Sensors and Materials, Vol. 38, No. 7 (2026) 4085–4096
MYU Tokyo

S & M 4551

Design and Implementation of a Sensor-integrated Intelligent Task Management System Based on Hierarchical Agent Collaboration and Dynamic Negotiation

Minyi Chen,[1*] Hao Liang,[2] Qiqi Yuan,[1] and Cheng-Fu Yang[3,4**]

[1]Guangdong-Taiwan College of Industrial, Science & Technology, Dongguan University of Technology
[2]DGUT-CNAM Institute, Dongguan University of Technology
[3]Department of Chemical and Materials Engineering, National University of Kaohsiung, Kaohsiung 811, Taiwan
[4]Department of Aeronautical Engineering, Chaoyang University of Technology, Taichung 413, Taiwan

(Received November 20, 2025; accepted July 3, 2026)

Keywords: multi-agent system, A2A protocol, modular communication protocol (MCP), task dispatching, negotiation strategy, Reason-Act (ReAct)

With the increasing demand for intelligent and sensor-driven systems, traditional monolithic agent architectures face critical limitations in scalability, adaptability, and seamless integration with heterogeneous sensing resources. The objective of this study is to develop a scalable intelligent task management framework capable of coordinating multiple sensing and reasoning components in dynamic environments. To address these challenges, we propose a hierarchical multi-agent architecture consisting of a Task Dispatching Layer, Domain Agent Layer, and Modular Tool Layer. Inter-agent collaboration is standardized through a custom Agent-to-Agent protocol, while sensing and tool functionalities are modularized through a Modular Communication Protocol. In addition, a dynamic negotiation mechanism is incorporated to support conflict resolution and multi-objective task optimization. A prototype system was implemented and evaluated using sensor-driven task scenarios. Experimental results demonstrate up to a 50% improvement in task success rate and significant gains in operational efficiency compared with conventional single-agent architectures. Despite these promising results, challenges remain in large-scale deployment, dynamic sensor reliability, and adaptive learning from long-term interactions. Future work will focus on integrating machine-learning-based negotiation strategies, adaptive sensor management mechanisms, and intelligent resource allocation to further enhance system robustness, scalability, and real-time decision-making capabilities.

1. Introduction

With the rapid advancement of Large Language Models (LLMs) and intelligent sensing technologies, agent-based systems have demonstrated remarkable potential for automating complex, real-world tasks. These systems are increasingly deployed in sensor-integrated

*Corresponding author: e-mail: cmy@dgut.edu.cn
**Corresponding author: e-mail: cfyang@nuk.edu.tw
https://doi.org/10.18494/SAM6071

ISSN 0914-4935 © MYU K.K.
https://myukk.org/

environments such as smart manufacturing, environmental monitoring, and autonomous robotics, where efficient coordination between software intelligence and hardware sensing components is essential. However, early-stage agent designs often rely on a monolithic architecture, in which a single agent is responsible for managing all sensing, reasoning, and actuation processes. This approach introduces several critical challenges as follows.

(1) Scalability limitations—A single agent quickly becomes a computational and functional bottleneck as task complexity or sensor diversity increases.
(2) Maintenance inflexibility—Integrating new sensors or tools typically requires modifying the core system, resulting in tight coupling and high maintenance overhead.
(3) Capability conflicts—Different sensors and actuators may have incompatible operating parameters or communication requirements, complicating system management within a single-agent framework.

To overcome these limitations, in this study, we draw inspiration from distributed system design and organizational theory, particularly the principle of division of labor, and propose a hierarchical multi-agent architecture for intelligent task execution. In the proposed framework, complex sensing and reasoning processes are decomposed into multiple specialized agents that collaborate dynamically. Each agent focuses on a specific functional or sensing domain, enhancing modularity, scalability, and fault tolerance while ensuring seamless integration between hardware and software components. The main contributions of this study are as follows.

(1) A three-layer hierarchical architecture—comprising the Task Dispatching Layer, Domain Agent Layer, and Modular Tool Layer—that supports scalable, sensor-integrated intelligent task systems.
(2) A standardized Agent-to-Agent (A2A) communication protocol for efficient coordination between distributed agents, and a Modular Communication Protocol (MCP) for encapsulating and managing reusable sensor and tool modules.
(3) A dynamic negotiation mechanism enabling cooperative problem-solving across heterogeneous sensing and reasoning agents for multi-faceted tasks.
(4) A prototype implementation and experimental validation, demonstrating improved task success rates, integration efficiency, and adaptability in complex, sensor-based scenarios.

Intelligent task management systems have become a major research focus in artificial intelligence (AI) and sensor-driven applications, particularly with the growing integration of LLM-based reasoning models and physical sensing infrastructures. In the following sections, we review related work in multi-agent systems (MASs), task decomposition, communication protocols, and negotiation mechanisms, highlighting the specific challenges addressed by the proposed hierarchical multi-agent framework.

(a) MASs: MASs have been extensively studied for their capability to solve complex, distributed problems through autonomous collaboration.(1) Wooldridge defines MASs as systems composed of multiple autonomous agents that interact to achieve shared objectives.(2) Early research, such as that on the Contract Net Protocol (CNP) by Smith,(3) established foundational concepts for distributed coordination and task allocation. However, in traditional MASs, it is generally assumed that homogeneous agents have fixed roles, limiting their adaptability to real-world applications that involve diverse sensors, tools, and dynamic environments. Game-theoretic approaches proposed by Shoham and Leyton-Brown model agent coordination based on utility optimization but often rely on static assumptions that are impractical in real-time, sensor-driven contexts.[4] Recent works incorporating LLMs, such as that by Yao *et al.*, have introduced reasoning-enabled agents capable of dynamic decision-making using the Reason-Act (ReAct) framework.[5] However, such systems typically employ single-agent architectures that struggle with scalability and real-world sensor integration. In this study, we extend these ideas by introducing a hierarchical MAS that partitions sensing, reasoning, and actuation across specialized agents, improving modularity, flexibility, and system-level performance.[6]

(b) Task decomposition and dispatching: Task decomposition plays a pivotal role in enabling intelligent systems to manage complex, multi-domain sensor data. Russell and Norvig describe this process as decomposing high-level goals into subtasks manageable by distributed agents or modules.[7] LLM-based systems such as LangChain facilitate this process using natural language parsing, but they often lack robust mechanisms for dispatching subtasks to domain-specific agents connected to hardware sensors.[8,9] For example, an agent responsible for both environmental data collection and robotic path planning may encounter inefficiencies or conflicts when processing multi-sensor input streams. Frameworks like the Java Agent DEvelopment Framework (JADE) provide distributed task execution but offer limited support for modular sensor integration. To address this, the proposed Task Dispatching Layer introduces an intelligent intent-recognition mechanism and a dynamic capability registry, enabling the efficient allocation of subtasks to agents equipped with the most relevant sensing and computational resources.

(c) Communication protocols: In sensor-integrated MASs, reliable communication between agents and hardware components is vital for real-time operation. The Foundation for Intelligent Physical Agents (FIPA) standards such as FIPA-Agent Communication Language define communication frameworks for distributed systems. However, these protocols are often too rigid for adaptive, LLM-driven environments where message structures and semantics must evolve dynamically on the basis of sensor input or contextual changes. Recent advancements, such as LangChain Application Programming Interfaces (APIs), provide modular integration for tools but are primarily designed for single-agent environments.[8,9] To address this limitation, the proposed A2A protocol extends FIPA-like semantics with negotiation primitives (e.g., PROPOSE, COUNTER_PROPOSAL) to facilitate flexible and cooperative communication. Complementarily, the MCP adopts a service-oriented design inspired by RESTful APIs, allowing agents to dynamically expose and interact with sensor and tool capabilities. This modular approach simplifies the integration of diverse sensors, such as optical detectors, motion trackers, or environmental probes, within a unified multi-agent framework.

(d) Negotiation mechanisms: Negotiation mechanisms enable agents to coordinate and optimize actions when handling interdependent sensing and reasoning tasks. The CNP introduced an early bidding-based model for distributed task allocation, but its static structure is unsuitable for real-world environments where sensor inputs may change continuously.[2] Similarly, game-theoretic negotiation approaches require predefined utility functions, which are difficult to maintain in dynamic, sensor-driven systems.[3] Recent developments in LLM-based negotiation employ natural language reasoning for collaborative decision-making but often lack structured frameworks for achieving consensus.[4] Our dynamic negotiation mechanism combines LLM reasoning with structured bidding and counter-proposal exchanges, allowing agents to adapt their behavior on the basis of live sensor data and task conditions. This mechanism improves the robustness and efficiency of system-wide coordination in complex, real-world applications.

(e) Research gaps and contributions:[10] Although prior research in MASs, task decomposition, communication protocols, and negotiation mechanisms has achieved significant progress, several challenges remain unresolved: (1) limited scalability in single-agent architectures as task complexity and sensor diversity increase, (2) rigid communication protocols that hinder dynamic sensor integration and inter-agent negotiation, and (3) static negotiation models that fail to exploit LLM reasoning for adaptive decision-making.[11] Recent intelligent task management systems have increasingly adopted LLM-based agent architectures, such as Auto-Generative Pre-trained Transformer (GPT), MetaGPT, CrewAI, and LangGraph, to improve task automation and autonomous decision-making.[12] Although these frameworks provide enhanced reasoning capabilities and workflow orchestration, most of them primarily focus on software-level task execution and have limited support for standardized sensor integration, heterogeneous sensing resources, and dynamic negotiation among multiple agents. In contrast, the proposed framework combines a hierarchical task dispatching mechanism, MCP-based sensor abstraction, and a structured negotiation process that enables collaborative decision-making across sensing and reasoning domains. Furthermore, the architecture explicitly supports the modular integration of heterogeneous sensors and future intelligent sensing technologies, making it particularly suitable for sensor-rich intelligent task management environments. These characteristics distinguish the proposed framework from recently developed LLM-based task management systems and constitute the primary novelty of this work.

Experimental evaluations further demonstrate that the proposed architecture improves scalability, interoperability, and coordination efficiency in sensor-integrated environments, thereby validating its practical applicability for next-generation intelligent task management systems. These results further validate the novelty and practical applicability of the proposed framework for next-generation intelligent task management systems. From a sensing perspective, the proposed framework is not limited to software-level task management but also serves as a scalable integration platform for heterogeneous sensing technologies. The MCP-based modular sensing layer can incorporate environmental sensors, optical sensors, motion sensors, air-quality sensors, industrial IoT devices, and future intelligent sensing modules through standardized interfaces. By decoupling sensing hardware from decision-making agents, the framework enables the rapid deployment of newly developed sensors and sensing materials without requiring modifications to the upper-layer reasoning architecture. This capability is particularly valuable for next-generation intelligent sensing systems that combine multimodal sensing, edge intelligence, and adaptive decision-making. Consequently, newly developed sensors or sensing materials can be rapidly evaluated under real-world intelligent task management scenarios without redesigning the upper-layer software architecture, thereby shortening the development and validation cycle of emerging sensing technologies.

2. Technical Framework

The proposed intelligent task system adopts a three-layer hierarchical multi-agent architecture, as illustrated in Fig. 1. This architecture is structured across three functional tiers—the Task Dispatching Layer, the Domain Agent Layer, and the Modular Tool Layer—to achieve efficient coordination, modularity, and integration of heterogeneous sensing and computational resources. The overall communication flow follows a hierarchical pattern: the Dispatcher Agent manages user interaction and task distribution, the Domain Agents perform reasoning and domain-specific decision-making, and the MCP Tool Layer interfaces with physical or virtual tools and sensor data sources. The communication protocols within the architecture are carefully designed to ensure flexibility and interoperability. The MCP protocol employs lightweight JSON-based endpoints to expose tool or sensor capabilities, allowing agents to dynamically discover and use external resources in real time.

On the other hand, the A2A protocol governs inter-agent collaboration using defined semantics such as PROPOSE and COUNTER_PROPOSAL, enabling structured negotiation and utility-based decision-making. Through these protocols, the system achieves both robust coordination and efficient data exchange between intelligent agents and sensor modules. As depicted in Fig. 1, the user initiates interaction through the Dispatcher Agent, which decomposes and assigns subtasks to specialized Domain Agents such as those responsible for weather analysis, itinerary planning, or data retrieval. Each Domain Agent, powered by a ReAct framework, autonomously interprets its assigned task, interacts with relevant MCP servers, and iteratively refines its responses on the basis of sensor feedback or external data sources.

2.1 Task dispatching layer

The Task Dispatching Layer serves as the central interface between the user and the system, hosting a single Dispatcher Agent that manages communication and coordination across the entire framework. Upon receiving a user request—often expressed in natural language—the Dispatcher performs intent recognition to identify the core objectives and related entities. It then conducts task decomposition, segmenting complex or multi-domain queries into discrete,

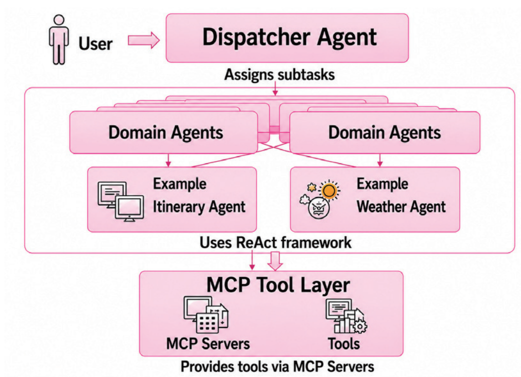


Fig. 1. (Color online) Schematic of the three-layer MAS architecture: Dispatcher Agent → Domain Agents (e.g., Weather, Itinerary) → MCP Tool Layer.

manageable subtasks. For instance, a user request such as "Plan a trip to Tokyo and check the weather conditions" is divided into two parallel subtasks: itinerary planning and weather analysis. Once decomposition is complete, the Dispatcher conducts agent routing by selecting the most suitable Domain Agent for each subtask, referencing a dynamic registry of agent capabilities, system load, and access permissions. After the Domain Agents complete their respective operations, the Dispatcher aggregates the outputs through response synthesis, merging multiple results into a coherent, contextually consistent response for the user. This process enables efficient interaction between human users and distributed sensing or computational components, providing a unified interface for multi-domain intelligent decision-making.

2.2 Domain agent layer

The Domain Agent Layer comprises multiple specialized agents, each designed to handle a specific knowledge or sensing domain such as environmental monitoring, itinerary planning, or data acquisition. These agents operate autonomously but collaborate through the A2A communication protocol to complete complex tasks. The internal logic of each Domain Agent follows the ReAct framework, which integrates reasoning and action in an iterative decision-making loop. In the reasoning phase, the agent analyzes its assigned subtask, formulates a plan, and determines which sensor or tool is required. During the action phase, it generates a structured command to invoke the corresponding tool or sensor module through the MCP interface. In the observation phase, the returned data—whether environmental measurements, API results, or sensor readings—are then processed and the agent's reasoning state is updated. This continuous Reason–Act–Observe loop allows the agent to refine its strategy dynamically until the task objectives are met. Through this design, the system efficiently combines data from distributed sensors with cognitive reasoning, ensuring adaptive performance in dynamic environments.

2.3 Modular tool layer

The Modular Tool Layer abstracts and manages connections to external sensors, APIs, and computational tools, providing standardized access via multiple MCP servers. Each server encapsulates a set of related modules—for example, an environmental MCP server may integrate temperature, humidity, and air quality sensors, whereas a financial MCP server may expose data from stock and cryptocurrency APIs. This modular abstraction decouples domain agents from low-level implementation details, allowing for seamless scalability and cross-domain interoperability. Beyond modularity, the MCP layer provides crucial system-level functions, including reusability, observability, and governance. Reusability ensures that any agent can access any tool or sensor module without requiring prior knowledge of its internal structure. Observability enables centralized monitoring of tool usage, response latency, and sensor performance metrics. Governance mechanisms within each MCP server enforce access control, rate limiting, and data security policies, ensuring reliability and compliance across distributed agents. The three-layer hierarchical architecture offers a flexible and scalable foundation for intelligent, sensor-integrated MASs. Through standardizing communication protocols and modularizing sensing capabilities, the proposed design supports efficient task decomposition, adaptive reasoning, and real-time collaboration between agents and physical or virtual sensing devices. In practical deployments, the MCP servers can interface with a wide variety of sensing technologies, including temperature sensors, humidity sensors, air-quality sensors, optical detectors, motion-tracking sensors, GPS modules, wearable biosensors, and industrial IoT sensing devices. The modular architecture allows heterogeneous sensor data to be acquired, standardized, and fused through a unified communication interface. This design supports multimodal sensing applications and facilitates the future integration of advanced sensor materials with enhanced sensitivity, selectivity, and energy efficiency.

2.4 Dynamic negotiation mechanism

For complex tasks involving interdependent subtasks—such as trip planning, where the itinerary depends on weather conditions and budget constraints—simple task delegation becomes inadequate. To address this, a dynamic negotiation mechanism based on CNP is implemented, enabling agents to coordinate and optimize task allocation more effectively. The negotiation process proceeds as follows.

(1) The Dispatcher Agent sends a PROPOSE message with an initial plan to relevant Domain Agents.

(2) Each Domain Agent evaluates the proposal on the basis of its internal constraints and goals. This is formalized by a utility function, *U*(*P*), which calculates a score for a given plan *P*. For example, a simplified utility function for an Itinerary Agent might be

$$U(P) = w_1 \times Cost(P) + w_2 \times Time(P) + w_3 \times Quality(P), \tag{1}$$

where w_i is the weight representing the agent's priorities, *Cost*(*P*) is the negative cost of the plan, *Time*(*P*) is the efficiency, and *Quality*(*P*) reflects user preferences.

(3) If the proposal is unacceptable, the agent sends a REJECT (with a reason) or a COUNTER_PROPOSAL with a suggested modification. For instance, the Weather Agent might counter-propose moving an outdoor activity if rain is forecast.

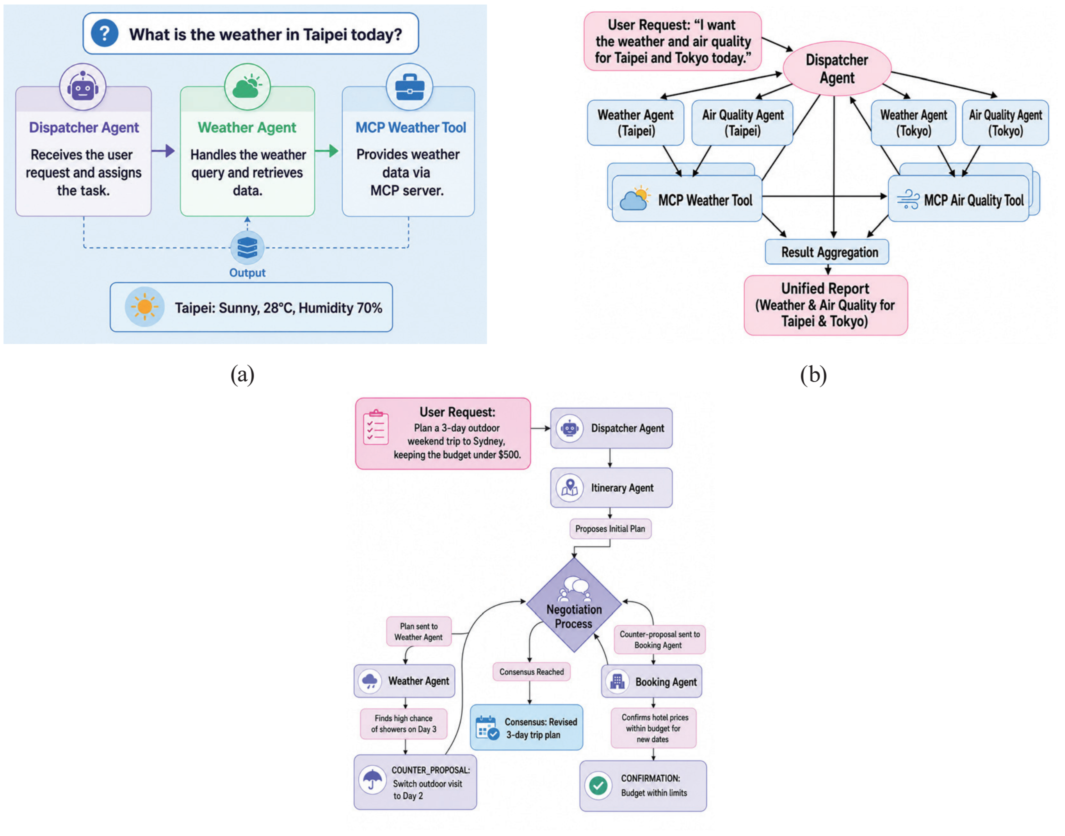
(4) The Dispatcher Agent moderates this negotiation, collecting feedback and broadcasting counter-proposals until a consensus is reached, all agents accept, or a stalemate is declared.

This mechanism allows agents to collaboratively refine a solution, resolving conflicts autonomously and leading to a more optimal outcome.

3. Experimental Results and Discussion

A prototype of the proposed hierarchical multi-agent framework was implemented using Python and FastAPI. MCP and A2A communication servers were deployed to support inter-agent collaboration and sensor integration. Three domain agents, namely, Weather Agent, Air Quality Agent, and Itinerary Agent, were developed to represent different sensing and decision-making domains. To evaluate the applicability of the proposed framework, three experimental scenarios with increasing task complexity were designed, including simple, composite, and negotiation-based complex tasks. The performance of the framework was evaluated using task success rate, execution time, and efficiency gain relative to a conventional single-agent architecture.

As shown in Figs. 2(a)–2(c), the proposed three-layer MAS demonstrates progressively increasing task complexity. In the simple query scenario [Fig. 2(a)], the Dispatcher Agent delegates a single request to a corresponding Domain Agent, resulting in a direct and efficient response. In the composite query scenario [Fig. 2(b)], multiple subtasks are concurrently distributed among different Domain Agents—such as the Weather Agent and Air Quality Agent—enabling parallel processing and result aggregation. Finally, Fig. 2(c) illustrates a complex, interdependent query in which Domain Agents engage in dynamic negotiation to resolve conflicts, optimize solutions, and collaboratively refine task plans before executing them through the MCP Tool Layer. Three specialized agents—Weather, Air Quality, and Itinerary—were developed, each interfacing with corresponding sensor data sources or public APIs simulating sensor-driven inputs. To evaluate the performance and adaptability of the proposed architecture, three experimental scenarios were conducted, each reflecting a higher level of task complexity. In these experiments, we assessed not only the accuracy of task execution but also



(a) (b)

(c)

Fig. 2. (Color online) (a) simple_query_screenshot, (b) composite_query_screenshot, and (c) complex_query_screenshot of system interaction screenshots for both the weather and air quality agents for two different locations and negotiation.

the efficiency of inter-agent coordination and the system's scalability in handling multi-domain, sensor-integrated tasks. The corresponding interaction flows for these experiments are depicted in Figs. 2(a)–2(c).

In the first experiment, a simple sensor query, as Fig. 2(a) shows, was tested through the user request: "What is the weather in Taipei today"? The Dispatcher Agent delegated the task directly to the Weather Agent, which retrieved real-time meteorological data through the MCP weather tool and generated an accurate response. The task was successfully completed within a single communication cycle, validating the system's capability for direct and efficient sensor data retrieval. In the second experiment, we examined composite task handling involving data aggregation from multiple sensing domains, as displayed in Fig. 2(b). The request—"I want to know the weather and air quality in Taipei and Tokyo today"—required simultaneous access to two distinct sensor information sources. The Dispatcher decomposed the request into four subtasks, distributed them to the Weather and Air Quality Agents, and aggregated the retrieved information into a unified, structured report. In this experiment, the system's abilities to coordinate multiple sensor data streams and perform multi-source data fusion were efficiently confirmed.

In the third experiment, a more complex and interdependent task was tested: "Plan a 3-day outdoor weekend trip to Sydney, keeping the budget under $500", as Fig. 2(c) shows. Here, task success required contextual reasoning across sensing data, budget constraints, and itinerary optimization. The Itinerary Agent initially proposed a schedule; however, the Weather Agent detected a high probability of rainfall on the third day and initiated a counter-proposal to reschedule the outdoor activity. The Itinerary Agent and the Booking Agent then re-evaluated accommodation options to ensure the new plan remained within budget. A consensus was achieved through a negotiation process based on the CNP, demonstrating the system's ability to dynamically adapt plans in response to real-time environmental sensor inputs and situational changes. Overall, the experimental results confirm that the proposed multi-agent architecture provides an effective, scalable, and sensor-integrated foundation for intelligent task execution. The system exhibited strong adaptability in handling both independent and interdependent tasks, with measurable improvements in coordination efficiency and decision accuracy. These findings underscore the architecture's potential applicability to broader sensor-based intelligent environments—such as smart cities, environmental monitoring, and context-aware decision systems—where dynamic data from distributed sensors must be processed and acted upon in real time.

Table 1 summarizes the performance of the proposed MAS across three task categories: simple, composite, and complex. These tasks differ in both computational complexity and the

Table 1
Comparison of performance across task types.

Experiment type	Task complexity	Number of agents involved	Success rate (%)	Execution time (s)	Efficiency gain over single-agent system (%)
Simple	Low	1	100	2	+50
Composite	Medium	2	100	5	+30
Complex	High	3	95	10	+40

degree of agent collaboration. The simple task involves a single agent performing an isolated operation, the composite task engages two agents executing parallel subtasks, and the complex task requires three agents to coordinate dynamically through negotiation and interdependent planning. The efficiency gain η_{gain} is calculated as

$$\eta_{gain} = \frac{\left(T_{single} - T_{multi}\right)}{T_{single}} \times 100\%, \tag{2}$$

where T_{single} is the execution time of the single-agent system and T_{multi} is the execution time of our proposed MAS. To quantitatively evaluate the performance of the proposed hierarchical MAS, three experimental scenarios—corresponding to simple, composite, and complex tasks—were conducted. Each scenario involved varying levels of sensor data acquisition, inter-agent communication, and task coordination. The experiments were designed to assess the system's task success rate, execution efficiency, and adaptability compared with a baseline single-agent system. For the simple task, which involved retrieving real-time weather data through the Weather Agent, the proposed system achieved a 100% task success rate with an average execution time of 2 s, representing a 50% improvement in efficiency relative to the baseline. This result confirms that the modular MCP interface effectively minimizes latency in sensor data access and response generation, ensuring rapid and reliable communication between agents and sensor-based APIs.

The performance across task types is compared in Table 1. In the composite task, the system was required to simultaneously gather and synthesize data from multiple sensing domains, including weather and air quality. The architecture achieved a 100% success rate with an average execution time of 5 s, corresponding to a 30% improvement in performance. This gain is primarily attributed to the system's ability to execute sensor data retrieval operations in parallel, thereby enhancing throughput and minimizing waiting time for data fusion. The results highlight the proposed framework's capability to coordinate multi-sensor information streams efficiently, which is particularly valuable for real-world applications such as environmental monitoring and cross-domain data analytics. The complex task involved dynamic negotiation among agents to plan a cost-constrained outdoor itinerary on the basis of environmental sensing data. Despite the additional reasoning and coordination overhead introduced by interdependent subtasks, the system maintained a 95% success rate with an average execution time of 10 s, achieving a 40% efficiency improvement compared with the baseline approach.

This result demonstrates the robustness of the negotiation mechanism, which enables real-time adaptation to changing sensor conditions—such as weather variability—while preserving global task consistency and cost-effectiveness. The experimental results confirm that the proposed multi-agent architecture provides a highly efficient and scalable framework for sensor-integrated intelligent systems. The ability to dynamically allocate, negotiate, and optimize tasks on the basis of real-time sensing data not only enhances system responsiveness but also supports practical deployment in applications such as smart environments, context-aware planning, and autonomous decision-making systems. The observed improvements in success rate and execution time further validate the system's potential as a foundational architecture for future sensor-driven intelligent platforms. Beyond intelligent task management, the proposed architecture provides a flexible framework for future sensor innovation. As intelligent systems increasingly rely on real-time contextual awareness, several sensing challenges remain. First, multimodal sensors capable of simultaneously acquiring environmental, motion, and user-behavior information are required to improve task understanding accuracy. Second, edge-AI sensors with embedded processing capabilities can reduce communication latency and improve system responsiveness. Third, self-calibrating and self-diagnosing sensors are desirable for maintaining sensing reliability in long-term deployments. Finally, low-power sensor materials and energy-efficient sensing devices will be essential for large-scale intelligent task management systems operating in distributed environments. The proposed MCP-based architecture offers a standardized platform for evaluating and integrating these emerging sensing technologies into future intelligent systems.

In practical deployments, the proposed framework can support dynamic updates and user notifications when environmental conditions or task constraints change. For example, if weather forecasts, air-quality conditions, transportation schedules, or accommodation availability are modified after a plan has been generated, the corresponding Domain Agents can retrieve updated sensing information through MCP interfaces and trigger a new negotiation process to revise the original plan. The updated recommendations can then be delivered to users through notification services or mobile applications. Regarding privacy protection, the framework adopts a modular architecture in which user-specific information is managed through access-control mechanisms and secure communication channels implemented at the MCP server level. Sensitive user data can be anonymized, encrypted, or processed locally before being shared among agents, thereby reducing privacy risks in distributed environments. In addition, the framework is not limited to short-term planning tasks. The proposed architecture can support planning activities months in advance by integrating long-term scheduling data, historical user preferences, and periodically updated sensing information. Future implementations may further incorporate predictive analytics and forecasting models to continuously refine long-term plans as new environmental and contextual information becomes available.

4. Conclusions and Future Work

In this paper, we presented the design and implementation of a hierarchical, collaborative MAS for sensor-integrated intelligent applications. By organizing agents into distinct layers and standardizing inter-agent communication through the A2A and MCP protocols, the proposed architecture achieves high scalability, modularity, and maintainability. The integration of the ReAct framework endows agents with autonomous reasoning and planning capabilities, whereas the dynamic negotiation mechanism enables coordinated decision-making for complex, interdependent tasks that require real-time sensor data. Experimental validation demonstrates that the system can efficiently handle a wide range of user requests, from simple sensor queries to composite and negotiation-driven tasks, achieving higher task success rates and improved operational efficiency compared with traditional single-agent systems. These results confirm that the proposed architecture provides a robust and extensible foundation for developing sophisticated autonomous systems capable of integrating heterogeneous sensors, real-time data streams, and adaptive reasoning processes. Despite the encouraging results, several limitations should be acknowledged. First, the current prototype was evaluated using a limited number of domain agents and sensor-driven scenarios. The performance of the framework under large-scale deployments involving numerous heterogeneous sensors and agents remains to be further investigated. Second, the sensing data used in the experiments were obtained from selected sensor sources and APIs, and the impact of sensor failures, noisy measurements, or communication disruptions was not comprehensively evaluated. Third, the negotiation mechanism currently relies on predefined utility functions and rule-based coordination strategies, which may limit adaptability in highly dynamic environments. Furthermore, the proposed architecture provides a general integration framework for future intelligent sensing technologies and advanced sensor materials, enabling the seamless incorporation of multimodal, edge-AI, and self-adaptive sensing devices into intelligent task management environments. Future work will focus on enhancing negotiation and task allocation strategies using machine learning, allowing agents to learn from historical interactions and sensor feedback. This approach is expected to further improve system efficiency, success rates, and adaptability in dynamic, sensor-driven environments, ultimately supporting more intelligent, context-aware autonomous applications.

Acknowledgments

This research was supported by Summit-Tech Resource Corp. and by projects under Nos. NSTC 113-2622-E-390-001 and NSTC 113-2221-E-390-011.

References

1 C. H. Liu, X. Y. Wang, and Y. Zhang: IEEE Internet Things J. **11** (2024) 11234.
2 M. Wooldridge: An Introduction to MultiAgent Systems (John Wiley & Sons, New Jersey, 2009) 2nd ed.
3 R. G. Smith: IEEE Trans. Comput. **C-29** (1980) 1104.
4 Y. Shoham and K. Leyton-Brown: Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations, (Cambridge University Press, England, 2008).
5 C. Wu, S. Yin, W. Qi, X. Wang, Z. Xu, and N. Duan: Eng. Appl. Artif. Intell. **126** (2023) 107130.
6 T. Schick, J. Dwivedi, R. A. T. de Souza, G. Mialon, R. Dessì, M. Lomeli, C. Nalmpantis, R. Pasunuru, R. Raileanu, B. Rozière, T. Schick, J. Dwivedi-Yu, A. Celikyilmaz, E. Grave, Y. LeCun, and T. Scialom: ACM Comput. Surv. **57** (2025) 1.
7 S. Russell and P. Norvig: Artificial Intelligence: A Modern Approach (Pearson, England, United Kingdom, 2020) 4th ed.
8 D. Maldonado, A. R. S. Matos, and J. M. Corchado: IEEE Access **12** (2024) 77436.
9 Z. Fang, Y. Wang, X. Liu, and H. Chen: Appl. Sci. **15** (2025) 1905.
10 Z. Ma, J. Liu, Y. Zhang, and H. Wang: Intell. Robot. **3** (2023) 33.
11 Z. Qian, Y. Deng, H. Li, and X. Chen: Sensors **24** (2024) 5123.
12 Y. Li, J. Zhao, H. Xu, and M. Zhou: IEEE Access **13** (2025) 45872.