World Journal of Information Technology
Print ISSN: 2959-9903
Online ISSN: 2959-9911
DOI: https://doi.org/10.61784/wjit3069

DESIGN AND APPLICATION OF A KNOWLEDGE-GRAPH-BASED INTELLIGENT QUESTION ANSWERING SYSTEM FOR REACTOR OPERATION AND MAINTENANCE

Yuan Zhang*, Chao Si, YanKun Li
Electrical Department, China Institute of Atomic Energy, Beijing 102400, China.
**Corresponding Author: Yuan Zhang*

Abstract: Traditional knowledge management in nuclear reactor operation and maintenance (O&M) relies on dispersed documents and individual expertise, limiting knowledge reuse and response efficiency. This paper introduces a knowledge-graph-driven intelligent question answering (QA) system that constructs a unified semantic representation of equipment, faults, and procedures through structured entity extraction, relationship modeling, and rule-based reasoning. Neo4j serves as the graph database, while the Dijkstra shortest path algorithm supports association path computation for fault cause inference and similar-case retrieval. The system integrates with a digital O&M platform for real-time data acquisition and structuring. Simulation-based validation in a reactor refueling system scenario demonstrates that the system shows promising potential in the following aspects: enhancing standardization and traceability of fault information, reducing fault diagnosis and handling time, and providing reliable data support for team capability evaluation and equipment health assessment. This work provides a scalable framework for knowledge management in complex industrial systems, contributing to improved O&M efficiency and safety in nuclear operations.
Keywords: Knowledge graph; Intelligent question answering; Nuclear reactor operation and maintenance; Fault diagnosis; Knowledge reasoning

1 INTRODUCTION

1.1 Research Background and Significance

As the core facility of nuclear energy utilization, the operation and maintenance (O&M) of nuclear reactors is directly related to equipment safety, personnel safety, and environmental safety. During long-term operation, reactors generate large volumes of O&M work orders, maintenance records, fault diagnosis reports, and expert documentation. These data are highly specialized, time-sensitive, and subject to stringent security constraints.

In conventional practice, O&M knowledge is mainly preserved in the form of paper records, office documents, and scattered experience summaries. It lacks unified structured organization and semantic linkage, which leads to several key problems:

1.1.1 Fragmented knowledge and low retrieval efficiency

Fault-handling experience is distributed across multiple systems and personal records. When facing complex or atypical faults, O&M personnel must manually search through numerous documents, resulting in low efficiency and inconsistent decision quality.

1.1.2 Knowledge transfer heavily depends on individuals

A large amount of tacit knowledge is concentrated in a small number of senior experts and key personnel. New staff members cannot easily obtain a systematic understanding of complex equipment fault patterns and handling strategies within a short period of time.

1.1.3 Limited support for intelligent analysis

In the absence of unified data models and standardized coding, historical fault data cannot be directly leveraged for big data analysis, health assessment, or intelligent recommendation.

Knowledge graphs have emerged as a powerful structured knowledge representation centered on entities and their relationships and have been extensively studied and applied in recent years, including in search engines, industrial internet platforms, and complex engineering systems [1–3]. By abstracting heterogeneous multi-source data into a graph of entity nodes and relationship edges, a knowledge graph supports semantic-level querying, reasoning, and visualization, thereby providing new means for knowledge organization and intelligent analysis in complex O&M environments. In the power and nuclear equipment domains, knowledge graphs have been used to support equipment management, fault diagnosis, and O&M decision support, including for power equipment operation and maintenance [4] and for knowledge-graph-based QA systems in technical domains [5].

Compared with general-purpose large language models (LLMs) such as ChatGPT, the reactor O&M domain imposes much stricter requirements on data security, knowledge accuracy, and traceability. O&M data usually reside in isolated intranet environments and contain sensitive information, making them unsuitable for large-scale open training. At the same time, O&M decision-making requires explicitly traceable knowledge sources and interpretable reasoning processes. Against this backdrop, constructing an intranet-based intelligent QA system with a knowledge graph as its core, which structurally consolidates proprietary O&M knowledge and provides controllable and explainable QA capabilities, is of significant engineering and practical value.

1.2 Related Work

1.2.1 Knowledge graphs and intelligent question answering

Knowledge graphs have emerged as a significant research area in artificial intelligence and data engineering over the past decade. Substantial research efforts have addressed knowledge representation, acquisition, fusion, refinement, completion, and evolution, leading to well-established construction methodologies and evaluation frameworks [1–3, 6].

At the application level, knowledge graphs are widely used in semantic search, intelligent QA, recommendation systems, and industrial O&M. By making entity relationships explicit, they enhance machine understanding and reasoning capabilities in complex domains. Knowledge-graph-based QA systems typically consist of three key components:

1. Natural language understanding (NLU).

Natural-language questions are transformed into logical or formal queries over the graph through entity recognition, intent recognition, and semantic parsing [2,5].

2. Graph querying and reasoning.

The knowledge graph is traversed using pattern matching and constraint filtering to retrieve relevant entities and relationships; graph algorithms and rule-based reasoning may be applied to infer new relations or rank candidate answers.

3. Answer generation.

Retrieved results are integrated and ranked and then presented to users in natural language or structured tabular/graphical form.

Compared with purely text-based large models, knowledge-graph-based QA systems generally offer higher knowledge accuracy, better interpretability, and stronger controllability, particularly in industrial scenarios characterized by strict regulations and high operational risk [4,5].

1.2.2 Graph databases and neo4j in industrial O&M

Graph databases are specialized databases designed for graph-structured data. They use nodes and edges as fundamental storage units and support efficient graph traversal and pattern matching. Compared with relational databases, graph databases provide higher flexibility and performance for representing complex relationships and executing path queries [7,8].

Neo4j is one of the most widely used open-source graph databases and is particularly suitable for building and querying enterprise knowledge graphs [7]. It offers robust ACID transaction support, native graph storage, the Cypher query language, and integration with a range of graph analytics libraries. These features make Neo4j well suited for applications in financial risk control, social networks, recommendation systems, and equipment O&M.

In the energy and nuclear sectors, several studies have used Neo4j to construct knowledge centers for power or nuclear equipment, in which operational data, equipment information, and procedural documents are organized as graphs to provide visual querying and decision support for O&M personnel. For example, knowledge graphs have been used to model power equipment operation and maintenance entities and their relationships, supporting visual analytics and fault tracing [4].

1.2.3 Graph processing frameworks and shortest path algorithms

Large-scale graph processing systems such as Pregel [9], GraphLab [10], GraphChi [11], Trinity [12], and Horton [13] provide distributed or out-of-core computation frameworks for massive graphs. They support iterative algorithms (including PageRank, label propagation, and shortest path algorithms) under different execution models, and have inspired the design of graph analytics capabilities in modern graph databases and engines.

The Dijkstra algorithm is a classic single-source shortest path algorithm for weighted directed graphs with non-negative edge weights. It has been widely used for route planning, network routing, and path optimization [14]. In knowledge graph scenarios, the entity–relation structure can be modeled as a weighted graph; by assigning weights to relationship edges, Dijkstra's algorithm can be used to quantify association strength between entities and discover the most relevant knowledge or case paths. This enables path-based reasoning and recommendation, which is particularly important in safety-critical industrial O&M settings.

1.2.4 Reactor O&M knowledge graphs and intelligent QA

In the context of reactor O&M, several studies have proposed using knowledge-graph technologies for semantic modeling of equipment information, operating parameters, O&M tickets, and safety procedures, and have built "smart nuclear power" platforms to support equipment management, safety management, and operator training. Knowledge-graph-based QA systems for technical domains have also been explored as a means of improving information retrieval and expert knowledge reuse.

However, publicly available literature shows that systems specifically targeting reactor refueling systems and, at the same time, integrating process information digitalization, knowledge graph modeling, shortest path reasoning, and intelligent QA within a unified framework remain relatively rare. Existing systems often focus on knowledge centers or visual analytics tools, with insufficient support for natural-language QA, intelligent recommendation, and deep integration with O&M workflow systems. Under intranet security constraints, how to organically combine O&M process data, fault knowledge, and intelligent QA remains an open research and engineering problem.

1.3 Research Objectives and Contributions

To address the above challenges, this paper focuses on the design and application of a knowledge-graph-based intelligent QA system for reactor O&M. The main research tasks are as follows:
a) Reactor O&M knowledge graph modeling
Based on the equipment structure, fault modes, and handling processes of the reactor refueling system, we design a knowledge-graph schema tailored to O&M scenarios, specify entity types, relationship types, and key attributes, and propose methods for entity and relationship extraction and cleaning from historical O&M tickets and expert experience.
b) Knowledge storage and update mechanisms based on Neo4j and a rule base
Neo4j is selected as the storage and query engine for the knowledge graph. A fault description lexicon, a fault handling method library, and an O&M knowledge base are constructed. Combined modeling using a rule base and graph database is implemented, together with a semi-automatic knowledge update mechanism appropriate to O&M scenarios.
c) Intelligent QA and recommendation using the Dijkstra algorithm
The Dijkstra shortest path algorithm is introduced into the knowledge graph. Associations among fault phenomena, equipment components, and handling solutions are modeled as a weighted graph. Path search and similarity computation methods are designed for the "fault diagnosis–handling recommendation–similar case retrieval" chain, enabling natural-language QA and intelligent recommendation for O&M personnel.
d) Integration with a digital O&M process information platform
The intelligent QA system is integrated with a digital O&M process information management platform. Business data related to work request creation, transmission, dispatch, and closure are captured in structured form and accumulated as knowledge, forming a closed loop of "data–knowledge–QA–O&M decision."
The main contributions of this work can be summarized as follows:
a) A complete modeling and construction process for a reactor O&M knowledge graph
The proposed method unifies equipment hierarchical structure, fault localization information, and handling strategies within a single graph model and provides an engineering path that can be implemented with intranet O&M data.
b) A graph reasoning mechanism that combines a rule base with the Dijkstra shortest path algorithm
While ensuring interpretability, the proposed mechanism supports fault path search and similar case recommendation, enabling the QA system to output path-level explanations.
c) Deep integration of the knowledge graph with a digital O&M process platform
O&M workflows and knowledge accumulation mechanisms are organically combined, offering an end-to-end solution for nuclear facility O&M that spans data collection, knowledge management, and intelligent QA, as illustrated in the overall system architecture and intelligent QA workflow.

2 RELATED TECHNOLOGIES AND RESEARCH METHODS

This section presents the key technical foundations underpinning the knowledge-graph-based intelligent QA system for reactor O&M. These include the construction methods and characteristics of knowledge graphs, collaborative modeling with the Neo4j graph database and a rule base, the principles and application of the Dijkstra shortest path algorithm, and natural-language processing (NLP) and semantic understanding techniques for O&M scenarios. Together, these elements constitute the theoretical basis and implementation framework of the intelligent QA system, and support intelligent retrieval, semantic reasoning, and handling method recommendation.

2.1 Overview of Knowledge Graph Technology

2.1.1 Definition and characteristics of knowledge graphs

A knowledge graph (KG) is a semantic network structure centered on "entity–relation–attribute" triples that describes entities in the real world and their interrelations. In essence, it is a directed graph consisting of nodes (entities) and edges (relations), characterized by strong connectivity, interpretability, and extensibility.
In reactor O&M scenarios, a knowledge graph can be used to represent:
1) Equipment hierarchical structure (system–equipment–component–subcomponent);
2) Fault types, fault locations, and fault phenomena;
3) Fault handling methods and expert experience;
4) O&M process records (work request creation–dispatch–closure);
5) Auxiliary information such as materials, toolkits, and personnel competencies.
In particular, a knowledge graph can effectively associate scattered expert experience, component knowledge, and fault data, and is well suited for knowledge management in highly confidential, highly specialized, and structurally complex O&M environments.
Compared with large language models, knowledge graphs are more appropriate in nuclear facility contexts mainly because:
1) Knowledge is explicit, accurate, and controllable, in line with stringent nuclear O&M safety requirements;
2) Relationships are explicitly structured, supporting path-level reasoning;
3) Rule-based knowledge updates and traceability are straightforward;
4) Integration with O&M workflow systems (e.g., work ticketing and fault records) is facilitated.
Thus, knowledge graphs are naturally suited to serve as the foundational architecture of the intelligent QA system.

2.1.2 Knowledge graph construction process

In general, knowledge graph construction includes data acquisition, entity recognition, relationship extraction, knowledge fusion, and storage in a graph database.
In the reactor O&M domain, the construction process typically involves:
a) Data acquisition
Historical O&M tickets and maintenance records; equipment structure trees and function trees; fault mode classification standards; expert experience and procedural texts.
b) Data preprocessing
Text cleaning and normalization; synonym normalization (e.g., different names for the same refueling manipulator); standardization and coding of fault records.
c) Entity recognition (NER)
Equipment entities (system/equipment/component/subcomponent); fault entities (type/location/phenomenon); personnel, toolkits, and work order information.
d) Relationship extraction
Equipment hierarchies; relations such as “fault phenomenon–handling method” and “fault location–fault type”; and other domain-specific relations.
e) Knowledge fusion and disambiguation
Conflict resolution and redundancy elimination while complying with existing structure and coding standards.
f) Graph database storage and index optimization (Neo4j)
Storing nodes, edges, and attributes in a graph database for efficient traversal and querying.
g) Knowledge update mechanisms
Automatic updates via NLP-based text extraction; semi-automatic updates with human review; and manual maintenance via expert input.
These steps provide the foundation for mapping natural-language questions to graph representations and performing subsequent retrieval and reasoning.

2.2 Graph Database and Rule Base Technologies

In practical engineering scenarios, a knowledge graph usually needs to manage large-scale entities, relationships, and attributes. This requires a reliable graph database and a rule base to express constraints and support logical reasoning.

2.2.1 Graph database technology (Neo4j)

Neo4j is one of the most widely used open-source graph databases and serves as the core data storage platform of the intelligent QA system [7,8]. It offers high performance, high availability, and ACID transaction support. Its main features include:
a) Efficient graph traversal
The Cypher query language supports pattern-matching-based relationship retrieval, which is well suited for fault path search and equipment relationship analysis.
b) Flexible knowledge evolution
As nuclear facility O&M knowledge evolves with equipment upgrades and accumulated experience, Neo4j can dynamically add nodes and edges without requiring costly schema changes.
c) Built-in graph algorithms
Neo4j provides interfaces for shortest path algorithms, including Dijkstra, which support path search and optimization in fault causality analysis and similar case retrieval.
In our system, Neo4j is deployed within the secure intranet environment as part of the central data center infrastructure (see Figure 2), and stores the core O&M knowledge graph (see Figure 5).

2.2.2 Rule base technology

In a single-domain, data-bounded setting, rule bases are an efficient and interpretable means of knowledge representation [5].
In reactor O&M scenarios, a rule base is typically used to encode:
a) Fault mode classification rules;
b) Mappings from fault phenomena to handling methods;
c) O&M workflow rules (e.g., work request creation, dispatch, and closure procedures);
d) Data quality checking rules.
A typical rule can be expressed as a conditional mapping from fault type and location to recommended handling actions.
Combined with Neo4j, the rule base:
a) Makes domain knowledge explicit;
b) Ensures the interpretability of inferences;
c) Supports more complex logic when coupled with graph traversal and analytics.
Consequently, Neo4j and the rule base together form the dual-core knowledge engine of the system.

2.3 Shortest Path Reasoning: The Dijkstra Algorithm

2.3.1 Principles of the dijkstra algorithm
The Dijkstra algorithm is a classic single-source shortest path algorithm for weighted graphs with non-negative edge weights [14]. Its time complexity is typically

$$O((|V| + |E|)\log |V|), \tag{1}$$

which makes it suitable for graphs with numerous nodes and relatively sparse edges, such as reactor equipment topologies and knowledge graphs.
Its core steps are:
1) Initialize the distance of each node (0 for the source node and ∞ for all others).
2) Repeatedly select the node with the current minimal cost among unvisited nodes.
3) Update the tentative shortest distances of its neighboring nodes.
4) Continue until all nodes have been visited.

2.3.2 Application of dijkstra in the reactor O&M knowledge graph
In the proposed system, the Dijkstra algorithm is used for:
a) Calculation of fault association paths
For example, starting from a fault phenomenon such as “abnormal vibration” to an equipment component such as the “refueling gripper arm,” there may be multiple paths via sensor readings, historical cases, fault types, and related components. Dijkstra’s algorithm helps identify the shortest and most plausible explanatory chain.
b) Similar fault retrieval
By defining edge weights based on similarity costs among fault entities, the shortest path between two fault nodes can be used as a similarity measure to retrieve similar historical cases.
c) Fault cause inference
The shortest causal chain in the knowledge graph can be used to infer the most probable cause for a given fault, providing interpretable evidence for O&M decision-making.
Edge weights can be determined by expert scores, historical occurrence frequencies, or statistical indicators. The shortest path-based reasoning mechanism is integrated into the overall QA workflow.

2.4 Natural-Language Processing and Semantic Understanding

To implement the intelligent QA system, the system must understand the natural-language input from O&M personnel (e.g., work request descriptions, fault descriptions) and map it to entities and relationships in the knowledge graph.

2.4.1 Chinese word segmentation and keyword extraction
Since the primary operational language in the target environment is Chinese, the system uses Chinese word segmentation and keyword extraction techniques. Typical techniques include TF–IDF-based keyword extraction, TextRank, and domain-specific lexicons constructed from the O&M knowledge base.
For example, from the description
“end-effector motion delay, suspected joint sticking,”
the system extracts entities such as “end effector” and “joint,” and fault phenomena such as “motion delay” and “sticking,” and then maps them to corresponding graph entities.

2.4.2 Intent recognition and semantic matching
Intent recognition identifies whether a query concerns fault diagnosis, handling guidance, historical case retrieval, or statistics. Semantic matching is implemented using a hybrid of keyword search, fuzzy string matching, and embedding-based semantic similarity. Word and sentence embeddings are learned using distributional representation methods such as word2vec [15], with domain adaptation for O&M terminology.
For instance, the query
“what should I do if the refueling device gets stuck?”
is mapped to a standardized handling pattern such as “sticking → lubrication check → removal of foreign objects → joint function test.”
Advanced embedding-based techniques for relational reasoning and one-shot learning over knowledge graphs [6] provide a conceptual reference for extending the current system to data-driven link prediction and completion in future work.

2.4.3 Knowledge graph query language
At the conceptual level, the system adopts SPARQL-like query patterns for knowledge retrieval. At the storage level, Cypher is employed to express graph patterns, constraints, and path queries. Users are not exposed to low-level query syntax; instead, queries are automatically generated by the NLU and reasoning modules and subsequently executed by the Neo4j engine.

3 SYSTEM DESIGN AND OVERALL ARCHITECTURE

This section presents the design of the knowledge-graph-based intelligent QA system for reactor O&M, including the overall system architecture, network architecture, data architecture, and the design of core functional modules. The system adopts an integrated architecture that combines process information digitalization, knowledge graphs, path reasoning, and intelligent recommendation. Fault information, equipment structure, handling methods, and expert experience in the O&M process are modeled in a unified manner, thereby realizing a full process loop from data collection and knowledge accumulation to intelligent QA.

3.1 Overall System Architecture

The overall system architecture consists of six layers: user layer, input layer, application layer, support layer, data layer, and infrastructure layer, as shown in Figure 1.

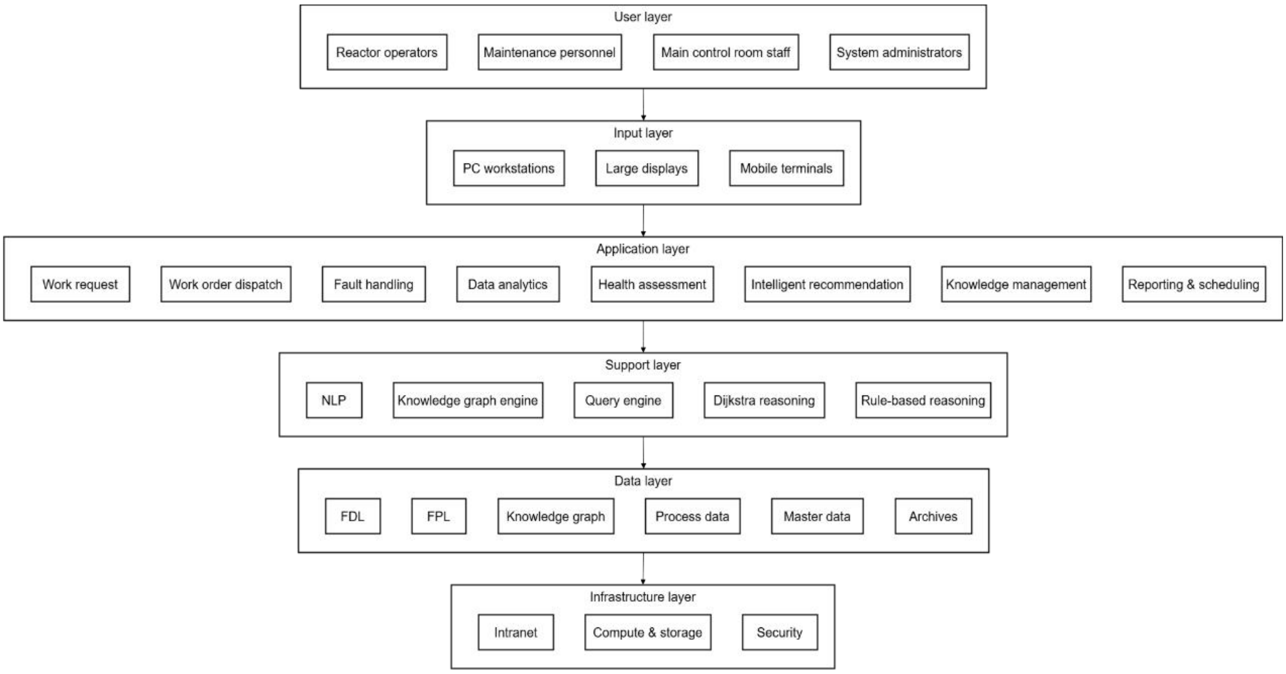

Figure 1 Overall System Architecture of the Reactor O&M Intelligent QA System

a) User layer
Reactor operators, maintenance personnel, main control room staff, and system administrators.
b) Input layer
PC workstations, large displays in control rooms, and mobile terminals.
c) Application layer (O&M business functions)
Work request creation and transmission; work order dispatching and notifications; fault handling and work order closure; O&M big data analytics; equipment health assessment; intelligent recommendation; knowledge base management; statistical reporting; and shift and team scheduling.
d) Support layer (NLP, knowledge graph, reasoning algorithms)
Chinese word segmentation and feature extraction; Cypher/SPARQL-based query engine; shortest-path reasoning (Dijkstra); and logical reasoning based on a rule base.
e) Data layer
Fault description lexicon; fault handling method library; O&M knowledge base; O&M process data; personnel and equipment master data; and historical archives.
f) Infrastructure layer
Intranet, storage, computing resources, and security mechanisms.
The overall design goals of the system are:
a) Knowledge explicitation
Transform equipment structure, fault modes, and handling experience into a knowledge graph.
b) Process information digitalization
Fully digitalize and structure workflows such as work request creation, work order dispatch, and closure, as illustrated in Figure 4.
c) Intelligent QA and reasoning
Support natural-language queries, shortest-path reasoning over the knowledge graph, and context-aware handling method recommendation (see Figure 6).
d) Data analytics and assessment
Provide equipment health assessment, team capability evaluation, and O&M status analysis.
e) Closed-loop application
Implement a sustainable knowledge cycle of “data collection → knowledge extraction → intelligent QA → result

feedback," which is realized through the joint operation of the digital process platform (Figure 4), the knowledge graph (Figure 5), and the QA workflow (Figure 6).

3.2 Network Architecture Design

The system runs in a controlled intranet environment and follows the principle of "multi-end input, centralized processing, and layered protection." The network architecture is shown in Figure 2.

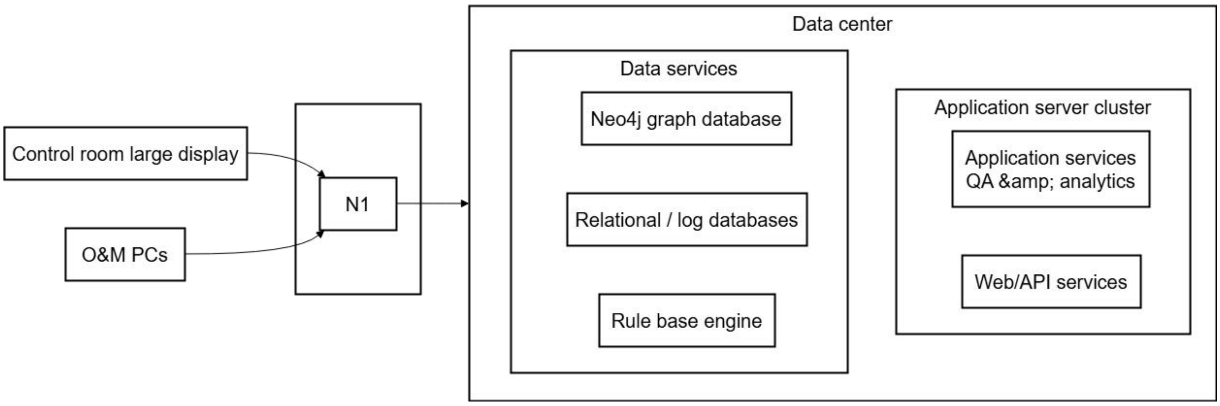

Figure 2 Network Architecture of the Reactor O&M Intelligent QA System

Control room displays and O&M PCs access the system via the intranet. A data center hosts a server cluster, including application servers, a Neo4j graph database, a rule base engine, and analytics services.
Key characteristics are:
a) Intranet isolation ensures that sensitive O&M data do not leave the secure environment.
b) Multi-terminal access supports PC, mobile, and large-screen terminals.
c) Centralized applications host intelligent QA, knowledge graphs, and big data analytics in the data center.
d) Security protection includes firewalls, access control, and audit logging.

3.3 Data Architecture Design

The data architecture is at the core of the intelligent QA system and can be abstracted into five zones, as shown in Figure 3.

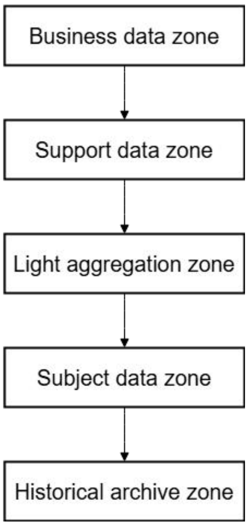

Figure 3 Data Architecture of the Reactor O&M Intelligent QA System

a) Business data zone
Structured O&M fault data, equipment master data, team and shift data, and knowledge graph entities (equipment, faults, handling methods, locations, components).
b) Support data zone
Indicator systems, model data (health and status assessment), system logs, and user permissions.
c) Light aggregation zone
Attendance statistics, fault statistics, team efficiency statistics, and refueling equipment operational data.

d) Subject data zone
Health assessment results, state diagnosis results, and data for integrated dashboards.
e) Historical archive zone
Historical data for work requests, dispatch, and closure, as well as knowledge about legacy components and typical fault cases.
The main categories of reactor O&M data are:
Master data: equipment codes, equipment tree, personnel data.
Source: equipment management system.
Use: basis for knowledge-graph entities.
Fault data: fault description, location, type, phenomenon.
Source: work request and closure data.
Use: nodes and relationships in the knowledge graph.
Handling methods: step sequences and toolkits.
Source: O&M experience and records.
Use: QA and recommendation.
Process data: work request/dispatch/closure records and repair duration.
Source: on-site O&M activities.
Use: big data analytics.
Unstructured data: free-text records and expert opinions.
Source: documents and expert interviews.
Use: NER and knowledge extraction.

3.4 Normalization and Digitalization of O&M Information

The normalization and digitalization of O&M information comprise three major components:
1. Standardization of equipment and component naming.
2. Construction of the fault description lexicon (FDL).
3. Construction of the fault handling method library (FPL).
These elements form the foundational knowledge layer of the intelligent QA system.
3.4.1 Standardization of equipment and component names
Reactor O&M equipment adheres to strict hierarchical structures, generally following

$$\text{Equipment tree} = \{D_i | D_i \rightarrow P_{ij} \rightarrow C_{ijk} \rightarrow S_{ijkl}\}, \tag{2}$$

where $D_i$denotes major units or systems, P_{ij}equipment, C_{ijk}components, and S_{ijkl}subcomponents.
Standardization requires unified codes for identically named equipment, synonym sets for multiple naming conventions, and compliance with existing structure rules and coding standards. This standardization is reflected in the master data modeling in the business data zone of Figure 3 and in the Device entities in the knowledge graph architecture in Figure 5.
3.4.2 Design of the Fault Description Lexicon (FDL)
The Fault Description Lexicon (FDL) consists of standardized entries extracted from high-frequency fault records and is formally expressed as

$$\text{FDL} = \{f_i | f_i = (\text{equipment,location,phenomenon,mode})\}. \tag{3}$$

Phenomena include “vibration,” “motion delay,” and “failure to reset,” while modes include “sticking,” “failure,” and “abnormal vibration.” The lexicon is obtained by systematically analyzing historical records and is continually enriched and normalized through the digital process flow shown in Figure 4.
3.4.3 Fault Processing Library (FPL)
The Fault Processing Library (FPL) contains standardized handling procedures:

$$\text{FPL} = \{s_j | s_j = (\text{step sequence,tools,duration})\}. \tag{4}$$

For example, a handling procedure for a joint sticking fault may specify an ordered sequence of lubrication checks, cleaning operations, and functional tests, together with required tools and estimated time. The library supports manual additions and periodic standardized updates while remaining synchronized with the knowledge graph and O&M process data.

3.5 Design of the Digital O&M Process Information Platform

The digital O&M process information platform is the main data entry point for the knowledge graph and provides real-time data support for intelligent QA. Its functional modules and workflows are illustrated in Figure 4.

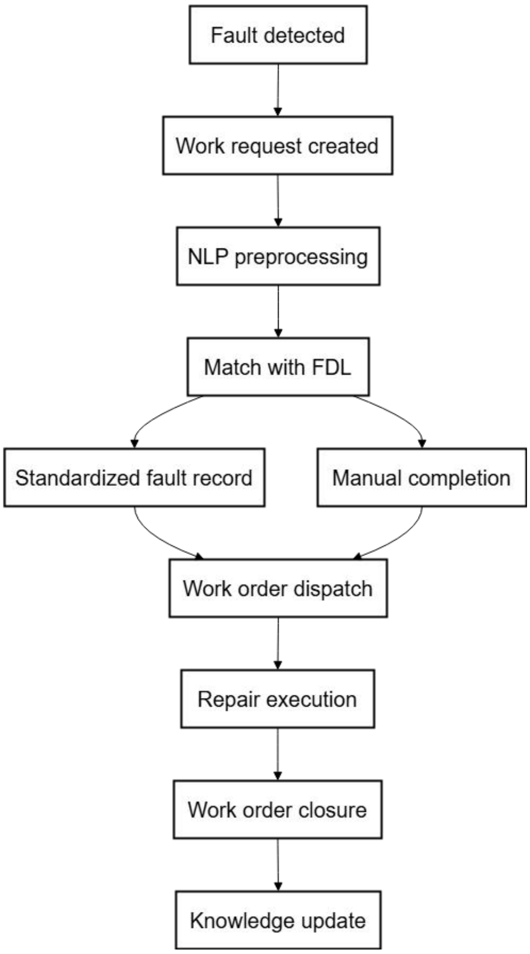

Figure 4 Digital O&M Process Flow for Work Requests, Dispatch, and Closure

Its functional architecture includes:
- O&M business management (work request creation, dispatch, transfer, closure).
- Fault information preprocessing and semantic analysis (NLP, fuzzy and semantic search, lexicon matching, automatic coding).
- Data storage (FDL, FPL, fault information database, equipment master data, and team data).

3.5.1 Work request (ticket creation) process

The work request process collects fault phenomena identified by O&M personnel. The main steps are:
1. Fault description is input via PC, large display, or mobile terminal.
2. Text parsing and keyword matching are executed, including similarity-based matching between the input description and FDL entries. The lexicon entry with the highest similarity score is selected.
3. Fault description is confirmed, with the option to add new entries when no suitable standardized description exists.
4. Automatic coding is applied according to fault mode coding standards.
5. Data are stored, forming a complete digital work request record and feeding the FDL and knowledge graph (Figure 5).

3.5.2 Work order dispatch process

Work order dispatch is triggered automatically or manually based on fault information and current shift arrangements. Dispatch decisions take into account equipment location, team assignment, and personnel competence. The system selects an appropriate team or defers the decision to manual dispatch when constraints are not satisfied and issues work orders to maintenance personnel, including fault descriptions, recommended handling methods, suggested toolkits, and location information.

3.5.3 Work order closure process

The closure module records actual handling methods, time consumption, and parts replacement. A structured record includes fault description, handling method, timestamp, personnel, and location. NLP-based analysis and assisted input ensure that fault descriptions and handling steps are mapped back into the FDL and FPL, thus closing the loop from process data to knowledge accumulation, as reflected in the cyclic flow in Figure 4.

3.6 Knowledge Graph System Architecture

The knowledge graph system comprises three components: knowledge modeling, graph database implementation, and knowledge maintenance. The overall construction pipeline is shown in Figure 5.

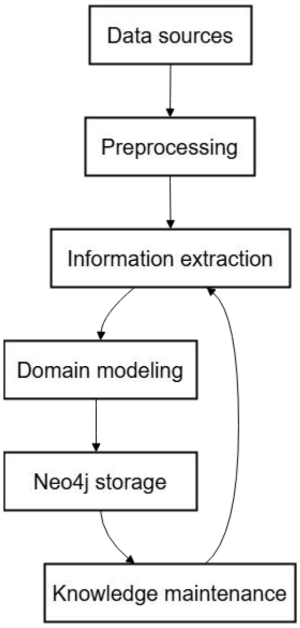

Figure 5 Knowledge Graph Construction Pipeline for Reactor O&M

Data sources include work requests, closure records, and expert experience. After entity and relationship extraction, domain knowledge such as equipment trees, fault modes, and handling methods is modeled and stored in Neo4j. Knowledge maintenance includes creation, update, merging, and versioning of entities and relationships to keep the graph consistent and up to date.

3.7 Intelligent QA System Design

The intelligent QA system consists of three core modules, whose interactions are depicted in Figure 6.

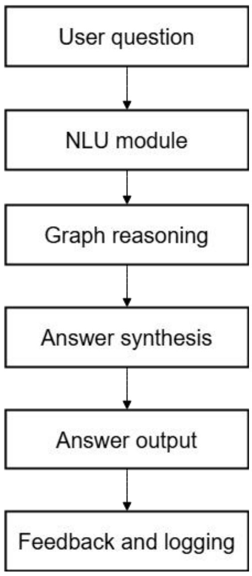

Figure 6 Intelligent QA Workflow based on the Reactor O&M Knowledge Graph

1. **Natural Language Understanding (NLU)**
Word segmentation and keyword extraction; recognition of fault phenomena and equipment references; mapping of text expressions to graph entities.
2. **Graph querying and reasoning**
Based on the graph structure, candidate paths are computed using weighted path search:

$$\mathrm{Shortest}_{\mathrm{Path}} = \arg\min_{\mathrm{path}\in G} \sum_{e\in\mathrm{path}} w_e \,, \tag{5}$$

where $w_e$is the edge weight determined by domain knowledge and historical statistics.
3. **Answer generation and recommendation**
The system outputs inferred fault causes, recommended handling steps, required toolkits, and similar historical cases,

forming a coherent and interpretable knowledge chain for O&M decision-making.

4 SYSTEM IMPLEMENTATION AND APPLICATION CASES

This section describes the engineering implementation of the intelligent QA system, including the digital O&M process management platform, the O&M knowledge base construction, the big data analytics module, the intelligent QA and recommendation functions, and representative application cases. Deployment in a real reactor refueling system validates the effectiveness of the system in terms of fault diagnosis efficiency, knowledge management quality, team capability assessment, and equipment health assessment.

4.1 Implementation of the Digital O&M Process Management Platform

The digital O&M process management platform is the primary data source for the knowledge graph, responsible for the structured acquisition and management of on-site text descriptions, fault handling information, and workflow data. Its functional architecture is consistent with the process shown in Figure 4 and includes O&M business management, fault information preprocessing and semantic parsing, and data storage for fault descriptions, handling methods, and master data.

Ticket creation follows the process described in Section 3.5.1, incorporating NLP-based normalization, automatic coding, and incremental knowledge accumulation. The resulting structured records are ingested into the knowledge graph through the pipeline illustrated in Figure 5.

The dispatch module implements a policy that evaluates equipment location, team shifts, and personnel capabilities, and provides automatic or manual task assignment. The system issues work orders to responsible personnel and attaches recommended handling methods and toolkits, based on graph-based inference paths in the QA workflow (Figure 6).

The closure module records the actual handling process, including time consumption and parts replacement. NLP analysis and assisted input ensure that fault descriptions and handling steps are mapped back into the FDL and FPL, forming a closed-loop data–knowledge cycle as depicted in Figure 4.

4.2 O&M Knowledge Base Construction

The O&M knowledge base is implemented in Neo4j, representing equipment, faults, and handling methods as a semantic graph.

4.2.1 Knowledge modeling structure

The knowledge base includes four core entity categories:

Device entities: system–equipment–component–subcomponent.

Failure entities: types, phenomena, locations.

Solution entities: handling methods.

ExpertRule entities: expert knowledge and decision rules.

Relationship types include equipment hierarchies; fault–equipment associations; fault–location links; mappings between fault phenomena and handling methods; associations between handling methods and toolkits; and connections between solutions and expert rules.

Formally, the knowledge graph can be written as

$$\mathrm{KG} = (V, E), \tag{6}$$

where

$$V = \{\text{Device,Failure,Solution,ExpertRule}\}, \tag{7}$$

and Econtains the corresponding domain-specific relations. The overall modeling is visualized in Figure 5.

4.2.2 Automated construction and expert fusion

The knowledge base supports three construction modes:

1. Automated construction from historical O&M data.
2. Semi-automated construction combining rule-based extraction with human review.
3. Manual construction via expert input of new experiences or special cases.

When newly extracted triples are added, existing relationships are updated if they already exist; otherwise, new triples are inserted. This mechanism ensures that the knowledge base remains compact and free of redundant relationships while steadily evolving with new data.

4.3 O&M Big Data Analytics Module

The system implements functions such as equipment status statistics, health assessment, O&M status evaluation, and team capability assessment. These analyses operate primarily on the light aggregation and subject data zones in Figure 3.

4.3.1 Equipment status statistics

Based on the temporal distribution of equipment faults, a fault frequency model is defined as

$$F_t(d) = \frac{N_t(d)}{\Delta t}, \tag{8}$$

where $N_t(d)$is the number of faults for device $d$in time interval t, and $\Delta t$is the window length (day/week/month). Trend, periodicity, and anomaly analyses can be performed using this indicator.

4.3.2 Equipment health assessment model
A health index (HI) is defined as

$$HI_d = \alpha f_1(d) + \beta f_2(d) + \gamma f_3(d), \tag{9}$$

where $f_1$is the fault frequency factor, $f_2$measures the impact range of faults, $f_3$characterizes historical repair time, and $\alpha, \beta, \gamma$are weights derived from expert experience or model training. The resulting health profile supports maintenance planning and preventive maintenance strategies.

4.3.3 Team capability assessment
Team capability is modeled via the average handling time for fault categories:

$$C_g = \frac{1}{n}\sum_{i=1}^{n} T_{g,i}\,, \tag{10}$$

where $C_g$is the capability index of team g(shorter time indicates higher capability), $T_{g,i}$is the handling time for the i-th fault category handled by team g, and $n$is the number of fault categories considered. This index serves as a basis for performance evaluation and team assignment optimization.

4.3.4 Fault ranking and statistics
Fault types are ranked according to their impact:

$$\mathrm{Rank}(f) = \mathrm{sort}(\lambda_1 N_f + \lambda_2 T_f), \tag{11}$$

where N_f is the number of occurrences of fault type f , T_f is its average handling time, and λ_1, λ_2 are weighting coefficients. Such ranking helps identify fault types that consume the most resources and guides the optimization of O&M strategies.

4.4 Implementation of Intelligent QA and Recommendation

The intelligent QA module is the core innovation of the platform.

4.4.1 Natural language understanding
Given a user query such as
“how should I handle sticking in the refueling gripper arm?”,
the system performs segmentation and entity recognition, identifies the fault phenomenon (“sticking”) and equipment (“gripper arm”), and maps them to nodes in the knowledge graph:

$$q \rightarrow (\text{Failure: “sticking”}) \wedge (\text{Device: “gripper arm”}). \tag{12}$$

4.4.2 Semantic retrieval over the knowledge graph
Graph-based relationship retrieval is executed to find candidate solutions:

$$\mathrm{CandidateSolutions} = \{s_j | (\mathrm{Failure}, s_j) \in E\}, \tag{13}$$

and results are ranked using textual similarity and historical effectiveness. This step is implemented as the retrieval and ranking phase in the QA workflow of Figure 6.

4.4.3 Shortest path reasoning (Dijkstra)
The Dijkstra algorithm is employed to identify the most probable cause and the most appropriate handling method through path search:

$$P^* = \arg\min_{p \in P} \Big(\sum_{e \in p} w_e\Big), \tag{14}$$

where $P$is the set of candidate paths and $w_e$is the edge weight determined by expert scoring or historical statistics [14]. The resulting shortest path provides an interpretable explanation that links fault phenomena, equipment components, and handling solutions.

4.4.4 Answer generation and recommendation
The final output to the user includes:
a) Inferred fault cause;
b) Recommended handling steps;
c) Required toolkits;
d) Similar historical cases;
e) Estimated handling time.
These elements form a coherent and interpretable knowledge chain for O&M decision-making.

4.5 Simulation-Based Validation

4.5.1 Scenario description
To evaluate the proposed system, a simulation scenario is designed based on a reactor refueling context. The simulated environment assumes that refueling equipment operates over extended periods, with certain components (e.g., gripper arms, rotation mechanisms, guide mechanisms) exhibiting relatively high fault frequencies. The scenario also considers complex shift scheduling and knowledge transfer challenges typical of real-world O&M environments. The system implements a full process loop integrating O&M big data analytics, process digitalization, knowledge graphs, and intelligent recommendation.

4.5.2 Case 1: Intelligent recommendation for a simulated fault
In the simulated scenario, a refueling gripper arm is assumed to exhibit motion delay and slight vibration.

System processing:
The NLP module extracts fault phenomena; the FDL matches them to a standardized sticking-type fault; the knowledge graph retrieves handling methods; and Dijkstra's algorithm is applied to identify the shortest reasoning path, as illustrated conceptually in Figure 6.
System output:
a) Likely cause: insufficient joint lubrication leading to increased friction.
b) Recommended handling: lubrication check, removal of foreign objects, and rotation test.
c) Toolkits: lubrication gun, cleaning materials, etc.
d) Similar cases: multiple simulated historical records with similar phenomena, including handling steps and time consumption.
The simulation results suggest that the system has the potential to reduce the time required for diagnosis and improve fault handling efficiency.
4.5.3 Case 2: team capability ranking
In the simulated scenario, the system analyzes the repair times of five hypothetical teams over a given quarter. A simplified summary is presented below:

Table 1 Five Teams Capability Rank

Team	Average handling time	Faults handled	Rank
A	42 min	58	1
B	55 min	62	2
C	60 min	41	3
D	75 min	50	4
E	90 min	39	5

The results in Table 1 demonstrate that the system is capable of providing a basis for optimizing shift scheduling and targeted training in practical applications.
4.5.4 Case 3: equipment health profiling
Based on simulated historical fault and repair data, the system automatically generates a health profile for the refueling equipment, including a health index (e.g., 0.73), a predicted increase in fault probability over a specified future period, and high-frequency fault locations such as rotation joints. These simulation results indicate that the system could potentially support the establishment of preventive maintenance plans and help reduce unplanned downtime.

5 CONCLUSION AND FUTURE WORK

5.1 Conclusion

Focusing on reactor O&M, a domain characterized by high complexity, stringent safety requirements, and strong specialization, this study conducts systematic research on the design and application of a knowledge-graph-based intelligent QA system. The system is intended to address key challenges of traditional O&M, such as fragmented knowledge, experience-based decision-making, low fault response efficiency, and poor knowledge reuse. Based on the real business needs of a reactor refueling system, the system integrates data collection, knowledge modeling, graph databases, digital O&M platforms, and intelligent QA into a practical and extensible solution tailored to the nuclear industry.
First, the paper systematically analyzes the knowledge structures of reactor O&M, including equipment hierarchies, fault characteristics, and the semantic relationships among fault features and handling methods, and proposes a knowledge-graph modeling method suitable for nuclear O&M. By leveraging large amounts of historical records, ticket data, expert experience, and equipment structure trees, a knowledge graph comprising Device, Failure, and Solution entities is constructed, and knowledge is expressed through entity extraction, relationship construction, and rule integration. This method effectively addresses the looseness and poor retrievability of traditional document-based knowledge representations and provides a unified knowledge carrier for intelligent QA.
Second, the paper designs a knowledge storage and logical reasoning system for intelligent QA based on the Neo4j graph database and a rule base. Neo4j offers powerful graph traversal performance, flexible data schemas, and a user-friendly query language, enabling efficient querying of complex equipment topologies and fault chains. The rule base encodes domain logic, expert knowledge, and experience rules so that the QA system can not only “find the right information” but also “use it correctly”. Additionally, by applying the Dijkstra shortest path algorithm to reasoning over the knowledge graph, the system provides path-level explanations of fault causes, associated components, and potential handling solutions, significantly improving the interpretability and accuracy of intelligent recommendations.
Third, the paper builds a digital O&M process information platform that fully structures workflows such as work request creation, transfer, dispatch, and closure, enabling complete data collection, standardized representation, and real -time management (Figure 4). Meanwhile, the system uses NLP techniques to process free-text descriptions, which are integrated into the FDL and FPL for continuous expansion, thereby realizing a dynamic cycle of “data–knowledge–QA” and supporting the continuous evolution of the knowledge graph.
Finally, the system is validated through simulation in a reactor refueling scenario. The results indicate that the intelligent QA system shows promising potential in improving fault diagnosis efficiency, reducing maintenance

response time, enhancing the granularity of team capability assessment, and supporting trend prediction of equipment health status. By integrating historical O&M data, typical cases, and expert experience, the system is expected to facilitate sustainable accumulation, visual presentation, and intelligent use of knowledge, thereby contributing to improved reliability and standardization of O&M practices.

In summary, the proposed intelligent QA system achieves an integration of knowledge graphs, NLP, and path reasoning techniques, and demonstrates potential engineering value in the simulation study. It offers a feasible and reference-worthy approach for improving O&M efficiency, strengthening knowledge management, and supporting digital transformation in the nuclear industry.

5.2 Future Work

Despite encouraging results, there remains room for further enhancement. Future research directions include:

5.2.1 *Incorporating large language models to enhance semantic understanding*

The current system relies primarily on traditional NLP and lexicon-based semantic matching, which show limitations in handling complex sentences, implicit semantics, and cross-sentence reasoning. Future work may explore deploying lightweight or distilled LLMs in intranet environments and combining knowledge graphs with LLMs to achieve more robust natural-language understanding, especially for ambiguous descriptions and long texts.

1. Automated knowledge extraction and continual learning

Knowledge base updates still require human review. Future research can strengthen automated knowledge extraction techniques such as relation extraction, event extraction, and incremental learning, enabling continuous extraction of high-quality knowledge from new textual records so that the knowledge graph can self-update, self-improve, and self-correct.

2. Supporting cross-system and multimodal data fusion

Nuclear O&M involves substantial multimodal data, including drawings, video monitoring, vibration signals, and telemetry. Future work may incorporate computer vision and time-series analysis to integrate multimodal data into the knowledge graph and realize multimodal QA and reasoning over complex scenarios.

3. Developing quantitative knowledge credibility evaluation

Since O&M knowledge arises from multiple sources (expert experience, historical cases, automatically extracted content), future studies can design knowledge quality scoring models such as

$$\text{Credibility} = \alpha \cdot \text{support count} + \beta \cdot \text{expert authority} + \gamma \cdot \text{historical success rate}, \tag{15}$$

to sort answers based on knowledge credibility and improve the reliability of O&M decision-making.

4. Building cross-reactor and cross-facility nuclear O&M knowledge graphs

Under appropriate security conditions, future work may explore knowledge transfer and sharing across different reactors and refueling devices and construct a higher-level multi-site nuclear O&M knowledge graph. By defining unified modeling methods and interoperable graph interfaces, knowledge resources can be widely reused across facilities.

COMPETING INTERESTS

The authors have no relevant financial or non-financial interests to disclose.

FUNDING

This work is supported by the Talent Fund Project No. 25816 of China Institute of Atomic Energy (CIAE).

REFERENCES

[1] Ji S, Pan S, Cambria E, et al. A survey on knowledge graphs: Representation, acquisition and applications. IEEE Transactions on Neural Networks and Learning Systems, 2022, 33(2): 494-514.

[2] Chen J, Zhang H, Wang H. Knowledge graph and its application in intelligent question answering: A survey. In: 2020 IEEE International Conference on Big Data (Big Data), 2020: 2273-2282.

[3] Paulheim H. Knowledge graph refinement: A survey of approaches and evaluation methods. Semantic Web, 2017, 8(3): 489-508.

[4] Wang X, Pan W, Gao X. Application of knowledge graph technology in power equipment operation and maintenance. In: 2021 IEEE International Conference on Power System Technology (POWERCON), 2021: 338-344.

[5] Zhang CY. Research and implementation of automatic question answering system based on knowledge graph. Beijing University of Posts and Telecommunications, 2018.

[6] Xiong W, Yu M, Chang S, et al. One-shot relational learning for knowledge graphs. In: Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL), 2018: 1980-1990.

[7] Robinson I, Webber J, Eifrem E. Graph Databases. 2nd ed. O'Reilly Media, 2015.

[8] Miller JJ. Graph database applications and concepts with Neo4j. In: Proceedings of the Southern Association for Information Systems Conference, 2013: 1-6.

[9] Malewicz G, Austern MH, Bik AJ, et al. Pregel: A system for large-scale graph processing. In: Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data, 2010: 135-146.

[10] Low Y, Bickson D, Gonzalez J, et al. Distributed GraphLab: A framework for machine learning and data mining in the cloud. VLDB Endowment, 2012, 5(8): 716-727.

[11] Kyrola A, Belli L, Guestrin C. GraphChi: Large-scale graph computation on just a PC. In: 10th USENIX Symposium on Operating Systems Design and Implementation (OSDI), 2012: 31-46.

[12] Shao B, Wang H, Li Y. Trinity: A distributed graph engine on a memory cloud. In: Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data, 2013: 505-516.

[13] Sarwat M, Elnikety S, He Y, et al. Horton: An online query execution engine for large distributed graphs. In: 2012 IEEE 28th International Conference on Data Engineering (ICDE), 2012: 1289-1292.

[14] Zhang F, Liu J, Li Q. A shortest path optimization algorithm based on Dijkstra. Remote Sensing Information, 2004(2): 4-7.

[15] Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space. In: International Conference on Learning Representations (ICLR), 2013.