Journal of Computer Science and Electrical Engineering
Print ISSN: 2663-1938
Online ISSN: 2663-1946
DOI: https://doi.org/10.61784/jcsee3105

GRAPH-RAG ENHANCED RETRIEVAL AND MULTI-AGENT COLLABORATIVE METHOD FOR INTELLIGENT NUCLEAR POWER ELECTRICAL DESIGN

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

Abstract: The design of nuclear power plant electrical systems is characterized by an extensive and intricate body of standards, strong cross-disciplinary coupling, and stringent safety and compliance requirements. Traditional digital design workflows face significant limitations in knowledge reuse, intelligent decision support, and quality assurance. At the same time, general-purpose large language models (LLMs) exhibit hallucinations and lack reliability in safety-critical engineering domains. To address these challenges, this paper proposes an intelligent nuclear power electrical design methodology that integrates knowledge-graph-enhanced retrieval-augmented generation (Graph-RAG) with a multi-agent system (MAS).
First, a domain ontology covering IEC/GB standards, equipment parameter libraries, and design rules is constructed to provide a structured knowledge backbone. Second, a graph-guided hybrid retrieval strategy that combines semantic retrieval with graph path reasoning is designed to enhance retrieval accuracy and contextual relevance. Third, a multi-agent collaboration architecture—comprising requirement analysis, intelligent recommendation, automatic document generation, consistency checking, format review, and interactive question answering agents—is developed. Task orchestration is implemented through a directed acyclic graph (DAG), enabling parallel and coordinated execution of complex design workflows.
Theoretical analysis and scenario-based validation indicate that the proposed approach substantially reduces hallucination rates, improves design efficiency, and ensures the standard compliance of design outputs. Each design decision becomes traceable to explicit standard clauses and equipment parameters, demonstrating the feasibility of the method for the intelligent transformation of nuclear power electrical design and its potential transferability to other complex engineering systems.
Keywords: Nuclear power electrical design; Knowledge graph; Retrieval-augmented generation; Multi-agent systems; Intelligent engineering

1 INTRODUCTION

1.1 Research Background

Nuclear power plant electrical systems constitute a critical component of the overall safety architecture of a nuclear power station. Their design quality directly affects defense-in-depth capability, operational reliability, and controllability under abnormal and accident conditions. With the continuous expansion of nuclear power construction in China, the complexity of safety-related system design has significantly increased. The depth of specialization, knowledge density, and cross-disciplinary coupling in nuclear electrical design far exceed those of conventional industrial power engineering.
According to recent industry statistics, by the end of 2024 China had 55 nuclear power units in operation, and the number of units under construction had ranked first in the world for many consecutive years. The deployment of third-generation reactor technologies such as Hualong One, CAP1400, and EPR has further enlarged the design space, increased equipment parameter complexity, and made technical standards more fine-grained and strongly interconnected across disciplines.
In current practice, nuclear power electrical design still relies heavily on manual consultation of standards, ad hoc retrieval of historical design documents, and item-by-item engineering calculations and compliance checks. Given the extremely stringent safety and regulatory requirements of nuclear power projects, any design deviation may propagate into major risks during subsequent construction and operation stages. Design documentation must therefore be traceable, internally consistent, and highly accurate.
In parallel, the Chinese government has launched an “AI+” national strategy, explicitly calling for the deep integration of advanced AI technologies—including large models—into key industries such as energy, power, and nuclear power. Policy documents such as the *AI+ Energy Implementation Guidelines* emphasize the use of AI models throughout the lifecycle of nuclear power design, manufacturing, and construction, and advocate the development of intelligent nuclear power plants. This provides an unprecedented opportunity for the intelligent upgrading of nuclear electrical design.
However, intelligent nuclear electrical design is far from a simple matter of applying general-purpose LLMs to domain-specific tasks. While LLMs perform well on general language understanding and open-domain question answering,

their applicability in high-safety, highly regulated, and highly complex scenarios such as nuclear power is severely limited, especially in terms of controllability, traceability, and compliance [1].

1.2 Key Challenges in Nuclear Power Electrical Design

Nuclear power electrical design is a prototypical knowledge-intensive engineering activity. Its complexity is reflected in multiple dimensions:

1.2.1 Extensive standards and complex constraints

Nuclear electrical design must simultaneously comply with international standards (IEC series), national standards (GB series), industry standards (NB/DL series), and nuclear safety regulations (HAF series). A single nuclear project involves a large number of mandatory and recommended standards, each containing numerous clauses. These clauses define intricate constraint relationships involving equipment types, safety classes, environmental conditions, EMC requirements, test conditions, and documentation rules.

1.2.2 Difficult knowledge reuse and long experience transfer cycles

Nuclear electrical design is highly dependent on engineers' domain expertise and accumulated project experience. It typically takes many years of training before an engineer can independently undertake nuclear electrical design tasks. This apprenticeship-based knowledge transfer leads to low efficiency, high dependence on key individuals, and difficulty in systematically reusing design knowledge across projects.

1.2.3 Limited applicability of general-purpose LLMs in specialized domains

Studies have shown that general-purpose LLMs exhibit a relatively high factual error rate in specialized engineering tasks and often generate hallucinations. In the nuclear domain, directly querying an LLM about standards may result in non-existent standard identifiers, incorrect clause references, or misinterpretation of technical requirements, which is unacceptable in safety-critical design.

1.3 Related Work

1.3.1 Retrieval-Augmented Generation (RAG)

Lewis et al. proposed retrieval-augmented generation (RAG), which integrates external knowledge retrieval into the generation process and has proven effective for knowledge-intensive natural language processing tasks [2]. Subsequent surveys and extensions have systematically analyzed RAG architectures and demonstrated their ability to mitigate hallucinations and improve factuality in large language models [3].

1.3.2 Knowledge graphs and semantic retrieval

Knowledge graphs provide structured representations of entities and relations, enabling more precise retrieval and logical reasoning over standards, technical manuals, and historical cases. Recent work on graph retrieval-augmented generation (Graph-RAG) shows that incorporating graph-based reasoning into RAG pipelines significantly enhances retrieval accuracy and multi-hop reasoning capability [4].

1.3.3 Multi-Agent Systems (MAS)

LLM-based multi-agent systems decompose complex tasks into collaborative workflows executed by specialized agents. Recent surveys highlight the potential of MAS in workflow orchestration, infrastructure design, and complex task execution, while also pointing out challenges in stability, communication, and governance [5]. Buehler, for example, combines ontologic graphs, RAG, and multi-agent strategies for interpretable materials design, illustrating the value of integrating knowledge graphs, retrieval, and agent collaboration in engineering design pipelines [6].

1.3.4 AI applications in nuclear engineering

Comprehensive reviews indicate that AI has been increasingly applied to reactor analysis, fault diagnosis, safety assessment, and other nuclear engineering areas, and underscore the importance of interpretability and safety in such applications [7]. Domain-specific knowledge graphs have also been used to design safety review decision-support systems for nuclear power plants [8].

These advances collectively motivate the integration of knowledge-graph-enhanced RAG with MAS for nuclear electrical design.

1.4 Research Objectives and Main Contributions

The central objective of this study is to construct an intelligent nuclear power electrical design methodology that integrates domain knowledge, logical reasoning, and parallel collaboration, thereby enabling end-to-end automation from requirement analysis and scheme generation to document delivery.

The main contributions of this work are as follows:

1. **Domain knowledge graph construction:** We propose a knowledge graph construction method tailored to the nuclear electrical domain and build a knowledge backbone covering standards, equipment parameters, and design rules.
2. **Graph-RAG architecture:** We design a graph-guided hybrid retrieval-augmented generation (Graph-RAG) architecture that combines the statistical matching capability of traditional RAG with the logical reasoning capability of knowledge graphs, improving retrieval accuracy and generation quality.
3. **Task-driven multi-agent collaborative design system:** We construct a task-driven multi-agent system (MAS) and implement DAG-based task decomposition and dynamic scheduling, enabling parallel and coordinated execution of complex design workflows.

4. **Method validation in typical engineering scenarios:** We validate the proposed method using representative nuclear electrical design scenarios and demonstrate its advantages in knowledge reliability, design traceability, collaboration efficiency, and quality assurance.

2 INTELLIGENT NUCLEAR ELECTRICAL DESIGN FRAMEWORK

2.1 Overall System Architecture

Nuclear power electrical design is an engineering process that is **constrained by standards, driven by rules, centered on documents, and organized around workflows**. Design activities span the full lifecycle from requirement definition, equipment selection, and engineering calculations to standard compliance checking and document generation.
To address these characteristics, we propose a four-layer integrated architecture consisting of:
1. A **User Interaction Layer**,
2. An **Intelligent Orchestration Layer**,
3. A **Multi-Agent Execution Layer**, and
4. A **Knowledge Enhancement Layer**.
The overall system architecture is illustrated in Figure 1, which shows how user intent flows through orchestration, agents, and knowledge services to yield standardized design documents.

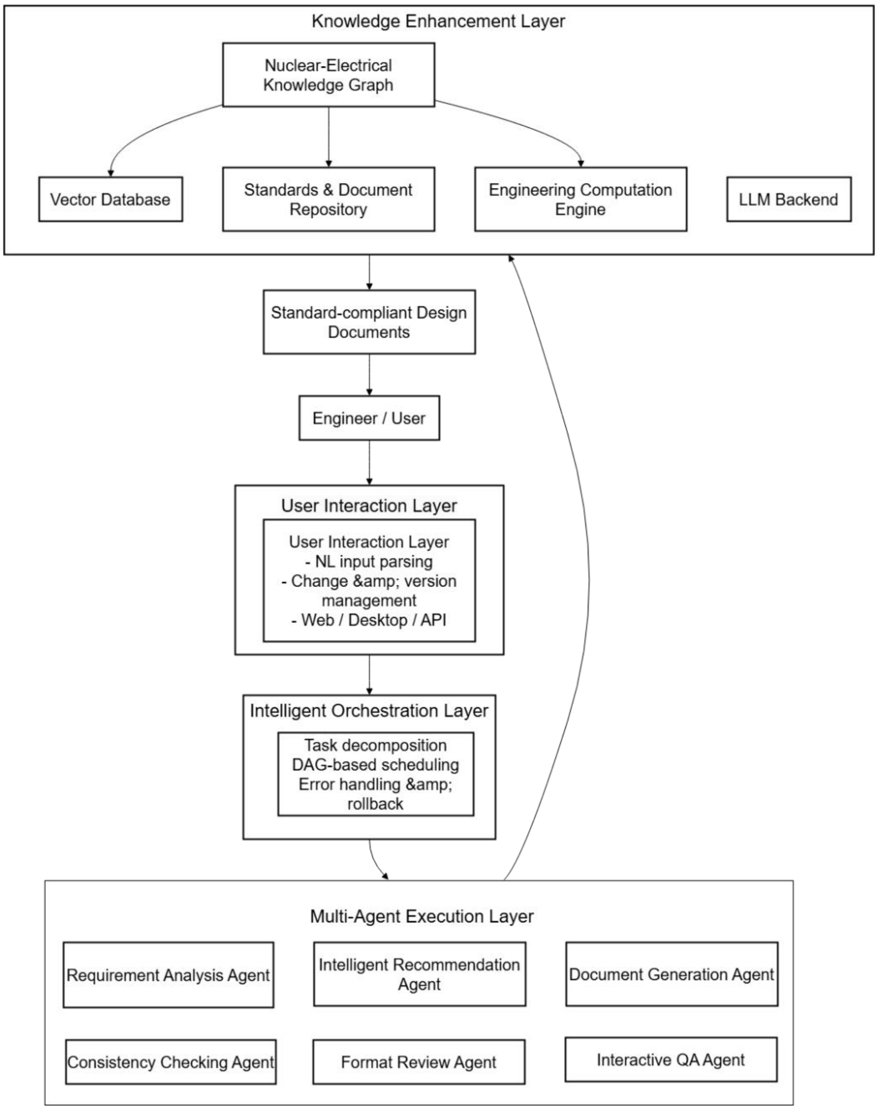

Figure 1 Overall Four-Layer Architecture of the Proposed Intelligent Nuclear Electrical Design System

2.1.1 User interaction layer
The user interaction layer serves as the interface between engineers and the system. It is responsible for:
• Transforming natural-language engineering intent into structured, machine-interpretable task descriptions;
• Supporting design change management, version control, and review comment handling;
• Providing web-based, desktop, and API interfaces for enterprise-wide collaborative design.

Its core objective is to ensure that user intent is accurately captured and represented, providing a clear context for subsequent intelligent design processes.

2.1.2 Intelligent orchestration layer

The intelligent orchestration layer acts as the system's central controller. By decomposing, scheduling, and coordinating tasks, it manages the overall multi-agent collaboration process.

Key mechanisms include:

• **Task decomposition:** Breaking down complex nuclear electrical design problems into manageable subtasks such as parameter acquisition, equipment selection, calculation checking, and document generation;

• **Dependency modeling:** Representing task dependencies using a directed acyclic graph (DAG);

• **Topological scheduling:** Executing tasks in an order that strictly respects logical and regulatory constraints;

• **Asynchronous messaging:** Enabling robust communication and data sharing among agents;

• **Exception handling and rollback:** Supporting error diagnosis, failure handling, and automatic retry when checks fail.

An illustrative example of the DAG-based task orchestration is shown in **Figure 2**, where nodes represent agents or computation modules and edges encode dependency and data flow relationships.

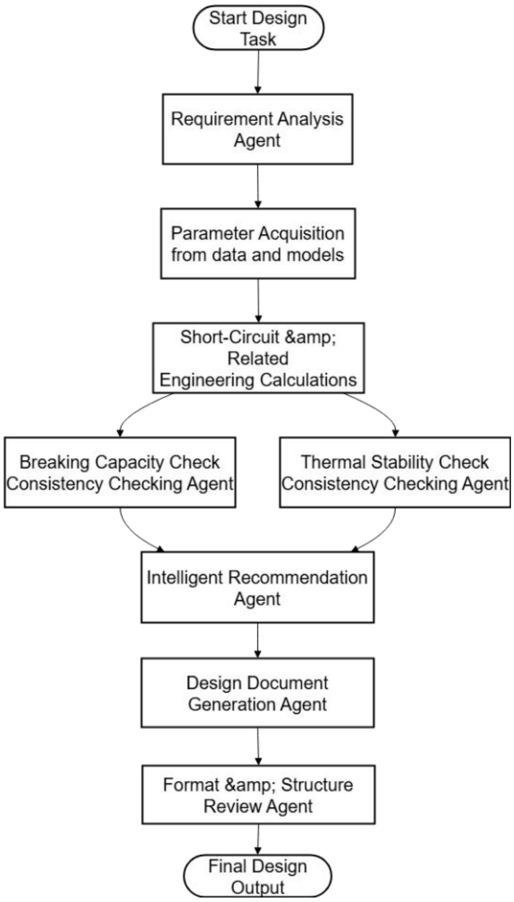

Figure 2 DAG-Based Task Orchestration for Nuclear Electrical Design

2.1.3 Multi-agent execution layer

The multi-agent execution layer consists of six specialized agents, each bound to specific knowledge subgraphs, business rules, and document templates, forming a role system with clear responsibilities and boundaries:

• **Requirement Analysis Agent:** Performs requirement structuring, constraint identification, and determination of relevant standards;

• **Intelligent Recommendation Agent:** Provides equipment selection and scheme recommendations based on standard constraints and calculation results;

• **Automatic Document Generation Agent:** Generates design documents, technical specifications, and calculation notes;

• **Consistency Checking Agent:** Validates the consistency and correctness of parameters, equipment selections, and standard clause references;

• **Format Review Agent:** Checks document structure and formatting against nuclear industry documentation norms;

• **Interactive QA Agent:** Provides explanations of design decisions, interpretations of standards, and real-time question answering.

This multi-agent architecture transforms nuclear electrical design from "manual division of labor" to "intelligent division of labor," resembling a digitalized in-house engineering team.

2.1.4 Knowledge enhancement layer

The knowledge enhancement layer provides the machine-understandable, retrievable, and inferable knowledge base needed by RAG, agents, and document generation. Its core components include:

• A nuclear electrical knowledge graph (standards, clauses, equipment parameters, design rules);
• A vector database for semantic retrieval;
• A document repository containing standards, drawings, and technical manuals;
• An engineering computation engine (short-circuit current, thermal stability, cable ampacity, etc.);
• Large language models for generation, reasoning, and dialogue.

Graph-RAG retrieval, agent reasoning, and document generation all rely on this layer. The end-to-end intelligent design workflow across these layers is depicted in **Figure 3**.

2.2 Formal Modeling of Nuclear Electrical Design Tasks

To make the design process formal, computable, and verifiable, we model the nuclear electrical design task as a mathematical object.

A design task is defined as a quintuple:

$$T = (R, C, K, O, D) \tag{1}$$

where:

• R(Requirements): a set of requirements, including functional requirements, performance metrics, and environmental conditions;
• C(Constraints): a set of constraints, including hard constraints (e.g., core safety clauses) and soft constraints (e.g., recommended clauses);
• K(Knowledge): a knowledge base containing standards, equipment data, formulas, and engineering cases;
• O(Objectives): design objectives such as maximizing safety margin, balancing cost and performance, and maximizing reliability;
• D(Design Outputs): outputs including selection results, calculation tables, and technical documents.

Example: short-circuit breaking capacity constraint

For a circuit breaker, the breaking capacity constraint can be expressed as:

$$c_1 : I_{cu} \geq k \cdot I_k \tag{2}$$

where:

• I_{cu}: rated short-circuit breaking capacity;
• I_k: expected short-circuit current at the installation point;
• k: safety factor defined in IEC/GB standards.

Example: seismic qualification requirement for safety-class equipment

For safety-class equipment, a typical constraint is:

$$c_2 : \mathrm{SafetyClass}(e) = 1E \Rightarrow \mathrm{SeismicQualified}(e) = \mathrm{True} \tag{3}$$

This formalization provides the mathematical basis for agent reasoning, computation, and checking, and it underpins the verifiability and traceability of automation.

2.3 Intelligent Design Workflow

Starting from user design intent, the intelligent design workflow proceeds through requirement structuring, knowledge retrieval, engineering calculations, multi-agent collaboration, and standardized document generation, forming a complete and closed-loop design process. The overall workflow and its main stages are illustrated in **Figure 3**.

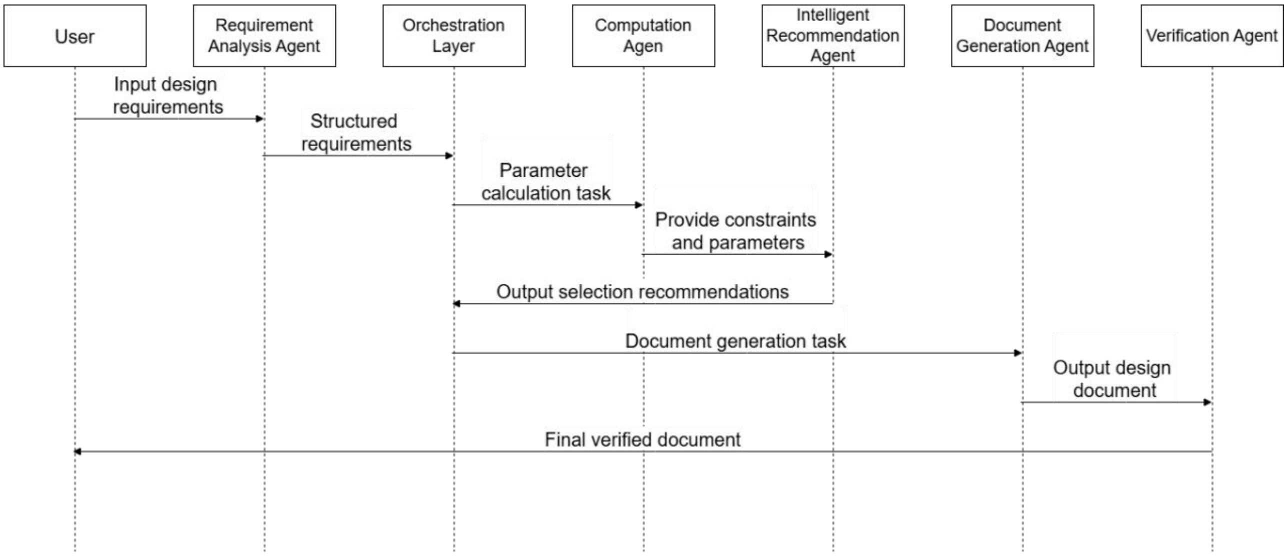

Figure 3 End-to-end Intelligent Workflow from Engineering Intent to Standardized Design Documents

From a technical perspective, the workflow exhibits the following core characteristics:

2.3.1 Automatic mapping from intent to structured tasks

The Requirement Analysis Agent semantically parses natural-language user input and, based on the domain ontology and constraint models in the knowledge graph, decomposes complex requirements into:

- Design objectives;
- Functional requirements;
- Performance metrics;
- Safety constraints.

This enables the system to interpret design instructions as "engineering task units," ensuring that downstream steps are well-defined and executable.

2.3.2 Graph-RAG as the core knowledge support mechanism

During design execution, the system does not rely solely on the internal parameters of an LLM. Instead, it uses a Graph-RAG retrieval module to:

- Extract authoritative constraint conditions from standard clauses;
- Identify equipment entities, parameter relations, and multi-hop reasoning chains from the knowledge graph;
- Retrieve relevant cases and historical design rationales from the document corpus.

The retrieved knowledge is organized into **evidence chains** and passed along with task context to the agents. This evidence-based paradigm effectively mitigates hallucinations and ensures that agents operate under a rigorous standard framework.

2.3.3 Collaborative multi-agent processing for parallelism and division of labor

After task decomposition, the Intelligent Orchestration Layer constructs a DAG over subtasks, enabling:

- **Serial tasks** that must follow a strict design order (e.g., short-circuit calculation → breaking capacity checking);
- **Parallel tasks** that are independent (e.g., short-circuit calculations at multiple buses);
- **Iterative tasks** that require re-generation (e.g., if a consistency check fails, control is passed back to a generation agent for revision).

Each agent has clearly defined boundaries, dedicated knowledge subgraphs and rule sets, and structured communication channels, resembling a well-organized digital design team.

2.3.4 Automation of engineering computation and standard checking

The computation-related agents automatically perform engineering calculations such as short-circuit currents, thermal stability checks, and cable ampacity assessments by calling:

- Formula libraries;
- Parameter models;
- Typical equipment values;
- Standard constraints encoded in the knowledge graph.

The results are then validated by the Consistency Checking Agent, ensuring that outputs can be traced back to specific standard clauses and explicit calculation steps.

2.3.5 Structured and standardized document generation

The Automatic Document Generation Agent uses:

- Template libraries of technical specifications;
- Document formatting standards;
- Evidence chains and intermediate calculation results,

to generate comprehensive design documents, including:
• Design descriptions;
• Technical specifications;
• Equipment lists;
• Calculation notes.
After processing by the Format Review Agent, documents are standardized in layout, chapter numbering, and clause referencing, satisfying formal nuclear industry delivery requirements.
2.3.6 Traceable, verifiable, and explainable design loop
Each step in the workflow produces:
• Execution logs;
• Retrieval records;
• Lists of referenced standard clauses;
• Data version identifiers.
This yields a highly transparent and auditable design process, providing strong support for safety reviews and regulatory inspections.

3 KEY TECHNICAL METHODS

3.1 Construction of a Nuclear Power Electrical Knowledge Graph

The nuclear power electrical knowledge graph (Nuclear-Electrical KG) serves as the core knowledge backbone of the system. It provides the structured data foundation for Graph-RAG retrieval, agent reasoning, and design checking. The construction process consists of four stages: ontology design, knowledge extraction, knowledge fusion, and quality assessment.
3.1.1 Ontology design
To make nuclear electrical domain knowledge machine-understandable, retrievable, and inferable, we adopt the Web Ontology Language (OWL) to build a three-layer ontology comprising an **upper ontology**, a **domain ontology**, and an **application ontology**. This layered structure balances conceptual abstraction with engineering semantics and provides a unified schema for knowledge fusion, relation reasoning, and Graph-RAG retrieval. The overall ontology schema is illustrated in **Figure 4**.

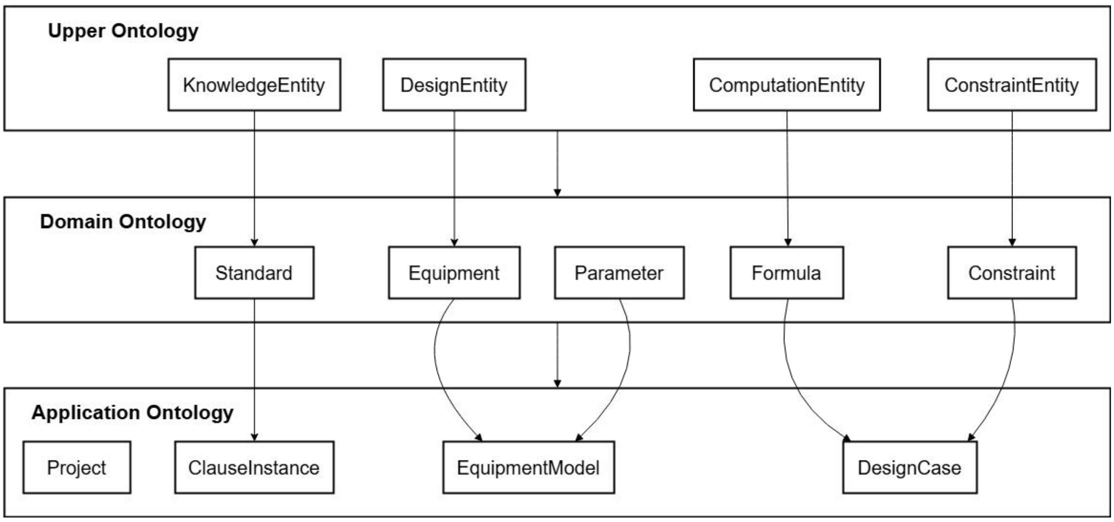

Figure 4 Three-Layer Ontology Schema for the Nuclear Electrical Knowledge Graph

(1) Upper ontology
The upper ontology captures the most general and abstract semantics of engineering design and serves as the foundational structure of the knowledge graph. It defines cross-domain shared classes and relations to ensure semantic consistency across heterogeneous sources.
Core classes include:
• **DesignEntity:** abstract representation of all design objects, such as equipment, systems, and components;
• **KnowledgeEntity:** representation of knowledge sources such as standards, clauses, and documents;
• **ConstraintEntity:** representation of engineering constraints, including safety classes, environmental conditions, and installation requirements;
• **ComputationEntity:** representation of computational procedures, formulas, input/output parameters, and related mathematical structures.
(2) Domain ontology

The domain ontology provides structured modeling of professional knowledge specific to nuclear electrical design. It defines entities, attributes, and constraints in engineering terms to ensure rigor and domain specificity.

Key modeling objects include:

a) **Standard classes:** covering frequently used mandatory and recommended standards, such as IEC international standards, GB national standards, DL power industry standards, NB nuclear standards, and HAF nuclear safety regulations. These are linked to equipment and constraints via relations such as *refersTo*, *constrains*, and *appliesTo*.

b) **Equipment classes:** including major electrical equipment such as circuit breakers, transformers, cables, and protection devices, each with attributes such as rated current, impedance, breaking capacity, insulation level, and environmental category.

c) **Parameter classes:** describing key variables in design calculations, e.g., rated current, impedance, breaking capacity, and capacity, which are associated with formulas and equipment via *hasValue*, *hasUnit*, and *usedInFormula*.

d) **Formula classes:** representing short-circuit current, cable ampacity, voltage drop, and other engineering formulas, with detailed definitions of input/output parameters, applicability conditions, and source standards.

e) **Constraint classes:** encoding typical nuclear engineering constraints such as safety class 1E requirements, seismic categories, environmental qualification categories (e.g., K1/K2/K3), temperature rise limits, fire protection requirements, and EMC constraints.

These entities and relations provide the logical basis for automatic checking and decision support.

(3) Application ontology

The application ontology instantiates the ontology using concrete engineering projects, plant types, and actual standard clauses, forming an "engineering knowledge network" suitable for project-level reasoning.

Its main objectives are to:

- Recognize specific equipment types and models used in real projects;
- Capture dependency relations among plant subsystems and components;
- Support cross-clause semantic reasoning;
- Provide traceable knowledge chains linking design data to specific standard clauses.

Key contents include:

- Instantiated standard clauses (e.g., specific articles from HAF102-2016);
- Equipment models with bound parameter values;
- Historical project design cases;
- Cross-references among clauses;
- Complete logical chains from equipment entities to parameters, formulas, and standards.

3.1.2 Knowledge extraction

Knowledge sources include standard texts, equipment manuals, engineering documents, and historical project records. We adopt a hybrid strategy combining deep learning, rule-based extraction, and manual validation.

- **Entity recognition (NER):** Domain-adapted BERT-based NER models are trained to identify entities such as equipment names, standard numbers, parameter names, and environmental categories.
- **Relation extraction:** Datasets are built using distant supervision supplemented by manual correction. Relations such as equipment-parameter, standard-clause, and clause-constraint are extracted, and multi-hop reasoning chains (e.g., standard → equipment type → parameters) are supported.
- **Attribute extraction:** Structured documents are parsed using rule-based methods, whereas unstructured texts are processed with sequence labeling models such as BiLSTM and Transformer architectures.

3.1.3 Knowledge fusion and conflict resolution

To resolve conflicts across heterogeneous sources such as different standards and equipment manuals, we employ:

- **Entity alignment:** based on vector embeddings and string similarity features;
- **Relation normalization:** mapping extracted relations onto ontology-defined schema;
- **Conflict resolution:** using an authority-weighted strategy (national standards > industry standards > corporate standards);
- **Recency preference:** prioritizing the latest versions of standards.

3.1.4 Knowledge graph quality assessment

The knowledge graph is evaluated along four dimensions:

- **Completeness:** coverage of key entities, relations, and attributes in the domain;
- **Accuracy:** correctness of extracted entities, relations, and attributes;
- **Consistency:** logical and semantic coherence under the ontology schema;
- **Freshness:** timeliness of standard versions and equipment data.

This evaluation ensures that the knowledge graph is reliable and suitable for safety-critical design tasks.

3.2 Graph-Guided Hybrid Retrieval-Augmented Generation (Graph-RAG)

Graph-RAG is a core innovation of this study. By integrating a structured knowledge graph with vector-based retrieval, it provides high-precision knowledge recall and enables evidence-based generation. Recent surveys have shown that such graph-enhanced RAG approaches can significantly improve retrieval quality and interpretability [9-12].

3.2.1 Three-path hybrid retrieval

We adopt a three-path hybrid retrieval strategy:
• Path 1: Semantic vector retrieval
Queries and documents are encoded into dense vectors. This is suitable for fuzzy semantic matching and long texts, and it provides high recall.
• Path 2: Knowledge graph retrieval
Natural-language queries are transformed into SPARQL or graph queries. This path supports multi-hop relational reasoning and can extract logically coherent knowledge chains from the graph.
• Path 3: BM25 keyword retrieval
BM25 is used for precise matching of standard numbers, technical terms, and formulas, and offers robustness and speed for structured engineering documents.

3.2.2 Retrieval result fusion

We use a multi-factor scoring function:

$$\mathrm{Score}_{\mathrm{final}} = w_1 S_{\mathrm{rel}} + w_2 S_{\mathrm{auth}} + w_3 S_{\mathrm{fresh}} + w_4 S_{\mathrm{cite}} \quad (4)$$

where:

• S_{rel}: semantic relevance score;
• S_{auth}: authority score (prioritizing higher-level standards and regulations);
• S_{fresh}: freshness score (favoring newer versions);
• S_{cite}: citation frequency (frequency of usage in historical designs).

A cross-encoder-based re-ranking model is then applied to the candidate results, and the top 5-10 passages are selected as high-confidence evidence for downstream agents.

3.2.3 Context engineering and evidence chain construction

To avoid hallucinations, all retrieved evidence is passed to the LLM in the form of **evidence chains**, which include:
• Graph reasoning paths;
• Original standard clause texts;
• Parameter provenance information;
• Conflict annotations and resolution results.

This approach ensures that each generated fragment can be traced back to specific standard clauses and equipment data. It aligns with recent work on knowledge-graph-enhanced RAG and graph-enhanced agent architectures.

3.3 Multi-Agent Collaboration based on Large Language Models

3.3.1 Agent role definition and boundary control

To ensure that agents are controllable, explainable, and auditable, each agent is bound to:
• A specific knowledge subgraph;
• A subset of standard clauses;
• Dedicated document templates;
• A business rule set;
• A defined communication protocol.

This design ensures that each agent's outputs can be checked and validated against its responsibilities, and that errors can be localized.

3.3.2 DAG-based orchestration and scheduling

The system uses a DAG as the backbone of task orchestration:
• **Serial tasks:** tasks that must follow a strict order, such as "short-circuit calculation → breaking capacity check";
• **Parallel tasks:** independent tasks that can be executed concurrently, such as short-circuit calculations at multiple buses;
• **Iterative tasks:** tasks that must be repeated if checks fail, such as "consistency check fails → regenerate design parameters."

An example of DAG-based agent orchestration is shown in **Figure 2**, and a more detailed view of the execution process is illustrated in **Figure 6** during the validation discussion.

3.3.3 Inter-agent communication

We define four main types of inter-agent messages:
• Task assignment messages;
• Intermediate data exchange messages;
• Status update messages;
• Conflict negotiation messages.

These communication patterns underpin stable and interpretable collaboration among agents and ensure the consistency of shared context.

4 METHOD VALIDATION AND ANALYSIS

To verify the feasibility and effectiveness of the proposed Graph-RAG + MAS method in nuclear electrical design, we perform a descriptive validation based on typical engineering tasks, system behavior, key technical performance, and practical engineering value. Although a full-scale quantitative evaluation is beyond the scope of this work, simulated task execution, clause-level comparison, and expert interviews provide a systematic basis for assessing the method's applicability and deployment potential.

4.1 Overall Validation Approach

In nuclear electrical engineering, system validation cannot rely solely on simple numerical indicators; instead, it emphasizes:
• The completeness of the design workflow;
• The accuracy and appropriateness of knowledge usage;
• The rationality of logical chains;
• The structural and stylistic quality of document outputs.
Our validation approach includes:
• Selecting representative nuclear electrical design tasks;
• Preparing a corpus of standards, equipment data, and engineering documents;
• Simulating multi-round design execution under the proposed system;
• Inviting senior engineers to evaluate the design results.
The overall validation procedure—from task definition and corpus construction to system execution and expert evaluation—is summarized in **Figure 5**.

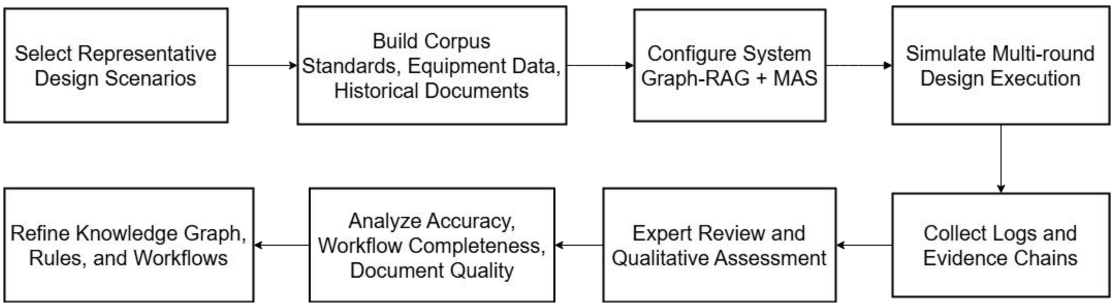

Figure 5 Validation procedure for the proposed Graph-RAG + MAS method

4.2 Validation Scenarios

To cover both **calculation-oriented tasks** and **document-oriented tasks**, we select two typical scenarios.
4.2.1 Scenario 1: Low-voltage circuit breaker selection
This scenario captures the characteristic features of nuclear design tasks involving chained logical reasoning, standard referencing, and engineering computation. The task involves:
• Parsing load parameters;
• Selecting and applying short-circuit calculation methods;
• Checking breaking capacity and thermal stability;
• Identifying safety class (e.g., 1E and environmental categories);
• Recommending equipment models under relevant constraints;
• Automatically generating a draft technical specification.
This scenario tests:
• The accuracy of Graph-RAG in retrieving relevant standard clauses;
• The traceability and correctness of the computation process;
• The quality of inter-agent collaboration;
• The completeness and compliance of the generated document.
An example of the DAG execution process for this scenario, including agent interactions and data flows, is illustrated in **Figure 6**.

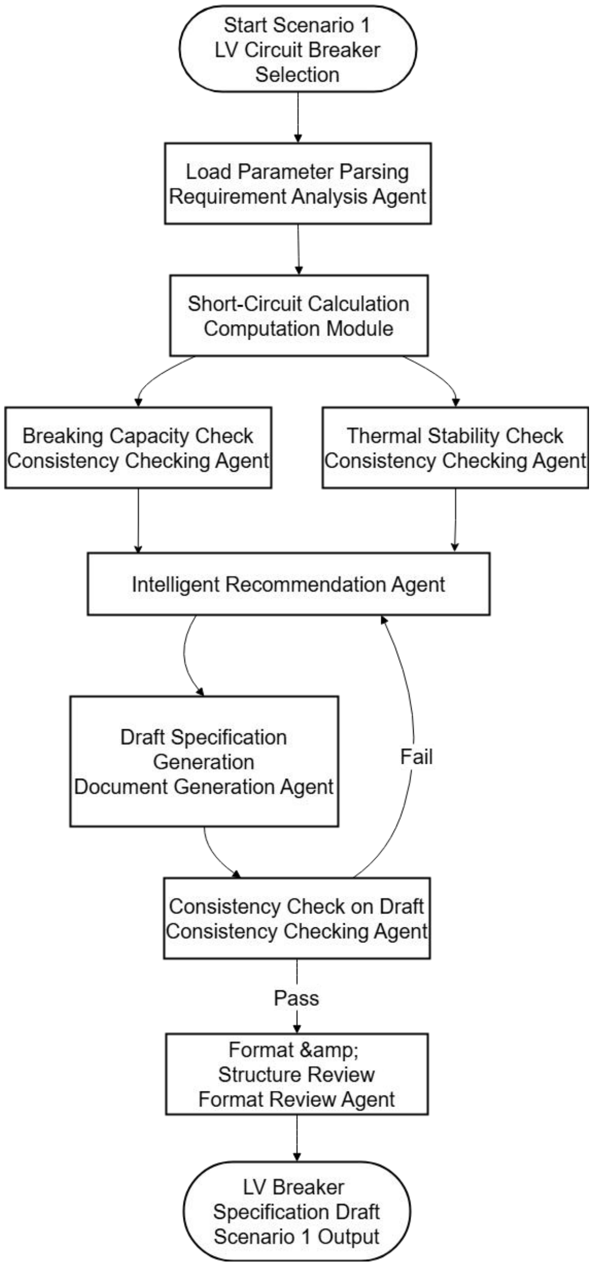

Figure 6 Example DAG Execution for Scenario 1: Low-Voltage Circuit Breaker Selection

4.2.2 Scenario 2: Technical specification for 1E-class switchgear

This scenario focuses on the system's capabilities for long-document generation, clause composition, and document structuring. The task involves:

• Determining applicable standards based on safety class;
• Automatically identifying constraints on switchgear compartmentalization, environmental qualification, and EMC;
• Retrieving 1E equipment-related content from the knowledge graph;
• Organizing the document structure into sections and subsections;
• Generating main chapters such as environmental conditions, performance requirements, and testing requirements;
• Producing a standardized technical specification draft.

This scenario tests:

• The system's global understanding of the standard system;
• Its ability to combine knowledge from multiple sources;
• The correctness of document hierarchy and structure;
• Its mastery of engineering writing style.

Expert feedback indicates that the generated documents exhibit clear logic, complete chapter structures, and accurate clause references, making them suitable as initial drafts for engineering use.

4.3 Analysis of Graph-RAG Retrieval Enhancement

Accurate retrieval of standard clauses and equipment parameters is central to design tasks. Observations suggest that Graph-RAG offers clear advantages over pure vector-based RAG:

4.3.1 Engineering-aware retrieval

By exploiting entity relations and clause dependencies in the knowledge graph, Graph-RAG can:

• Filter out irrelevant content based on standard-chapter-clause hierarchy;
• Automatically focus on test requirements and applicability ranges specific to the equipment type;
• Expand retrieval scope along graph paths to avoid missing key clauses.

4.3.2 Logically coherent results

Graph-RAG provides "clause chains" that explicitly show how clauses reference and constrain each other. This is critical in nuclear engineering because:

• Clauses often cannot be interpreted in isolation;
• Calculation conditions are frequently defined jointly by multiple chapters;
• Equipment selection must satisfy multiple interacting constraints.

4.3.3 Higher authority of retrieved content

The system prioritizes higher-level sources when fusing retrieval results:

• National standards (GB);
• Industry and nuclear standards (DL, NB, HAF);
• IEC international standards.

Consequently, the generated content aligns well with engineers' implicit hierarchy of standard authority.

4.4 Analysis of Multi-Agent Collaborative Design

In multiple simulated runs, the MAS demonstrates good stability and interpretability.

4.4.1 Clear division of labor and boundaries

Each agent is associated with different knowledge subgraphs, standard packages, and rule sets, which leads to:

• Outputs that reflect distinct role responsibilities;
• Clear error localization when issues arise;
• Non-overlapping decision boundaries.

For example, the Requirement Analysis Agent does not perform numerical calculations, the computation-related agents do not generate long-form documents, and the Document Generation Agent does not define technical calculation logic. Engineers note that this division of labor closely mirrors real design team collaboration patterns.

4.4.2 Smooth orchestration and clear logical chains

DAG-based orchestration ensures that task dependencies strictly follow nuclear design workflows. The orchestrator dynamically schedules serial, parallel, and iterative tasks according to dependency constraints and agent outputs. An illustrative example of this process is shown in Figure 6, which highlights the coordination among agents and the propagation of evidence chains.

4.4.3 Positive human-AI interaction

During design execution, the Interactive QA Agent provides:

• Explanations of standard clauses;
• Clarifications of parameter definitions;
• Justifications of design decisions.

This functionality significantly increases the transparency of AI behavior and reduces the "black-box" perception common in LLM-based systems.

4.5 Quality and Engineering Usability of Generated Documents

In multiple tasks, the Document Generation and Format Review Agents exhibit the following strengths:

4.5.1 Document structure aligned with nuclear industry practice

The generated chapter organization closely matches the structure of standard nuclear engineering templates, including:

• Scope;
• Normative references;
• Operating environment;
• Technical requirements;
• Test requirements;
• Appendices and data sheets.

The structure is complete, logically ordered, and convenient for review.

4.5.2 Appropriate engineering writing style

The generated text typically exhibits:

• A rigorous and objective tone;
• Reasonable paragraphing and sectioning;
• Clear indication of clause sources;
• Appropriate use of domain-specific terminology.

This yields documents with high engineering usability.

4.5.3 Traceable design rationale

Documents explicitly or implicitly annotate:
- Source standards;
- Knowledge graph entities;
- Constraints;
- Reasoning chains.

This greatly facilitates document review, re-checking, and regulatory auditing.

4.6 Engineering Feedback and Application Prospects

Expert interviews and repeated system runs suggest that:
- The method significantly reduces manual standard lookup and repetitive calculation work;
- Generated documents are close to actual design documents and suitable as drafts or auxiliary materials;
- The system's understanding and referencing of standards are highly reliable, supporting normalized design;
- The MAS architecture can be extended to additional subdomains such as protection, automation, and grounding;
- The overall methodology is practically deployable as an internal intelligent design platform for nuclear power enterprises.

These observations are consistent with prior experience in building knowledge-graph-based decision-support systems for nuclear safety review.

5 CONCLUSION

This paper proposes an intelligent nuclear power electrical design methodology that integrates knowledge-graph-enhanced retrieval-augmented generation (Graph-RAG) with a multi-agent system (MAS). The method addresses key challenges in nuclear electrical design, including an extensive standard, difficult knowledge reuse, and the limited reliability of general-purpose LLMs in specialized domains.

We construct a domain knowledge graph covering standards, equipment parameters, and design rules, providing a structured knowledge backbone. A graph-guided hybrid retrieval strategy is designed to integrate vector retrieval, knowledge graph retrieval, and BM25 retrieval. A multi-factor scoring and re-ranking mechanism significantly improves retrieval accuracy and effectively reduces hallucinations. A multi-agent collaborative architecture—including requirement analysis, intelligent recommendation, automatic generation, consistency checking, format review, and interactive QA agents—is implemented, and DAG-based task orchestration enables parallel and dynamic execution of complex design workflows.

Theoretical analysis and representative engineering scenarios demonstrate that, compared with traditional methods, the proposed approach offers notable advantages in knowledge reliability, design traceability, collaboration efficiency, and quality assurance. Each design decision can be traced back to explicit standard clauses and equipment parameters, and intelligent checking mechanisms ensure the standard compliance of design documents. The method provides a feasible pathway for the intelligent transformation of nuclear power electrical design and a transferable technical reference for the intelligent upgrading of other complex engineering systems.

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 Z, Lee N, Frieske R, et al. Survey of hallucination in natural language generation. ACM Computing Surveys, 2023, 55(12): 1-38.
[2] Lewis P, Perez E, Piktus A, et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. In: Proceedings of NeurIPS, 2020: 9459-9474.
[3] Gao Y, Xiong Y, Gao X, et al. Retrieval-augmented generation for large language models: A survey. arXiv preprint arXiv:2312.10997, 2023.
[4] Peng B, Galkin M, He P, et al. Graph retrieval-augmented generation: A survey. arXiv preprint arXiv:2408.08921, 2024.
[5] Wang Y, Zhong Q, Liu L, et al. A survey on LLM-based multi-agent systems: Workflow, infrastructure, and challenges. Autonomous Agents and Multi-Agent Systems, 2024, 38(2): 1-35.
[6] Buehler M J. Generative retrieval-augmented ontologic graph and multi agent strategies for interpretive large language model-based materials design. arXiv preprint arXiv:2310.19998, 2023.

[7] Huang Q, Peng S, Deng J, et al. A review of the application of artificial intelligence to nuclear reactors: Where we are and what's next. Heliyon, 2023, 9(10): e13883.
[8] Guo, C, Yi, Y, Luo, W, et al. Design and prototype implementation of a nuclear power plant safety review decision-support system based on knowledge graphs. Nuclear Safety, 2024, 23(3): 55-62
[9] Yang R, Xue H, Razzak I, et al. Divide by question, conquer by agent: SPLIT RAG with question driven graph partitioning. arXiv preprint arXiv:2505.13994, 2025.
[10] Wang S, Fan W, Feng Y, et al. Knowledge graph retrieval-augmented generation for LLM based recommendation. arXiv preprint arXiv:2501.02226, 2025.
[11] Zhu X, Xie Y, Liu Y, et al. Knowledge graph-guided retrieval augmented generation. In Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies. Albuquerque, New Mexico. Association for Computational Linguistics, 2025: 8912-8924. DOI: 10.18653/v1/2025.naacl-long.449.
[12] Shen H, Diao C, Vougiouklis P,et al. GeAR: Graph-enhanced agent for retrieval-augmented generation. arXiv preprint arXiv:2412.18431, 2024.