineering and Management,* vol. 150, no. 5, 2024, doi: 10.1061/jcemd4.Coeng-14436.

[7] R. Sacks, Z. Wang, B. Ouyang, D. Utkucu, and S. Chen, "Toward artificially intelligent cloud-based building information modelling for collaborative multidisciplinary design," *Advanced Engineering Informatics,* vol. 53, 2022, doi: 10.1016/j.aei.2022.101711.

[8] M. B. Jelodar, "Generative AI, Large Language Models, and ChatGPT in Construction Education, Training, and Practice," *Buildings,* vol. 15, no. 6, 2025, doi: 10.3390/buildings15060933.

[9] J. Schlenger, K. Pluta, A. Mathew, T. Yeung, R. Sacks, and A. Borrmann, "Reference architecture and ontology framework for digital twin construction," *Automation in Construction,* vol. 174, 2025, doi: 10.1016/j.autcon.2025.106111.

[10] L. Soibelman and H. Kim, "Data Preparation Process for Construction Knowledge Generation through Knowledge Discovery in Databases," *Journal of Computing in Civil Engineering,* vol. 16, no. 1, pp. 39-48, 2002, doi: 10.1061/(asce)0887-3801(2002)16:1(39).

[11] M. Mousavi, A. TohidiFar, and A. Alvanchi, "BIM and machine learning in seismic damage prediction for non-structural exterior infill walls," *Automation in Construction,* vol. 139, 2022, doi: 10.1016/j.autcon.2022.104288.

[12] T. Hegazy and T. Ersahin, "Simplified Spreadsheet Solutions. II: Overall Schedule Optimization," *Journal of Construction Engineering and Management,* vol. 127, no. 6, pp. 469-475, 2001, doi: 10.1061/(asce)0733-9364(2001)127:6(469).

[13] T. Valenko and U. Klanšek, "An integration of spreadsheet and project management software for cost optimal time scheduling in construction," *Organization, Technology and Management in Construction: an International Journal,* vol. 9, no. 1, pp. 1627-1637, 2017, doi: 10.1515/otmcj-2016-0028.

[14] M. S. Heidary, M. Mousavi, A. Alvanchi, K. Barati, and H. Karimi, "Semi-automatic Construction Hazard Identification Method Using 4D BIM," presented at the Proceedings of the 38th International Symposium on Automation and Robotics in Construction (ISARC), 2021, doi: 10.22260/ISARC2021/0080

[15] I. J. Palomar, J. L. García Valldecabres, P. Tzortzopoulos, and E. Pellicer, "An online platform to unify and synchronise heritage architecture information," *Automation in*

Construction, vol. 110, 2020, doi: 10.1016/j.autcon.2019.103008.

[16] Y. Sun, X. Shen, S. Zlatanova, K. Barati, and J. Linke, "Text-based automatic knowledge graph construction for road infrastructure operations management," *Automation in Construction,* vol. 182, 2026, doi: 10.1016/j.autcon.2025.106733.

[17] R. Zhang and N. El-Gohary, "Transformer-based approach for automated context-aware IFC-regulation semantic information alignment," *Automation in Construction,* vol. 145, 2023, doi: 10.1016/j.autcon.2022.104540.

[18] S. Kosse, V. Betker, P. Hagedorn, M. König, and T. Schmidt, "A Semantic Digital Twin for the Dynamic Scheduling of Industry 4.0-based Production of Precast Concrete Elements," *Advanced Engineering Informatics,* vol. 62, 2024, doi: 10.1016/j.aei.2024.102677.

[19] J. Akroyd, S. Mosbach, A. Bhave, and M. Kraft, "Universal Digital Twin - A Dynamic Knowledge Graph," *Data-Centric Engineering,* vol. 2, 2021, doi: 10.1017/dce.2021.10.

[20] R. Nyqvist, A. Peltokorpi, and O. Seppänen, "Can ChatGPT exceed humans in construction project risk management?," *Engineering, Construction and Architectural Management,* vol. 31, no. 13, pp. 223-243, 2024, doi: 10.1108/ecam-08-2023-0819.

[21] L. Zhang and Q. Ge, "Offsite construction risk management agent – a GPTs based analysis model," *Engineering, Construction and Architectural Management,* 2025, doi: 10.1108/ecam-05-2024-0624.

[22] S. A. Prieto, E. T. Mengiste, and B. García de Soto, "Investigating the Use of ChatGPT for the Scheduling of Construction Projects," *Buildings,* vol. 13, no. 4, 2023, doi: 10.3390/buildings13040857.

[23] B. Xiao, Y. Wang, Y. Zhang, C. Chen, and A. Darko, "Automated daily report generation from construction videos using ChatGPT and computer vision," *Automation in Construction,* vol. 168, 2024, doi: 10.1016/j.autcon.2024.105874.

[24] F. Yang and J. Zhang, "Prompt-based automation of building code information transformation for compliance checking," *Automation in Construction,* vol. 168, 2024, doi: 10.1016/j.autcon.2024.105817.

[25] M. Oral, Ö. Alboga, S. Aydınlı, and E. Erdis, "Usability of large language models for building construction safety risk assessment," *Engineering, Construction and Architectural Management,* 2025, doi: 10.1108/ecam-08-2024-1143.

[26] J. Lee, S. Ahn, D. Kim, and D. Kim, "Performance comparison of retrieval-augmented generation and fine-tuned large language models for construction safety management knowledge retrieval," *Automation in Construction,* vol. 168, 2024, doi: 10.1016/j.autcon.2024.105846.

[27] S. Baek, C. Y. Park, and W. Jung, "Automated safety risk management guidance enhanced by retrieval-augmented large language model," *Automation in Construction,* vol. 176, 2025, doi: 10.1016/j.autcon.2025.106255.

[28] C. Zheng, S. Wong, X. Su, Y. Tang, A. Nawaz, and M. Kassem, "Automating construction contract review using knowledge graph-enhanced large language models," *Automation in Construction,* vol. 175, 2025, doi: 10.1016/j.autcon.2025.106179.

[29] K. X. Guo, P. K.-Y. Wong, J. C. P. Cheng, C.-F. Chan, P.-H. Leung, and X. Tao, "Enhancing visual-LLM for construction site safety compliance via prompt engineering and Bi-stage retrieval-augmented generation," *Automation in Construction,* vol. 179, 2025, doi: 10.1016/j.autcon.2025.106490.

[30] S. Zhou, X. Liu, D. Li, T. Gu, K. Liu, Y. Yang, and M. O. Wong, "Integrating domain-specific knowledge and fine-tuned general-purpose large language models for question-answering in construction engineering management," *Automation in Construction,* vol. 175, 2025, doi: 10.1016/j.autcon.2025.106206.

[31] J. Lee and G. Lee, "Long context window-based zero-shot legal interpretation of building codes and regulations," *Automation in Construction,* vol. 179, 2025, doi: 10.1016/j.autcon.2025.106450.

[32] X. Hu, Y. M. Goh, and J. Tay, "Mixed-methods evaluation of automated personalised feedback in construction management training using RAG and LLMs," *Automation in Construction,* vol. 182, 2026, doi: 10.1016/j.autcon.2025.106745.

[33] H.-R. Xu, N. Zhang, Z.-Y. Yin, and P. G. A. Njock, "Multimodal framework integrating multiple large language model agents for intelligent geotechnical design," *Automation in Construction,* vol. 176, 2025, doi: 10.1016/j.autcon.2025.106257.

[34] Y. Pan, M. Wang, L. Lu, R. Lamsal, E. Pärn, S. Zlatanova, and I. Brilakis, "LLM-enabled multi-agent framework for natural language interaction with graph-based digital twins," *Automation in Construction,* vol. 183, 2026, doi: 10.1016/j.autcon.2026.106791.