Effect of State-Driven Rule-Based Supervisory Prompt Design on Routing Stability in Multi-Agent LLM Systems

Seungjin Baek [a,b], Gayeon Kim [a,b], Joowoon Cha [a,c], Yonggyun Yu [a,c], Seung Geun Kim [a,*]
[a] *Applied Artificial Intelligence Section, Korea Atomic Energy Research Institute, Daejeon 34057, Republic of Korea*
[b] *Korea University of Technology and Education (KOREATECH), Cheonan 31253, Republic of Korea*
[c] *University of Science and Technology, 217, Gajeong-ro, Yuseong-gu, Daejeon, 34113, Republic of Korea*

[*] *Corresponding author: sgkim92@kaeri.re.kr*

Keywords: Large Language Models, Multi-Agent Systems, Supervisory Control, Prompt Engineering, Routing Stability

1. Introduction

Since the public release of ChatGPT in late 2022, large language models (LLMs) have become a core technology widely recognized by the general public. Recently, they have expanded beyond simple question and answer to Agentic LLMs that can call external tools, plan tasks, and carry out tasks across multiple steps.

In this context, the Korea Atomic Energy Research Institute (KAERI) has been developing an operation support system based on Agentic LLMs using an integral pressurized water reactor (iPWR) simulator as a testbed. Previous research proposed a control agent that converts natural language instructions into simulator control actions [1]. Subsequent studies strengthened stability through a role-separated structure for planning, approval, and execution [2], and addressed issues related to long-running tool execution and session interruptions [3].

In this study, while maintaining the system architecture proposed in these prior studies, we verify how consistently and reliably inter-agent interactions and workflow control operate in a multi-agent environment.

In the current architecture, a high-level agent (Supervisor) routes tasks to lower-level agents based on structured state keys, such as whether a plan exists, the approval status, and the currently active step. This corresponds to a rule-based approach in which routing is determined by state keys, and the same routing decision is expected when the same state keys are provided. This approach is well suited to reactor operation support, where strict adherence to procedures is critical.

However, during repeated experiments, cases were observed in which the Supervisor's routing decisions differed even when state keys were identical. Our analysis confirmed that the system prompt design can strongly influence routing judgments. In a system that requires predictable control, such shifts can trigger incorrect workflow transitions. Therefore, this study compares three Supervisor system prompt designs and analyzes how routing stability changes.

2. System Overview

2.1 Overall Architecture

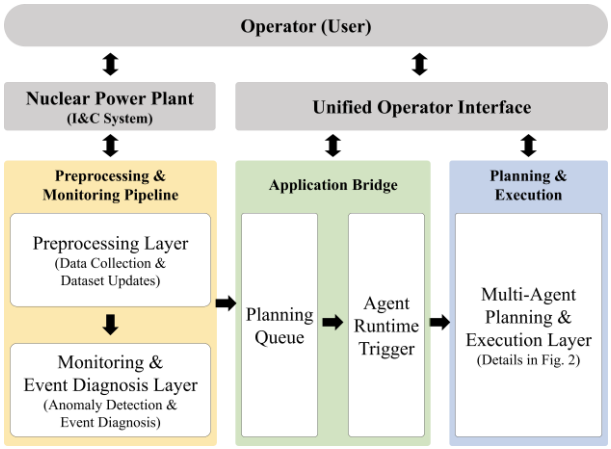

Fig. 1. Overall system architecture and processing pipeline

The system used in this study operates on the iPWR simulator and follows the hierarchical, agent-based architecture developed in previous studies [1–3]. As shown in Fig. 1, the overall system consists of a preprocessing and monitoring pipeline and a multi-agent planning and execution layer, and all interactions with the operator occur through a single integrated interface.

The preprocessing layer collects diagnostic data from the simulator and updates a structured dataset, which is analyzed by the monitoring and event diagnosis layer to perform anomaly detection and event diagnosis. Although anomalous states are recorded as events and an alarm identifier is generated, the agent workflow starts only when the alarm is registered in the plan queue through the application bridge, ensuring that only verified events trigger the subsequent decision-making process.

When an alarm arrives, the control flow is passed to the structured multi-agent planning and execution layer shown in Fig. 2. The system separates roles into PlanTeam, Manager, and WorkTeam to improve clarity and predictability in workflow control. PlanTeam generates step-by-step procedures based on retrieved domain knowledge and the diagnosed situational information, Manager manages approval and readiness to proceed through interaction with the operator, and WorkTeam invokes simulator control tools only for approved steps and reports the results in a structured form.

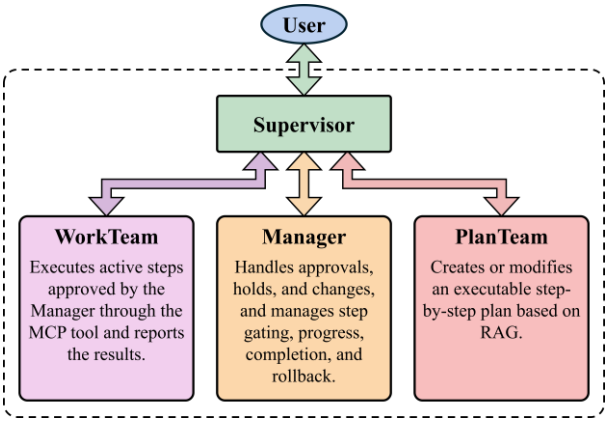

Fig. 2. multi-agent planning and execution workflow

The Supervisor is the high-level agent that coordinates these roles. The Supervisor does not generate plans or execute tools, and instead checks the structured current state at each interaction turn to determine the next control flow. Details of this mechanism are described in next section.

2.2 Supervisor I/O Structure

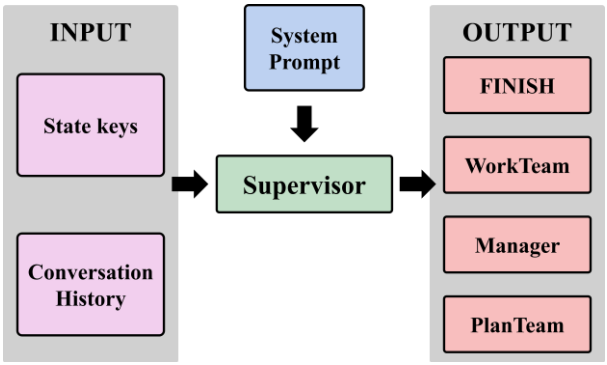

Fig. 3. Supervisor input-output structure

Fig. 3 shows the input and output structure of the Supervisor. At each routing point, it receives a set of structured state keys representing the current workflow conditions, including plan existence, approval status, execution permission, and the active step position. These keys are automatically updated whenever an agent completes its task and serve as the primary basis for routing.

In addition to the state snapshot, the Supervisor also receives the accumulated conversation history between the agent system and the operator, including user messages and outputs from PlanTeam, Manager, and WorkTeam. However, the system is designed so that routing decisions depend on the state keys rather than on the conversation content.

This design intention is stated in the system prompt, a fixed instruction consulted with highest priority that does not change during execution. It defines the allowable routing targets and their selection conditions so that the Supervisor applies consistent criteria.

Based on these inputs and instructions, the Supervisor outputs a single routing result in a fixed format, limited to one of PlanTeam, Manager, WorkTeam, or FINISH.

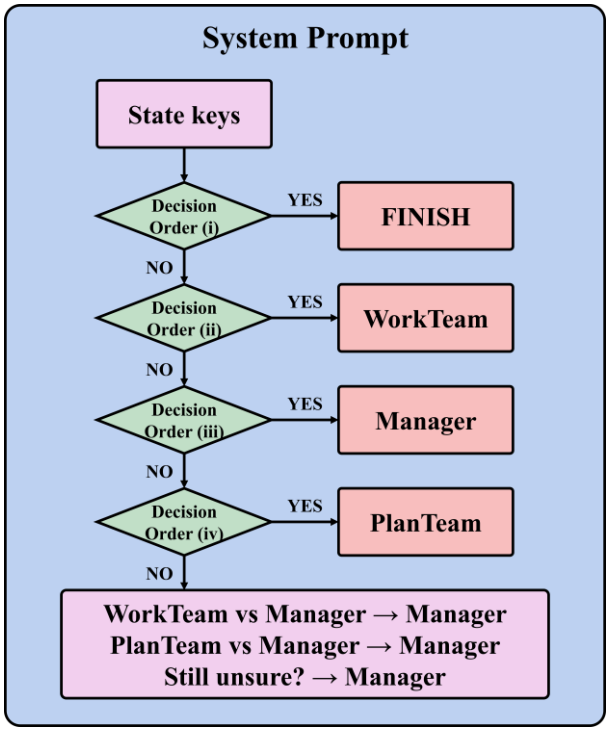

Fig. 4. The Supervisor's routing logic defined in the prompt

Table I: Routing conditions and expected outcomes

Decision Order	State condition	Expected routing
(i)	If **any** of the following conditions is satisfied: • should_finish = true • awaiting_user_response = true	FINISH
(ii)	If **all** of the following conditions are satisfied: • plan_exists = "yes" • total_steps > current_step_index • next_step_id ≠ "-" • approval_status = "approved" • approved_for_action = true • should_finish = false • awaiting_user_response = false	WorkTeam
(iii)	If **any** of the following conditions is satisfied: • plan_exists = "yes" and approval_status ∈ {pending, declined, ask, unknown} • plan_exists = "yes" and total_steps > current_step_index and next_step_id ≠ "-" and approved_for_action = false • plan_exists = "yes" and work_success = true	Manager
(iv)	If **any** of the following conditions is satisfied: • plan_exists = "no" total_steps = 0	PlanTeam

Fig. 4 and Table I summarize the rule-based routing logic. State-based conditions are evaluated in priority order; once a condition is satisfied, the corresponding target is selected and the remaining conditions are not evaluated.

PlanTeam is selected when no plan exists, Manager when approval handling or operator judgment is required, and WorkTeam when an approved executable step is available. FINISH is selected when user input is required or no further automated action can proceed, indicating that the current turn should end and control should return to the operator.

Although Table I captures the decision rules defined in the system prompt, during execution, cases were observed in which routing was influenced by the conversation history, contrary to the original design intention. Section 3 describes controlled experiments designed to quantitatively analyze this behavior while keeping the same rules.

3. Experimental Setup

3.1 Experimental Scenario

All experiments were conducted using a steam generator tube rupture (SGTR) response procedure scenario, and the procedure consists of twelve predefined steps.

The focus of this study is not the accident scenario itself, but the Supervisor's decisions under a fixed operational context. Accordingly, a single scenario was used throughout all experiments, and rather than comparing different accident types, routing decisions were evaluated at different points within the same workflow. During scenario execution, the workflow follows a recurring structure consisting of plan generation, approval handling, step execution, and waiting states.

3.2 Control of Routing Conditions

The routing conditions used in the experiments are defined in the Supervisor system prompt and follow the rule-based routing logic summarized in Table I of Section 2. The routing logic itself was not modified.

For each experimental case, the state keys were configured so that only a single routing condition was satisfied, thereby fixing the expected routing decision. The same condition was maintained across repeated executions, allowing routing consistency to be evaluated under controlled conditions.

3.3 Design of Conversation Contexts

To examine whether the conversation history actually influences routing decisions, two types of conversation history were designed, referred to as Context A and Context B.

Context A consists of a concise and clear conversation history and reflects the form commonly observed in the early stages of a scenario. It contains little role-related language or additional explanation, and because the conversation history aligns with the routing conditions defined by the state keys, it functions as an input with limited semantic cues that could influence routing decisions.

In contrast, Context B was reconstructed based on logs collected at points where routing failures were observed during actual scenario execution. Although the state keys satisfy the logical routing conditions, the conversation history contains accumulated descriptive text that implies operator intervention or role transitions, introducing semantic noise. Therefore, Context B is not intended to represent a typical conversation distribution, but rather serves as a stress test input to evaluate how well prompt design maintains routing consistency in the presence of semantic noise.

3.4 Supervisor Prompt Design

To control the extent to which information beyond the state keys influences routing decisions, three types of Supervisor system prompt were designed.

The 'Strict' system prompt restricts routing decisions strictly to state key information. That is, it instructs the Supervisor to ignore all text contained in the conversation history and to select routing decisions solely based on the state keys according to predefined rules.

The 'Hybrid' system prompt gives the highest priority to state keys while allowing limited reference to the conversation history only when the routing decision is ambiguous. In this case, the conversation history may be used only as supplementary information.

The third system prompt, 'Contextual', allows broad interpretation of the conversation history in addition to state key information. Under this prompt, the Supervisor may consider the overall workflow state and surrounding context when determining routing decisions.

These system prompt designs represent a gradual increase in the level of contextual allowance. No quantitative weighting or tuning procedures were applied, and the prompts were defined solely to enable comparative experiments based on design intent.

3.5 Experimental Settings and Evaluation Method

All experiments were conducted using the OpenAI GPT-4o model, and the temperature parameter was fixed at 0 to eliminate probabilistic variation. Except for the Supervisor system prompt, all other agent prompts and the workflow structure were kept identical across all executions.

For each routing condition, both Context A and Context B were applied. For every combination of routing condition, conversation history, and system

prompt design, the Supervisor was invoked 150 times with identical inputs.

Routing accuracy was evaluated by comparing the selected routing decision with the expected routing decision defined by the corresponding routing condition, and any mismatch was counted as an error. Routing determinism was evaluated based on whether the routing decision remained consistent across repeated executions under the same input conditions.

4. Results and Discussion

The routing performance for the four conditions defined in Section 3 is summarized in Table II. Conditions (i) through (iv) correspond to routing to FINISH, WorkTeam, Manager, and PlanTeam, respectively.

Table II: Routing accuracy
(number of correct routings out of 150 runs)

Condition	Context	System Prompt		
		Strict	Hybrid	Contextual
(i)	A	150 / 150	150 / 150	150 / 150
	B	150 / 150	150 / 150	150 / 150
(ii)	A	150 / 150	150 / 150	150 / 150
	B	150 / 150	0 / 150	0 / 150
(iii)	A	150 / 150	150 / 150	150 / 150
	B	150 / 150	70 / 150	138 / 150
(iv)	A	150 / 150	150 / 150	150 / 150

Under Context A, all four conditions produced correct routing decisions across all configurations in all 150 repeated executions. In this setting, the accumulated conversation history is limited, so the state-based rules were applied consistently as intended, and no routing instability was observed.

Under Context B, differences in routing decisions were observed for some conditions depending on the system prompt configuration. For Condition (i), all configurations consistently selected FINISH even under Context B. This is because Condition (i) represents a waiting or termination state that requires user input, and the state keys clearly indicate that no agent should proceed. Therefore, even when reference to the conversation history was allowed, the final outcome did not change.

In contrast, for Conditions (ii) and (iii), routing differences were observed under Context B depending on the system prompt configuration. The state keys indicate workflow progression (WorkTeam or Manager), but the accumulated descriptions in the conversation history can imply user participation and be interpreted as a waiting state such as FINISH. Because of this mismatch, configurations that allowed the conversation history to influence routing decisions sometimes transitioned incorrectly to waiting states.

Condition (iv) represents a state in which no plan exists and corresponds to the initial stage of the scenario, where PlanTeam is invoked. In the experiments, Condition (iv) occurred only at the beginning of the scenario and, by design, does not appear in situations where substantial conversation history has accumulated, such as Context B. Therefore, Condition (iv) was evaluated only under Context A, and under this condition, all configurations consistently selected PlanTeam.

Another point worth examining is the comparison between the Hybrid and Contextual system prompts. Because the Contextual system prompt allows broader use of the conversation history, it was expected to show lower accuracy than Hybrid. However, in some cases, Contextual showed slightly higher accuracy. This suggests that performance degradation may be determined not by the size of the allowed scope, but by whether the conversation history is allowed as a basis at all, and once information not directly related to routing is permitted to enter as evidence, incorrect routing can occur.

Overall, these results show that routing stability of the Supervisor is maintained at the highest level when routing decisions are strictly based on structured state keys alone.

5. Conclusions

This study analyzed how Supervisor system prompt design affects routing stability in multi-agent LLM systems. In an environment where agent control flow is managed using structured state keys, we experimentally evaluated the consistency of routing decisions when the same state keys are given.

When the Supervisor's routing basis was restricted to state keys alone, routing decisions remained identical across repeated runs under the same conditions. In contrast, when interpretation of the accumulated conversation history was allowed, routing shifts occurred at certain steps even with identical state keys. These results show that routing instability can arise not from randomness, but when the prompt fails to sufficiently separate routing criteria from descriptive context.

Therefore, in LLM-based systems that control multiple agents using structured state keys, the Supervisor must clearly restrict the scope of information used for routing at the system prompt level. The more the workflow separates planning, approval, and execution roles, the more such constraints determine the predictability of the control flow.

First, the evaluation in this study was conducted using limited conversation history types and at specific steps of a single accident scenario, and it may not fully reflect the diversity of history length, writing style, and role-specific utterance patterns observed in actual operations. Further verification that includes a wider range of accident types, procedures, and conversation history patterns is needed.

Second, we did not quantify differences in constraint levels between prompts. Future work should structure prompt constraints and establish a metric that quantitatively represents the boundary between state-based routing information and descriptive context.

Future studies can evaluate whether the same phenomenon is reproduced in settings that allow natural language communication between agents instead of state keys. Regardless of the representation, explicitly defining the scope of context that can influence routing remains a key research challenge.

Acknowledgment

This research was supported by a grant from Korea Atomic Energy Research Institute (KAERI) (No. KAERI-526140-26)

REFERENCES

[1] Y. P. Lee, S. G. Kim, and Y. Yu, LLM-Based Integrated Control Agent System for Nuclear Reactor, Transactions of the Korean Nuclear Society Spring Meeting, Jeju, Korea, May 22-23, 2025.

[2] G. Han, Y. Yu, and S. G. Kim, Development of an Autonomous Reactor Operation LLM Agent for Real-Time Detection and Proactive Mitigation, Transactions of the Korean Nuclear Society Autumn Meeting, Changwon, Korea, October 30-31, 2025.

[3] S. Chae, Y. Yu, S. G. Kim, and Y. Kim, Enabling Robust Parallel Tool Calling in Agentic AI via OS-Level Scheduling for the iPWR Simulator, Transactions of the Korean Nuclear Society Autumn Meeting, Changwon, Korea, October 30-31, 2025