2025 European Conference on Computing in Construction

CIB W78 Conference on IT in Construction

Porto, Portugal
July 14-17, 2025

GPT-POWERED MULTI-AGENT SYSTEM FOR FACILITY MANAGEMENT: KNOWLEDGE GRAPH-BASED DATA INTEGRATION

Amirhossein Babaei Ravandi and Tamer E. El-Diraby
Centre for Intelligent Buildings Digital Twinning, Dept. of Civil & Mineral Engineering,
University of Toronto, Toronto, Ontario M5S 1A4, Canada

Abstract

Facility management involves processing structured sensor data and unstructured maintenance records to monitor building performance and support decision-making. This study introduces a graph-based, multi-agent system that integrates a structured data storage, a knowledge graph, and a GPT-powered LLM execution framework. The system's web-based interface enables users to submit natural language queries and obtain actionable insights. A case study on UofT's Galbraith Building demonstrates how the system links historical maintenance records with sensor data. The results show the system's potential for predictive and prescriptive maintenance, reducing manual data processing efforts, decreasing reliance on technical expertise, and improving operational efficiency.

Introduction and Background

Facility management relies on extensive data collection to monitor building performance, optimize operations, and oversee maintenance activities (Atkin and Brooks, 2021; Shamshiri et al., 2024). A comprehensive management of its data, therefore, involves analyzing heterogeneous data formats from structured information (such as sensor readings) to unstructured records (such as work orders). For example, work orders (also known as service orders) provide direct insights into the experiences of building occupants and facility managers. They capture details about reported issues, assigned personnel, repair costs, and resolution timelines (Sobhkhiz and El-Diraby, 2023; Sobhkhiz and El-Diraby, 2023). However, their unstructured nature makes it difficult to extract actionable insights, which leads to inefficiencies in decision-making, resource allocation, and maintenance scheduling (Y. Li et al., 2024; Mo et al., 2020). Similarly, while sensor readings involve continuous environmental and operational data, they often remain separate from maintenance records. This isolation can make it challenging to correlate equipment performance with historical issues and service interventions (Dixit et al., 2019; Valinejadshoubi et al., 2022).

Traditional facility management approaches struggle with fragmented data storage, manual information retrieval, and limited analytical capabilities (Talamo and Bonanomi, 2015). In these approaches, on the one hand, sensor readings are typically stored in relational databases. On the other hand, work orders and service logs remain largely text-based and disconnected from other operational data sources. As a result, without an integrated system that links these data types, facility managers face challenges in identifying recurring issues, predicting equipment failures, and optimizing maintenance workflows (Ahmed et al., 2017; L. Li et al., 2024). In other words, the lack of automated insight generation forces decision-makers to rely on manual reviews and experience-driven judgments, which can lead to delayed responses and increased costs (Mannino et al., 2021).

The recent developments in knowledge graphs, multi-agent execution models, and large language models (LLMs) offer new opportunities that can be used to improve data integration, retrieval, and analysis. For example, knowledge graphs provide a structured way to represent relationships between facility components, maintenance history, and environmental conditions. This sets the stage for relationship-based queries, which are efficient in identifying hidden patterns (Peng et al., 2023). LLMs (such as GPT) bring a unique capability to facility management by serving as interpreters between human queries and structured data. For instance, GPT can dynamically translate diverse (and often ambiguous) natural language inputs into valid, structured queries that can be used in both relational databases and knowledge graphs (B. Li et al., 2024; Xi et al., 2025). Finally, multi-agent execution frameworks further enhance the system's ability to break down complex queries into specialized tasks (Singh et al., 2024) and allow different components (agents) to handle them in a more effective manner.

This study presents an integrated graph-based, multi-agent system that connects structured and unstructured facility data. One of its primary objectives is to allow facility managers to interact with sensor readings, work orders, and analytical tools in natural language. The system utilizes a relational database to store time-series sensor data, a knowledge graph to represent relationships between facility entities, and an LLM-powered query execution framework to automate information retrieval and insight generation.

In order to do so, a web-based interface has been developed for the Facilities and Services (F&S) Department ("Facilities & Services - University of Toronto," 2021) at the University of Toronto (UofT). This accessible platform assists users in submitting queries and retrieving meaningful insights. The system supports data-driven decision-making and establishes a foundation for predictive and prescriptive maintenance strategies (Cheng et al., 2020; Jeon et al., 2024) and network analysis (Babaei Ravandi and El-Diraby, 2024). A case study on the Galbraith Building at UofT is performed. The results demonstrate the proposed approach's ability to automate data retrieval, correlate diverse facility datasets, and generate operational intelligence for facility operations.

The remainder of this paper discusses the methodology, including the system architecture, data integration approach, and execution framework. Also, the case study section showcases the system's application in analyzing real-world facility management data. This is followed by a discussion of key findings, limitations, and potential future directions. The paper concludes by summarizing its contributions and reflecting on its implications for facility management and data-driven decision-making.

Proposed Framework

This study presents a framework that integrates a structured data storage system and a graph-based model to support facility management analysis. High-frequency numerical records (such as sensor measurements) are stored in the structured data storage system. The knowledge graph database, on the other hand, captures relationships between facility components (e.g., physical spaces, equipment, sensors, and maintenance activities). In addition to structured data, the system incorporates unstructured textual records, work orders in this study, which document reported issues and maintenance actions. These records are converted to a structure suitable for knowledge graphs, which eventually enables their integration with (structured) numerical data. This dual-layered design supports both time-dependent trend analysis and relationship-based reasoning and sets the stage for a deeper understanding of facility operations.

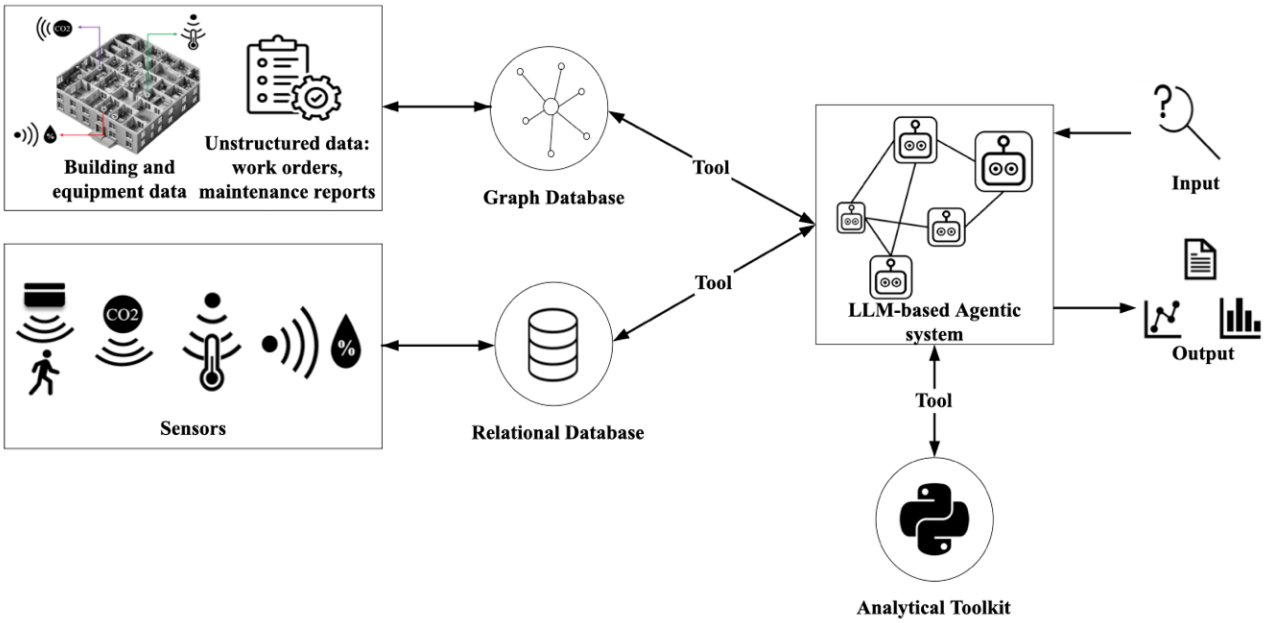

Figure 1: Data Integration and Processing Pipeline

In the proposed framework, sensor metadata—including type, location, and installation details—is represented in the knowledge graph. However, the sensor readings are stored in the structured database. The objective here is to efficiently handle high-frequency data. To connect the sensors to their readings, each sensor node in the graph contains references to its corresponding time-series data. This can also improve the contextual integration between historical trends and facility relationships.

In addition, work orders, building components, and sensor (meta) data are linked through graph relationships. These relationships allow the system to establish connections between maintenance actions and operational conditions. They also facilitate querying complex interdependencies, such as identifying how temperature fluctuations correlate with HVAC work orders. To process user queries and extract insights, the system utilizes an agentic LLM-based framework enhanced with prompt engineering. The system is designed as a multi-agent execution model. The agents handle specific aspects of data processing. Each one is responsible for unique tasks such as retrieving structured data, reasoning over the graph, or performing analytical operations. LLM-based agents autonomously interpret user queries, determine whether the request requires numerical data, relational insights, or both, and coordinate the execution process accordingly. The framework dynamically selects appropriate retrieval and processing tools to ensure that the system integrates

sensor data, maintenance records, and facility components into a unified analytical process.

To support information retrieval and analysis, the system includes specialized execution tools. These tools enable the system's interaction with structured databases, knowledge graphs, and analytical functions. For example, the relational query execution tool retrieves numerical records from the structured database and provides access to high-frequency sensor measurements. The graph execution tool processes graph-based queries and extracts relational insights from the knowledge graph. It also supports analyses that examine dependencies among facility components, historical maintenance records, and equipment interactions. The agentic system applies relationship-based reasoning to identify patterns, detect structural dependencies, and assess the impact of operational elements on facility performance. The analytics execution tool applies statistical models, anomaly detection algorithms, and visualization techniques to process retrieved data and refine the interpretation of results. These tools function within a structured execution pipeline, where the agentic system determines the appropriate tool(s) based on query intent. This approach allows the system to generate responses that go beyond direct data extraction and supports comparative analysis, trend identification, and multi-source correlation. Figure 1 illustrates the data integration and processing pipeline.

Case Study

This case study demonstrates the application of the proposed framework in analyzing facility management data for the Galbraith Building (GB), located at the St. George campus of UofT. Constructed in 1960 and renovated in 1991, the building consists of six floors, including a basement level, and houses academic and research spaces, such as classrooms, offices, and laboratories. Given its diverse occupancy and operational demands, facility management requires continuous monitoring of environmental conditions and equipment performance to ensure optimal functionality and occupant comfort. Figure 2 presents the developed Building Information Modeling (BIM) representation of GB.

Figure 2: 3D BIM Model of the Galbraith Building

This case study utilizes real-world data (measurements and work order records) provided by F&S at UofT. The dataset consists of numerical records from environmental sensors and textual descriptions of maintenance activities. The F&S department manages 130 buildings (totaling 14 million square feet) and processes 35,000 service requests and 30,000 invoices annually ("Facilities & Services - University of Toronto," 2020). Managing this volume of data requires efficient integration, retrieval, and analysis methods to support facility operations effectively. The case study focuses on executing queries that extract relevant insights from both structured and unstructured data sources. The system integrates sensor readings and maintenance records to provide a comprehensive assessment of facility operations. It also supports decision-making for maintenance planning and resource allocation.

Sensor Data

Sensor data is collected from a wide range of environmental monitoring devices installed throughout the GB building. The dataset includes temperature, humidity, CO_2, and occupancy sensors. These sensors' measurements are used for assessing indoor environmental conditions and identifying patterns that may have potential impacts on facility performance. The data is managed within the Environmental Monitoring and Reporting System (EMRS) at UofT, which serves as a centralized platform for organizing sensor readings across campus facilities.

In this study, the sensor readings are stored in a PostgreSQL (PSQL) relational database. The database is structured to support time-series data and optimize query efficiency. Its schema includes dedicated tables that store sensor metadata, measurement values, and timestamps. Each sensor entry is associated with attributes such as sensor ID, type, unit, and assigned location within the building. To maintain data integrity, a set of preprocessing procedures has been employed to identify missing values and detect outliers before analysis.

Knowledge Graph Construction

The case study also integrates a knowledge graph to model relationships between key facility management entities. The knowledge graph can provide a deeper understanding of the interactions between physical infrastructure, environmental data, and maintenance activities. The graph schema consists of key node types such as buildings, rooms, sensors, and work orders. Each node has its defined properties. For example, sensors include attributes like sensor ID, type, and unit. Similarly, work orders capture details such as work order ID, priority, status, total charge, hours spent, and date completed.

In addition to representing entity relationships, the knowledge graph integrates a categorization of work orders grounded in a taxonomy developed specifically for facilities management at UofT. This taxonomy, which is the subject of a separate study currently under review, was derived from an extensive analysis of historical work orders. It organizes maintenance tasks into hierarchical and distinct categories: actions and issues, systems, and work centers. These categories reflect different dimensions of facilities management; for example,

actions and issues encompass subcategories like repair and maintenance, quality control, and security and access, while systems pertain to building infrastructure components such as HVAC, lighting, and water distribution. Work centers identify the specific trades or teams responsible for executing tasks, such as electricians, plumbers, or operating engineers. The categorization is applied automatically to work orders before they are added to the graph.

The knowledge graph also incorporates a critical integration between the sensor nodes and the external relational database that stores high-frequency time-series data. Instead of storing sensor readings in the knowledge graph, the system maintains them in a PostgreSQL database to manage the high volume and temporal nature of the data as described earlier. The graph captures metadata such as sensor type and location, while each sensor node includes a reference to its corresponding dataset in the relational database. This design facilitates navigation between metadata stored in the graph and its associated measurements. This connection keeps the knowledge graph computationally efficient while still enabling advanced analytical use cases, such as correlating sensor measurements with operational workflows or identifying anomalies related to maintenance activities.

Figure 3 provides a visualization of the knowledge graph constructed for the Galbraith Building using Neo4j, a graph database designed for querying complex relationships. The central node represents a specific work order associated with Room 224 in this building. Room 224 has connections with sensor data and work orders. For example, the MONITORED_BY_SENSORS relationship links the room to a sensor group that includes environmental sensors such as carbon dioxide detectors.

The work order classification relies on multiple relationships. The HAS_CATEGORY link associates it with a specific issue type, while the IMPACTS_SYSTEM relationship connects it to affected building systems such as the ceiling and flooring. The ASSIGNED_TO_WC relationship designates the responsible trade (work center), which in this case is Steamfitters.

The node properties panel on the right presents detailed metadata for the selected work order. Key attributes include the work order ID (1099261), status (Completed), priority level (Medium), total charge (450 $), and hours spent (8). The panel also displays the contact information of the reporter, a description of the issue, and relevant timestamps for task creation and completion. To protect data confidentiality, certain details such as costs, contact information, and work order IDs have been altered.

It should be mentioned that this visualization focuses on a single room and includes only a small subset of its associated work orders and sensors within the Galbraith Building. The complete dataset contains approximately 2,500 work orders recorded between 2021 and 2024 for

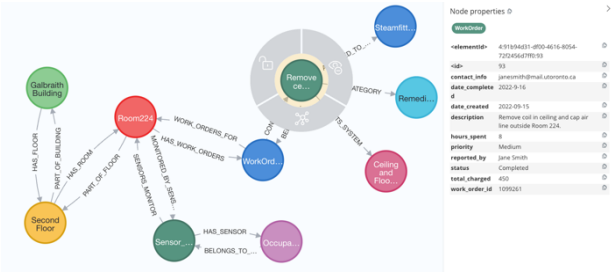

Figure 3: Galbraith Building Knowledge Graph

the Galbraith Building, all of which have been integrated into the knowledge graph.

Agentic System

The agentic system in this case study is built using OpenAI's API, with GPT-3.5 Turbo as the underlying language model. The model operates with a temperature setting of zero to minimize randomness and produce highly consistent responses. To ensure accurate query generation, the system employs prompt engineering techniques to guide the model in generating precise and executable queries for structured databases, knowledge graphs, and analytical tools. Carefully designed prompts ensure that the model correctly interprets user queries, identifies the most suitable retrieval method and tool(s), constructs valid SQL or Cypher queries, generates necessary Python code for analytical insights, and structures the output format.

To handle complex, multi-step user queries, the system utilizes an agent architecture designed for robust reasoning and interactive decision-making. It employs Reasoning and Acting (ReAct) agents, which enhance functionality by combining logical reasoning with action execution. These agents generate reasoning traces alongside their actions, allowing them to refine their approach iteratively, verify intermediate results, and adjust queries as needed. This capability allows them to execute multiple retrieval and processing steps to ensure accurate and contextually relevant responses (Yao et al., 2023).

The architecture of the agentic system, shown in Figure 4, follows a supervisory agentic model where a supervisor agent orchestrates query execution and manages specialized agents responsible for different tasks. The system consists of three specialized agents, each handling a specific type of data processing. Agent 1 retrieves relevant data from the knowledge graph, including details on buildings, rooms, sensors, and maintenance records. It navigates the graph structure to identify relationships between facility components and maintenance activities to retrieve relevant contextual information.

Agent 2 interacts with the structured relational database and designs and executes optimized SQL queries to retrieve required sensor readings. Agent 3 receives Cypher and SQL queries generated by Agent 1 and Agent 2, combines them when necessary, and develops the required Python code to generate insights. Based on the

query context, this agent applies statistical models, anomaly detection, and visualization techniques to analyze the retrieved data. If additional exploration is required, Agent 3 can refine queries and retrieve supplementary information from both the knowledge graph and the relational database.

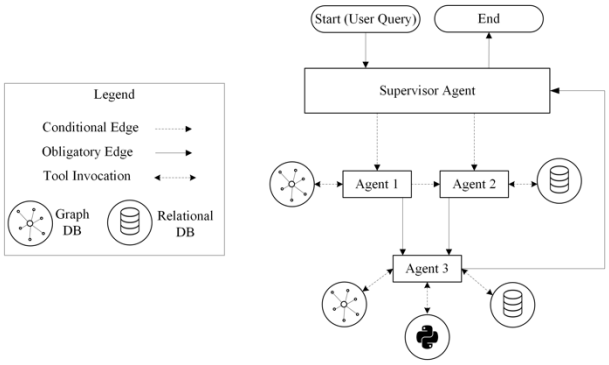

Figure 4: Agentic System Architecture and Execution Flow

The parallel design of Agents 1 and 2 enhances system efficiency by enabling simultaneous execution when a query requires retrieving data from both the relational database and the knowledge graph. If a user query demands information from both databases, the supervisor agent invokes agents 1 and 2 in parallel, reducing response time and improving overall performance. In cases where the query is only related to a specific data source (for example, when retrieving details about a particular sensor is required), the supervisor agent selectively engages the relevant agent (e.g., Agent 2 for structured sensor data). As a result of this dynamic execution, the system uses resources more efficiently and requires less computational effort. In the next step, when the necessary queries are generated, Agent 3 integrates the results, develops analytical code, and generates insights. The supervisor agent's responsibility is to maintain execution flow, monitor interactions between agents, and handle error resolution. Overall, the integration of the supervisor agent in the proposed system can lead to more accurate and contextually relevant responses.

In addition to the ReAct framework, the system incorporates manual validation functions. These functions evaluate the correctness and efficiency of generated queries before execution. They perform syntactic validation (i.e., checking for errors in SQL and Cypher) and schema validation. If an inconsistency is detected, the system modifies the input prompt and regenerates the query. This procedure reduces the likelihood of execution errors and increases response reliability. The validation layer also enforces query restrictions to block unintended modifications such as insertions, deletions, or schema alterations.

An example of the agentic system's process is a query such as "*Identify rooms in the Galbraith Building that have experienced multiple HVAC failures in the past month and determine whether these failures correlate with complaints about temperature fluctuations.*" To answer this, the system retrieves work orders categorized under HVAC-related failures from the knowledge graph. It then extracts metadata for relevant temperature sensors and generates an SQL query to obtain historical readings from the relational database. Finally, the Python tool is utilized to analyze the retrieved data using statistical methods and to detect correlations between failure events and temperature anomalies.

System Implementation for the Case Study

The overall system structure is represented in Figure 5. The system is deployed on the I2C server (Centre for Information Systems in Infrastructure & Construction) at UofT and consists of three main components: the user interface, the multi-agent system, and the data management module. Users interact with the system through a web-based interface that allows them to submit queries and retrieve structured responses. The interface connects to the backend through asynchronous RESTful APIs, which facilitate communication with the multi-agent system. They also route queries and manage response delivery.

The chat panel module functions as the primary access point for user interaction with the multi-agent system. When a user submits a query, the chat panel transmits the request to the multi-agent core via a RESTful API. Once the agents generate a response, the system sends the result back to the chat panel for display. Depending on the complexity of the request, the response may include retrieved information, analytical insights, or visual outputs such as charts and statistical summaries.

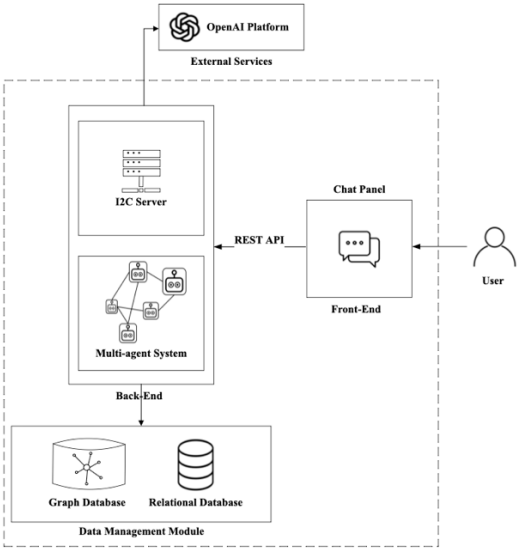

Figure 5: Overall System Architecture

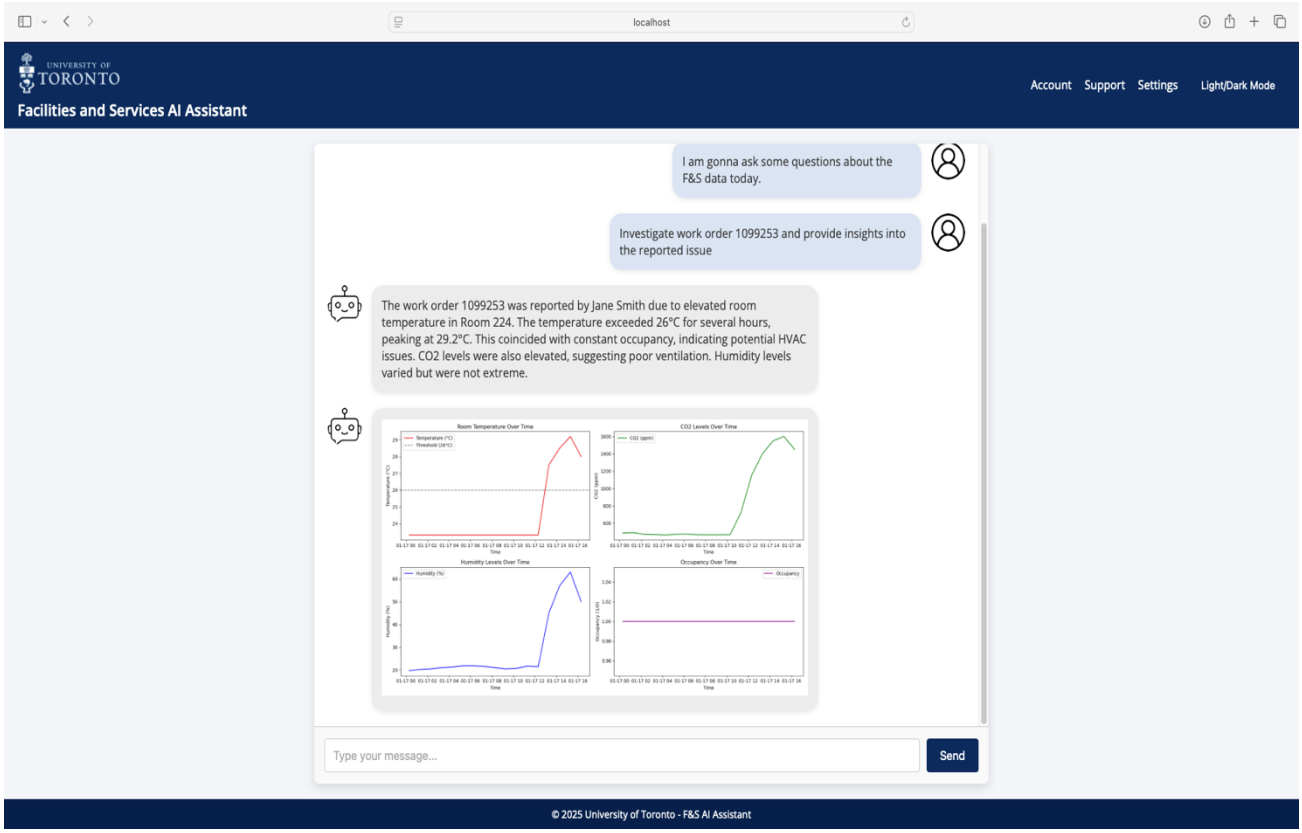

Figure 6: Web-Based User Interface of the Proposed System

Findings and Discussion

To demonstrate the proposed system's capability in retrieving and analyzing facility management data, a query is submitted to investigate a reported issue. The user request states: "*Investigate work order 1099253 and provide insights into the reported issue.*" Upon receiving the query, the agentic system autonomously retrieves the relevant work order from the knowledge graph. The description indicates that the issue involves elevated room temperature. Based on the graph relationships, the system identifies Room 224 as the affected location. It examines the properties of the corresponding Room node in the graph, finds the sensors of this room, and locates references to relevant sensor data in the relational database. It then queries the relational database for historical sensor readings corresponding to the date of the work order, January 17, 2025. The retrieved data includes temperature, CO_2 levels, humidity, and occupancy records, all of which provide insight into the reported issue.

To extract meaningful patterns, the system applies an analytical pipeline that examines variations in environmental conditions. As visualized in Figure 6, the temperature in Room 224 exceeds the acceptable threshold of 26°C, peaking at 29.2°C for several hours. This period of elevated temperature coincides with sustained occupancy, suggesting that the HVAC system may not adequately regulate indoor conditions. Additionally, CO_2 concentrations are significantly elevated, indicating potential ventilation inefficiencies. Humidity levels also fluctuate throughout the period, which may contribute to occupant discomfort. The system presents these findings in natural language and provides visual outputs, including charts that illustrate fluctuations in environmental conditions over time. It also summarizes the correlation between environmental sensor data and the work order description. This response showcases the system's capability to autonomously retrieve relevant data, apply analytical operations, and generate meaningful insights without requiring explicit user instructions to reference sensor records.

The case study demonstrates how the proposed system improves interaction with large-scale facility management data. Traditionally, facility managers have relied on manual information retrieval or fragmented data reviews, a practice that often limits their ability to extract meaningful insights efficiently. However, the system automates retrieval, correlation, and analysis across diverse data sources by leveraging a multi-agent execution framework and LLMs. This automation reduces the time and effort required for tasks such as investigating maintenance issues, assessing environmental conditions, and identifying trends.

LLMs enhance this process by allowing users to interact with complex datasets through natural language queries. Users can request insights in an intuitive way while the system determines the required retrieval steps. This capability makes facility data more accessible. The system extracts both explicit information, such as retrieving work orders related to specific rooms, and implicit insights, such as identifying recurring specific failures based on environmental sensor patterns.

In addition to improving accessibility, the system accelerates insight generation by automatically linking relevant data from different data sources. Work orders, sensor readings, and historical maintenance records often exist in isolation. This isolation limits the ability to detect meaningful patterns. This system bridges these data silos by establishing relationships between facility events and operational conditions, which in turn allows for cross-domain insights that inform decision-making.

The multi-agent execution model improves the system's ability to process complex queries. It divides these questions into manageable tasks and assigns specific responsibilities to specialized agents. As a result, the system can adapt to different facility management scenarios while maintaining efficiency in query execution. The multi-agent model also extends the system's scalability further by enabling the integration of new agents with specialized roles. As new data sources, analytical tools, or processing techniques become necessary, additional agents can be introduced and incorporated into the framework without requiring major modifications to existing components. This flexibility allows the system to evolve alongside changing facility management needs while preserving a structured and efficient execution process.

Despite its effectiveness in integrating facility management data, the system faces several limitations that restrict its current capabilities and suggest directions for future work. First, its performance depends on the completeness and consistency of work order descriptions and sensor metadata. This dependency makes it sensitive to missing entries or incorrect associations. The system also relies on the correctness of outputs generated by the LLM, which may occasionally produce invalid queries or partial reasoning chains. Although the implemented prompt design and query validation mechanisms reduce such risks, maintaining consistent reliability across a wide range of query types remains a technical challenge.

In terms of reliability evaluation, the current implementation includes syntactic and schema-level validation but does not yet incorporate formal benchmarking against ground-truth data or controlled test sets. Future work will include structured evaluations that combine automatic benchmarking—using curated query-response pairs for accuracy testing—with user-centered studies involving domain experts. These evaluations will assess the correctness, completeness, and contextual relevance of generated responses across diverse query types.

Future work will also include improving LLM capabilities to refine query interpretation, particularly for ambiguous or multi-intent queries. In addition, expanding predictive analytics will allow the system to forecast equipment failures and maintenance needs based on sensor patterns. Finally, extending the system to process real-time sensor data streams instead of relying solely on historical records will further improve its ability to support proactive facility management.

The system lays the groundwork for proactive and prescriptive maintenance by structuring facility data in a graph-based model that captures dependencies between sensors, equipment, and historical maintenance records. While the current implementation focuses on retrospective analysis, integrating graph-based anomaly detection, machine learning, and predictive modeling could enable early identification of failure patterns and recurring issues. Further advancements in graph traversal techniques, temporal analysis, and prescriptive analytics would allow the system to not only anticipate equipment failures but also recommend optimal maintenance actions, resource allocation strategies, and intervention timing. This shift from reactive to data-driven predictive and prescriptive management could enhance operational efficiency and extend the lifespan of facility assets.

Conclusions

This study presents a framework that integrates graph-based data modeling, structured time-series storage, and a multi-agent execution system to automate facility management data analysis and support decision-making. By leveraging LLMs for natural language interaction and multi-agent execution for query processing, the system automates retrieval, correlation, and analysis across diverse data sources. The case study demonstrates its ability to extract meaningful insights by linking work orders, sensor readings, and historical maintenance records, reducing the effort required for data interpretation. While the current implementation focuses on retrospective analysis, the system lays the foundation for predictive and prescriptive maintenance through future integration of graph-based anomaly detection, machine learning, and real-time monitoring. Expanding these capabilities will further improve decision-making, operational efficiency, and resource optimization in facility management.

Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work the author(s) used ChatGPT in order to assist in improving writing clarity. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the publication.

References

About us - Facilities & Services - University of Toronto [WWW Document], 2020. URL https://www.fs.utoronto.ca/about/ (accessed 1.31.25).

Ahmed, V., Tezel, A., Aziz, Z., Sibley, M., 2017. The future of big data in facilities management: opportunities and challenges. Facilities 35, 725–745.

Atkin, B., Brooks, A., 2021. Total facility management.

Babaei Ravandi, A., El-Diraby, T., 2024. Adopting Automation in Premanufacturing: A Two-Mode Network Analysis on Factors and Roles in Iran and North America's Construction Industry, in: Gonzalez-Moret, V., Zhang, J., García de Soto, B., Brilakis, I. (Eds.), Proceedings of the 41st International Symposium on Automation and Robotics in Construction. International Association for Automation and Robotics in Construction (IAARC), Lille, France, pp. 299–306. https://doi.org/10.22260/ISARC2024/0040

Cheng, J.C.P., Chen, W., Chen, K., Wang, Q., 2020. Data-driven predictive maintenance planning framework for MEP components based on BIM and IoT using machine learning algorithms. Autom. Constr. 112, 103087. https://doi.org/10.1016/j.autcon.2020.103087

Dixit, M.K., Venkatraj, V., Ostadalimakhmalbaf, M., Pariafsai, F., Lavy, S., 2019. Integration of facility management and building information modeling (BIM) A review of key issues and challenges. Facilities 37, 455–483.

Jeon, C.-H., Shim, C.-S., Lee, Y.-H., Schooling, J., 2024. Prescriptive maintenance of prestressed concrete bridges considering digital twin and key performance indicator. Eng. Struct. 302, 117383. https://doi.org/10.1016/j.engstruct.2023.117383

Li, B., Luo, Y., Chai, C., Li, G., Tang, N., 2024. The Dawn of Natural Language to SQL: Are We Fully Ready? Proc. VLDB Endow. 17, 3318–3331. https://doi.org/10.14778/3681954.3682003

Li, L., Jiang, S., Yuan, J., Zhang, L., Xu, X., Wang, J., Zhou, Y., Li, Y., Xu, J., 2024. From data silos to seamless integration and coordination: a data-asset centric approach to smart hospital facility management. Eng. Constr. Archit. Manag.

Li, Y., Liu, Y., Zhang, J., Cao, L., Wang, Q., 2024. Automated analysis and assignment of maintenance work orders using natural language processing. Autom. Constr. 165, 105501.

Mannino, A., Dejaco, M.C., Re Cecconi, F., 2021. Building information modelling and internet of things integration for facility management—Literature review and future needs. Appl. Sci. 11, 3062.

Mo, Y., Zhao, D., Du, J., Syal, M., Aziz, A., Li, H., 2020. Automated staff assignment for building maintenance using natural language processing. Autom. Constr. 113, 103150.

Peng, C., Xia, F., Naseriparsa, M., Osborne, F., 2023. Knowledge Graphs: Opportunities and Challenges. Artif. Intell. Rev. 56, 13071–13102. https://doi.org/10.1007/s10462-023-10465-9

Shamshiri, A., Ryu, K.R., Park, J.Y., 2024. Text mining and natural language processing in construction. Autom. Constr. 158, 105200.

Singh, I., Traum, D., Thomason, J., 2024. TwoStep: Multi-agent Task Planning using Classical Planners and Large Language Models. https://doi.org/10.48550/arXiv.2403.17246

Sobhkhiz, Soroush, El-Diraby, T., 2023. Dynamic integration of unstructured data with BIM using a no-model approach based on machine learning and concept networks. Autom. Constr. 150, 104859. https://doi.org/10.1016/j.autcon.2023.104859

Sobhkhiz, S., El-Diraby, T., 2023. Leveraging text mining and network analysis for a semi-automated work order process analytics, in: ECPPM 2022-eWork and eBusiness in Architecture, Engineering and Construction 2022. CRC Press, pp. 577–582.

Talamo, C., Bonanomi, M., 2015. Knowledge Management and Information Tools for Building Maintenance and Facility Management. Springer International Publishing, Cham. https://doi.org/10.1007/978-3-319-23959-0

Valinejadshoubi, M., Moselhi, O., Bagchi, A., 2022. Integrating BIM into sensor-based facilities management operations. J. Facil. Manag. 20, 385–400.

Xi, Z., Chen, W., Guo, X., He, W., Ding, Y., Hong, B., Zhang, M., Wang, J., Jin, S., Zhou, E., Zheng, R., Fan, X., Wang, X., Xiong, L., Zhou, Y., Wang, W., Jiang, C., Zou, Y., Liu, X., Yin, Z., Dou, S., Weng, R., Qin, W., Zheng, Y., Qiu, X., Huang, X., Zhang, Q., Gui, T., 2025. The rise and potential of large language model based agents: a survey. Sci. China Inf. Sci. 68, 121101. https://doi.org/10.1007/s11432-024-4222-0

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., Cao, Y., 2023. ReAct: Synergizing Reasoning and Acting in Language Models.

Your trusted partner on campus. - Facilities & Services - University of Toronto [WWW Document], 2021. URL https://www.fs.utoronto.ca/ (accessed 1.31.25).