Issue 18, part 2, 2025

https://doi.org/10.23939/sisn2025.18.2.089

УДК 004.8:004.75

COMBINING AGENTS' ORCHESTRATION WITH HUMAN CONTROL FOR AUTOMATIC PLANNING IN DISTRIBUTED SYSTEMS

Maksym Khorkanin[1], Dmytro Dosyn[2]

[1, 2] Lviv Polytechnic National University,
Department of Information Systems and Networks, Lviv, Ukraine
[1] Email: maksym.y.khorkanin@lpnu.ua, ORCID: 0009-0001-6263-6878
[2]Email: dmytro.h.dosyn@lpnu.ua, ORCID: 0000-0003-4040-4467

Modern distributed information systems increasingly require sophisticated automation approaches for task planning, resource allocation, and execution management. While AI agents based on large language models (LLMs) demonstrate significant potential for solving complex distributed computing tasks, fully autonomous solutions present substantial risks without proper human oversight and control mechanisms. This paper proposes a comprehensive hybrid multi-agent architecture that effectively combines automated planning and execution capabilities with strategically integrated human-in-the-loop (HITL) components for distributed intelligent systems. The proposed framework employs a hierarchical structure consisting of orchestrator and executor agents working in coordination with human specialists at critical decision points. The orchestrator agent manages high-level task decomposition and resource allocation across heterogeneous computing clusters, while executor agents handle local optimization and code generation for specific computational nodes. The hybrid nature of this architecture results from applying centralized orchestration at the task level while maintaining decentralized execution at the subtask and individual cluster level. Key innovations include the formal definition of agent interaction protocols, comprehensive toolsets for both orchestrator and executor agents, and multi-layered verification mechanisms that combine automated static code analysis with mandatory human expert review. The HITL components are strategically positioned at critical junctions including task decomposition confirmation, resource allocation approval, and code verification before execution on real distributed infrastructure. This approach addresses the fundamental challenge of balancing automation efficiency with operational safety and control. The framework's practical applicability is demonstrated through clearly defined agent responsibilities, tool specifications, and interaction mechanisms that enable safe deployment in real-world distributed computing environments. This research contributes both practically and scientifically to the field by providing a structured approach to deploying AI-driven distributed computing solutions while maintaining necessary human oversight, formally characterizing hybrid human-AI collaboration in multi-agent systems and establishing robust verification protocols that ensure operational safety without sacrificing computational efficiency.

Keywords – automatic planning, distributed informational systems, agentic systems, multiagent interactions, human-in-the-loop.

Problem Statement

The application of modern process automation methods based on AI agents to solve complex tasks such as modeling, task and resource distribution, and overall planning in distributed information systems allows for these tasks to be executed much more efficiently. Considering the nature and capabilities of these agents, a correct combination of several agentic models can not only make planning decisions but also directly carry out the execution of those plans.

However, without a proper interaction architecture, effective agent orchestration, and, most importantly, oversight from human specialists, there are significant risks associated with using fully autonomous solutions. Therefore, it is highly relevant to develop hybrid approaches that combine the benefits of automated planning, orchestration, and execution with the ability for human control and intervention.

Analysis of Recent Studies and Publications

Historically, the first approaches used for automatic planning for distributed systems were classical machine learning models (Al-Fraihat et al., 2024), particularly reinforcement learning for adaptive automatic planning (Mao et al., 2019). More recently, LLM agents have been used for planning in cloud, Spark, and other distributed environments (Kankaniyage Don et al., 2024).

In a broad sense, an AI agent is a software system whose execution logic is determined by the generative output of an underlying LLM (Sypherd & Belle, 2024). Within the context of planning in distributed intelligent systems, agents can manage task distribution across clusters, control the execution order of these tasks (Mediakov & Khorkanin, 2025), and even generate or optimize the code that contains the task's execution logic on a specific node or cluster. When using multiple agents, their interactions can be organized either centrally (orchestrated by one designated agent) or decentral approach, where agents have equal roles.

This work represents a logical extension of the hybrid architecture proposed in (Mediakov & Khorkanin, 2025) for automatic planning in distributed systems using multi-agent language model interaction. This hybrid approach employs a centralized orchestrator at the task level while maintaining a decentralized approach at the subtask and individual cluster level.

In both theoretical and practical contexts, the application of agent systems for resource and task planning in distributed intelligent systems (DIS), as well as other decision-making problems that impact real computing resources or the external world, are evaluated based on risks (Kankaniyage Don, Ravi, & Toxtli, 2024), and other metrics like efficiency, quality, and generalization (Li et al., 2025). Some of these risks are directly linked to issues within the underlying language models that govern the agent's behavior.

One of the most significant and recent methods for minimizing the generalization problem and mitigating certain risks is the integration of a human-in-the-loop (HITL) into the agent's decision-making process (Zhang et al., 2025). As this component is a key part of the architecture proposed in this paper, it is essential to review recent research on the use of HITL in multi-agent systems.

Numerous studies have experimentally demonstrated that incorporating a HITL improves the generalization capabilities of agents (Li et al., 2025), allows for adding clarifications and explanations to the decision-making process (Takerngsaksiri et al., 2024), and enables the application of specialist expertise during the execution of a planning task (Zou et al., 2025).

Formulation of the Article's Objective

The primary objective of this work is to formulate and propose a framework architecture for a multi-agent planner designed for intelligent distributed systems, including the description and formalization of necessary agent types, their toolsets, communication methods, and the tasks they must solve. Additionally, the work aims to define the integration points for human-in-the-loop components that enable monitoring, control, approval, or modification of automatically made decisions or generated content, such as distribution plans or task execution code. This comprehensive framework addresses the critical gap between fully autonomous distributed computing systems and the practical need for human oversight in real-world deployments, providing a structured approach to balance automation efficiency with operational safety and control.

Main Results

In the context of this work, we propose a comprehensive hybrid approach that combines a multi-agent system for automatic planning and computations on distributed information systems (DIS) with a human-in-the-loop (HITL) component for monitoring and controlling the process. The hybrid nature of this architecture results from applying an orchestrator agent for overall planning, while execution and local optimization of tasks occur in a decentralized manner at the level of individual computing clusters under the management of intelligent executor agents. The HITL component allows users or specialists to accept or modify decisions in various parts of the multi-agent system or its interaction with the real execution environment, thereby ensuring the necessary level of control over critical operations.

The proposed multi-agent system for task planning and execution on distributed information systems has a clearly defined scope of application and limitations that are critically important for ensuring the safety and efficiency of its operation. The system is designed exclusively for solving intelligent data processing tasks that do not involve critically important real-time operations or control of physical processes. Such tasks include data analysis for identifying patterns, trends, and anomalies; machine learning and training of artificial intelligence models; natural language processing, including text classification, sentiment analysis, and summarization; computer vision for image classification, object detection, and segmentation; extract, transform, and load (ETL) processes; statistical analysis and forecasting.

As defined in the main objectives of this research, it is necessary to examine and describe the core functions of each agent level (orchestrator and executor), their interactions, and the role and functions of HITL application. A generalization of the proposed solution architecture is presented in Fig. 1.

The primary function of the orchestrator is to decompose and distribute subtasks for data processing among available clusters of the distributed intelligent information system, considering their heterogeneous structure and characteristics. To implement this function, the agent uses the reasoning capabilities of the base LLM, a specialized set of tools, available information about tasks and DIS, as well as HITL component. Fig. 1 shows a high-level schema of the orchestrator's role and the general architecture of the automatic planner.

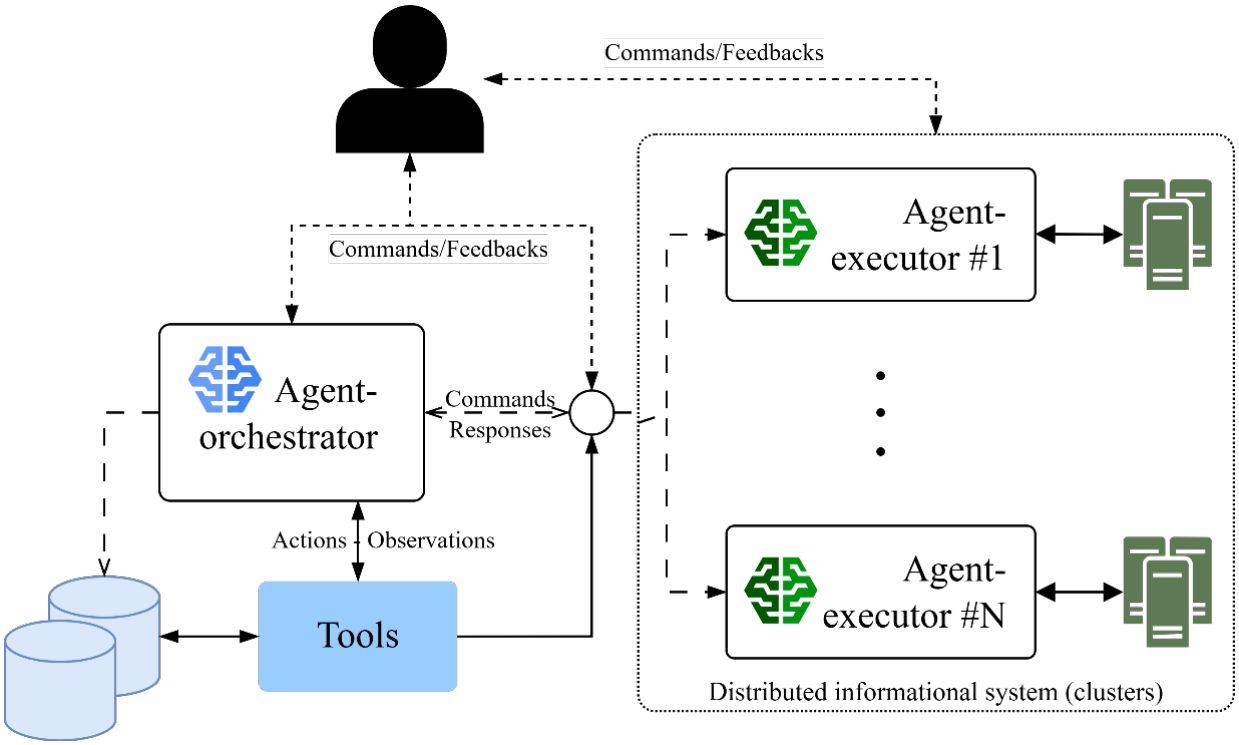

Fig. 1. Visual representation of the orchestrator's role in the proposed multi-agent planner architecture with HITL component

The classical request processing flow includes subtask planning (if applicable), checking available resources (clusters), allocation of subtasks to corresponding executor agents, launching subtask execution on the cluster, and forming the final result. Each executor agent must return confirmation of accepting the subtask allocation to the cluster it controls. If an executor agent does not confirm execution, the orchestrator must perform reallocation.

HITL components at the orchestrator agent level include the following: manual human confirmation of task decomposition into subtasks with the ability to edit the list, confirmation of subtask distribution to clusters, and manual allocation and/or reallocation.

To ensure effective execution, we define a minimally required set of external data and memory sources (accessible via corresponding tools) for the agent, including:

- List and status of DIS clusters and their corresponding executor agents.
- User documentation with extended task description.
- Set of request constants (data references, result storage location references, etc.).
- Agent's long-term memory (previously successfully executed requests).

It is also important to clearly identify the basic and necessary set of tools required for implementing the agent's access to information and task allocation:

- Documentation query tool.
- Current DIS state query tool.
- Long-term memory query tool.
- Tool for sending task allocation requests to agents-executors.
- Subtask reallocation tool.
- Information retrieval tools from databases and long-term memory.
- Tool for sending execution requests for allocated tasks to executor agents.
- Process completion notification tool.
- Error notification tool for informing responsible persons/users (HITL).

The main task of the executor agent is to prepare a software codebase for executing intelligent tasks in a distributed environment (cluster of nodes for which the specific agent is responsible). This involves creating Python programs using libraries that support distributed computing. After preparing (generating and post-processing) program files, the agent initiates their execution on the cluster through a specialized trigger tool.

Considering the previously defined methods of task allocation to agents-executors, the agent must analyze the request from the orchestrator and decide regarding acceptance or rejection of the task. If allocation occurs through manual user request (via HITL), it should be automatically accepted.

Furthermore, the direct language model that serves as the main component of the agent does not necessarily have to be the same for allocation confirmation and code writing, or for executing different subtasks. Adding the HITL component allows for dynamic selection of different base models depending on the task. Visual representation of the described agent-executor can be found in Fig. 2.

Code generation that will subsequently be executed on the user's real distributed infrastructure is a high-risk process without adequate verification levels. In this work, we propose combining two levels of verification – specialist review (HITL) and static code verification in post-processing.

Post-processing includes classical methods of static Python code verification. Specifically, tools such as Pylint allow checking generated code for syntax errors. If errors are found in the code that cannot be automatically corrected, the code is sent back for refinement and correction by the agent.

A mandatory part of generated code verification is confirmation from the HITL component. Users will have the ability not only to check the code but also to request corrections to specific parts or add corrections manually. Such integration of human control provides an additional level of security and quality for generated solutions.

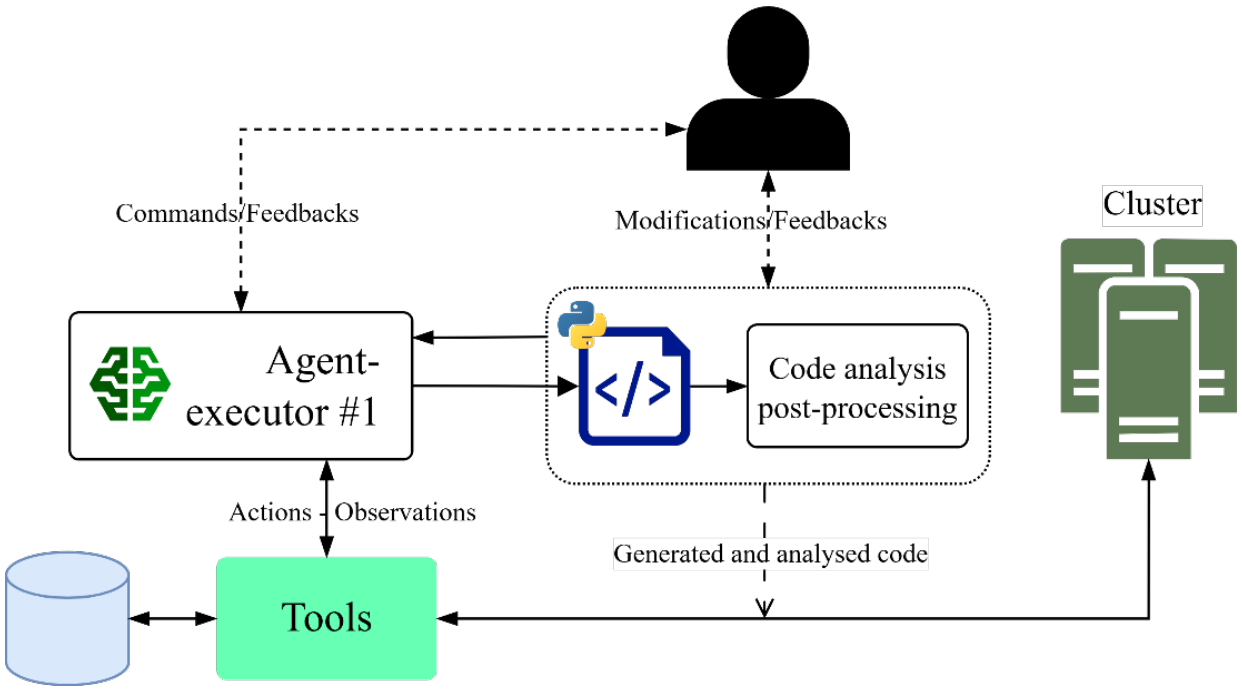

Fig. 2. High-level overview of the agent-executor structure

Summarizing and formalizing the set of tools necessary for implementing these agent tasks, specifically to prepare code for solving subtasks received from the orchestrator:

- Current cluster state query tool.
- Long-term memory query tool.
- Additional documentation database query tool.
- Tool for saving generated code to a specific file.
- Tools for correcting and modifying previously written code.
- Code execution launch tool (with confirmation request from HITL component).

This work makes both practical and scientific contributions to the field of distributed intelligent systems and autonomous task execution. From a practical standpoint, the proposed hybrid multi-agent architecture addresses critical challenges in real-world distributed computing environments where full automation may be insufficient or unsafe. By integrating human-in-the-loop components at strategic decision points, the system provides a practical framework for deploying AI-driven task planning and execution while maintaining necessary human oversight for critical operations. The system's ability to dynamically allocate computational resources across heterogeneous clusters while generating and validating executable code represents a substantial advancement in practical distributed computing solutions, particularly for organizations seeking to leverage AI for data processing tasks without sacrificing operational control.

From a scientific perspective, this research contributes to the theoretical understanding of hybrid human-AI collaboration in multi-agent systems by formally defining the interaction protocols between orchestrator and executor agents, establishing clear boundaries for automated versus human-controlled decision-making, and proposing novel verification mechanisms that combine automated static analysis with human expert review.

Conclusions

This paper presents a comprehensive hybrid multi-agent architecture that effectively combines automated task planning and execution capabilities with essential human oversight for distributed informational systems. The proposed solution addresses the critical need for safe and reliable AI-driven

distributed computing by integrating orchestrator and executor agents with strategically placed human-in-the-loop components at key points including task decomposition, resource allocation, and code verification. Through the combination of automated static code analysis and mandatory human review processes, the system reduces risks for both operational efficiency and safety when executing generated code on real distributed infrastructure. The clearly defined scope of application, focusing on intelligent data processing tasks while explicitly excluding real-time critical operations, establishes appropriate boundaries for safe deployment. The formal characterization of agent interactions, tool specifications, and verification mechanisms provides a robust foundation for implementing reliable distributed AI systems that maintain necessary human control over critical operations while leveraging the scalability and efficiency of automated multi-agent coordination.

REFERENCES

Al-Fraihat, D., Sharrab, Y., Al-Ghuwairi, A.-R., Alzabut, H., Beshara, M., & Algarni, A. (2024). Utilizing machine learning algorithms for task allocation in distributed agile software development. *Heliyon*, *10*(21), e39926. doi: 10.1016/j.heliyon.2024.e39926

Kankaniyage Don, R. T. P., Ravi, A., & Toxtli, C. (2024). *Assessing the Task Management Capabilities of LLM-Powered Agents*. doi:10.13140/RG.2.2.11776.85768

Li, Z., Wu, W., Wang, Y., Xu, Y., Hunt, W., & Stein, S. (2025). HMCF: A Human-in-the-loop Multi-Robot Collaboration Framework Based on Large Language Models. doi: https://arxiv.org/abs/2505.00820

Mao, H., Schwarzkopf, M., Venkatakrishnan, S. B., Meng, Z., & Alizadeh, M. (2019). Learning Scheduling Algorithms for Data Processing Clusters. doi: https://arxiv.org/abs/1810.01963

Mediakov, O., & Khorkanin, M. (2025). Automated Planning in Intelligent Distributed Systems Using a Multi-Agent Approach Based on LLM. *Visnyk of Vinnytsia Politechnical Institute*, *179*(2), 111–117. doi:10.31649/1997-9266-2025-179-2-111-117

Sypherd, C., & Belle, V. (2024). Practical Considerations for Agentic LLM Systems. doi: https://arxiv.org/abs/2412.04093

Takerngsaksiri, W., Pasuksmit, J., Thongtanunam, P., Tantithamthavorn, C., Zhang, R., Jiang, F., ... Wu, M. (2024). Human-In-the-Loop Software Development Agents. doi:10.48550/ARXIV.2411.12924

Zhang, W., Zeng, L., Xiao, Y., Li, Y., Cui, C., Zhao, Y., ... An, B. (2025). AgentOrchestra: A Hierarchical Multi-Agent Framework for General-Purpose Task Solving. doi:10.48550/ARXIV.2506.12508

Zou, H. P., Huang, W.-C., Wu, Y., Miao, C., Li, D., Liu, A., ... Yu, P. S. (2025). A Call for Collaborative Intelligence: Why Human-Agent Systems Should Precede AI Autonomy. doi: https://arxiv.org/abs/2506.09420

ПОЄДНАННЯ ОРКЕСТРУВАННЯ АГЕНТІВ З КЕРУВАННЯМ ОПЕРАТОРОМ ДЛЯ АВТОМАТИЧНОГО ПЛАНУВАННЯ В РОЗПОДІЛЕНИХ СИСТЕМАХ

Максим Хорканін[1], Дмитро Досин[2]

[1, 2] Національний університет “Львівська політехніка”,
кафедра інформаційних систем та мереж, Львів, Україна
[1] Email: maksym.y.khorkanin@lpnu.ua, ORCID: 0009-0001-6263-6878
[2] Email: dmytro.h.dosyn@lpnu.ua, ORCID: 0000-0003-4040-4467

Сучасні розподілені інформаційні системи потребують складних підходів до автоматизації для планування завдань, розподілу ресурсів та управління виконанням. Незважаючи на значний потенціал використання ІШІ агентів на основі великих мовних моделей (LLM) для розв'язання складних завдань розподілених обчислень, повністю автономні рішення представляють суттєві

ризики без належного людського нагляду та механізмів контролю. У цій роботі пропонується комплексна гібридна мультиагентна архітектура, яка ефективно поєднує можливості автоматизованого планування та виконання зі стратегічно інтегрованими компонентами "людина в контурі керування" (HITL) для розподілених інтелектуальних систем. Запропонована архітектура має ієрархічну структуру, що складається з агента-оркестратора та агентів-виконавців, які працюють у координації з людьми-спеціалістами у критичних точках прийняття рішень. Агент-оркестратор керує високорівневою декомпозицією завдань та розподілом ресурсів через гетерогенні обчислювальні кластери, тоді як агенти-виконавці обробляють локальну оптимізацію та генерацію коду для конкретних обчислювальних вузлів. Гібридна природа цієї архітектури виникає внаслідок застосування централізованої оркестрації на рівні завдань при збереженні децентралізованого виконання на рівні підзавдань та окремих кластерів. Ключові інновації включають формальне визначення протоколів взаємодії агентів, набори інструментів як для агентів-оркестраторів, так і для агентів-виконавців, та багаторівневі механізми верифікації, які поєднують автоматизований статичний аналіз коду з обов'язковим експертним рецензуванням людини. HITL-компоненти стратегічно розміщені у критичних вузлах, включаючи підтвердження декомпозиції завдань, схвалення розподілу ресурсів та верифікацію коду перед виконанням на реальній розподіленій інфраструктурі. Цей підхід дозволяє мінімізувати проблему балансу між ефективністю автоматизації та операційною безпекою і контролем. Практична застосовність структури демонструється через чітко визначені відповідальності агентів, специфікації інструментів та механізми взаємодії, які забезпечують безпечне розгортання в реальних середовищах розподілених обчислень. Це дослідження робить як практичний, так і науковий внесок у галузь, надаючи структурований підхід до розгортання рішень розподілених обчислень на основі ШІ при збереженні необхідного людського нагляду, формально характеризуючи гібридну співпрацю людини та системи ШІ та встановлюючи надійні протоколи верифікації, які забезпечують операційну безпеку без жертвування обчислювальною ефективністю.

Ключові слова – автоматичне планування, розподілені інформаційні системи, агнетні системи, мультиагентна взаємодія, людина в контурі керування.