Hochschule Wismar

University of Applied Sciences: Technology, Business, and Design

Summary of Multi-Agent Collaboration Mechanisms: A Survey of LLMs

Supervisor:
Prof. Frank Krüger

Author:
Vraj Chandreshkumar Bhavsar

October 26, 2025

Contents

Chapter 1

Introduction

Due to the excellent capabilities of LLMs to solve complex tasks, they are widely used in various domains. As LLMs are trained on massive amounts of text data, such as web data, they can understand and reason in a human-like language. However, LLMs often lack domain knowledge and struggle with multiple data types, resulting in hallucination.

When a large language model (LLM) is given a distinct goal, memory, and the required tools, it functions as an agent. An agent can retrieve information, create its interpretation of that information, and take actions aligned with its purpose. When LLMs are assigned a specific task and provided with a structured prompt, a single LLM can effectively operate as multiple agents. For instance, if an LLM is prompted with "As a Data Analyst, your goal is to analyse the data and generate statistical insights," it behaves as a data analyst and undertakes the necessary actions based on the assigned objective.

A Multi-Agent System (MAS) is characterised by the collaboration of several intelligent agents who cooperate and exchange information to address complex challenges as a group. This system operates similarly to how humans function in a team, focusing on collective objectives rather than pursuing separate individual goals, enabling them to work together toward a common purpose to achieve results. The primary focus of this report is to understand the theory and collaboration mechanism as discussed in the reference paper.

LLM-Based MAS:

- Distributed their tasks among agents in an environment
- Share knowledge with each other to understand the context
- Execute sub-tasks, and align their actions with the shared objective

Chapter 2

Core Components of Multi-Agent Systems (MASs)

The elements outlined below clarify the overall functionality of the system and describe their composition.

Agents

The core component of a multi-agent system is agents, which are actors responsible for performing assigned tasks based on their roles. Their capabilities, such as reasoning, planning, and decision-making, lend intelligence to the system and themselves.

Environment

An environment is a physical or simulated world where agents are placed. In other words, this is the place where agents perform their tasks and share their knowledge with other agents.

Interactions

This component is responsible for the communication between two or more agents. It defines how they share their understanding. Whether to cooperate or compete with each other, and what structure to use to process the information, is defined by this component. All this is happening using standard communication language or protocol, such as JSON format.

Organisation

It is a social architecture of the system, defining what does what, who reports to whom, how they cooperate, and what rules they follow. They may have hierarchical flow or organised behaviour.

The MASs can solve tasks that are not feasible by LLMs or a single agent, as they work as a team. They possess the inherent ability to learn and make autonomous decisions, offering more flexibility. Their interactions with each other or with the environment to understand the context and actions. Each agent, based on the available context, takes actions in an environment to solve their task.

Chapter 3

Multi-agent collaborative system

3.1 Single Agent

An agent is an autonomous entity defined by a model that governs its reasoning or decision-making process, an objective that directs its behaviour, an environment with which it interacts, and a mapping between inputs (perceptions or observations) and outputs (actions or responses).

Mathematically, an agent is represented as:

$$a = \{m, o, e, x, y\}$$

Where:

- m = model
- o = Objective
- e = Environment
- x, y = Input and output(action) respectiely

Model

The model includes its architecture, memory, and the extra plug-ins called adapters, which can be optionally included. The architecture is the LLM used, which in simple words agent's brain design, such as GPT-4, LLaMA-3, and Claude. The memory is like a personal notebook to store context, goals and previous context, such as databases.

The adapters are additional modules that help the model to specialise in some tasks without retraining the LLM, like LoRA.

$$model = \{architecture, memory, adapter\}$$

Objective

An objective of an agent defines what it is trying to achieve. A specific goal, a reason that guides agents' actions within the system. An agent must have an objective to operate in the system.

$$Objective(o) = \{goal, reward, constraintsrules\}$$

where:

- goal = what the agent wants to achieve.
- reward = reward or evaluation metrics, how it knows it is doing well.
- Constraints = boundaries on what an agent can and cannot do.

Environment

It's a world or a workspace where the agents live and perform it's tasks. The environment gives the agent context, data, and rules that affect what the agent can do.

$$Environment(e) = \{state, action, perception, rules\}$$

where:

- state= The current dataset, what the environment currently looks like.
- action = What actions agents can perform in this environment.
- perception = What data agents can see about the state.
- rule/dynamics = How the environment changes when actions happen.

Collaboration Stages

Each agent(mentioned above) can collaborate with other agents based on the following stages:

1. Early-stage collaboration: Exchanging data, sharing context and environment for model development
2. Mid-stage collaboration: Exchanging parameters, or weights of multiple models.
3. late-stage collaboration: Ensembling outputs/actions to achieve a goal.

Collaboration Mechanism

The collaborative system S is represented as:

$$y_{\text{collab}} = S(O_{\text{collab}}, E, x_{\text{collab}} \mid A, C) = \{c_j\left(\{a_i(o_i, E, x_i) \mid a_i, o_i, x_i \in c_j\}\right) \mid c_j \in C\}$$

where:

- y_{collab} = MASs output.
- O_{collab} = collective set of objectives.
- E = Shared Environment from which agents receive contextual data.
- x_{collab}= collective dataset for the system to be percieved.
- $A = \{a_i\}_{i=1}^{n}$: number of LLM-based agents in the system
- $C = \{c_j\}$: set of Collaboration channel

Collaboration channel

The collaboration channels define the MAS's crucial component, where interactions happen between agents. These channels allow agents to exchange information(data, context, output of other agents) based on objectives, environment and inputs. The core concept of the collaboration mechanism depends on the types, structures and strategies of these channels. If two channels, meaning interaction between two or more agents, have different objectivessenvironmentssnt, or inputs, then the channel is considered as a separate channel. **In a nutshell, a channel is a medium of collaboration between two or more agents.**

Each Communication channel has four properties, defining its behaviour.

1. **Actors:** Agents involved in the channel
2. **Type:** What type of interaction is happening in the channel.
3. **structure:** How agents are organised and how they communicate.
4. **strategy:** How they coordinate

3.2 Collaboration types

Cooperation

Agents in the system work together to achieve a shared goal by aligning their individual goals. They cooperate by exchanging information, validating results, and integrating their outputs, enhancing efficiency, accuracy, and reasoning.

Competition

Agents in the system prioritise their own objective and oppose other viewpoints, conflicting with other systems. Even if they compete with each other, they orient towards the collective goal. For example, a simulated courtroom where each agent is a lawyer works to get justice. This type of collaboration helps in innovation and improves the agent's responses, encouraging advanced reasoning and creative problem-solving.

Coopetition

This type is a mixture of both cooperation and competition types, where they cooperate on shared tasks and compete in others. This type of collaboration is useful in negotiation.

3.3 Collaboration Strategies

A strategy defines how agents decide to act or interact — it's the behavioural logic that guides communication, task execution, and cooperation among agents. Each defines how an agent makes decisions when facing inputs or other agents.

Rule-based Protocol

Rule-based strategy controls the agents' behaviour and coordination by predefined rules. It asks the agent to follow the rules, such as when to act, how to respond to others, what kind of information to share, and how to make decisions. As they are predefined, we can easily trace and debug errors. But this limitation of power prevents agents from being flexible and adaptable.

Role-based Protocol

Role-based protocol mirrors human teamwork, where each agent has already been assigned an individual role and works as an efficient team player to achieve the shared goal. This strategy of division of labour improves efficiency, organisation, and interpretability. This protocol offers flexibility and scalability, as we can add more agents based on a specific goal.

Model-based Protocol

Based on the given input, environment and objective, agents carry out probabilistic decision making. So, instead of reacting to messages or fixed rules, the agent thinks ahead using its internal understanding. In simple terms, one model predicts the other model's behaviour and plans its action accordingly. It offers adaptive and intelligent systems and enables deeper cooperation. Despite advantages, it is complex to implement and requires a higher computational cost.

3.4 Communication Structure

Centralised Topology

In this structure, there is one central agent who organises and manages other agents, assigns tasks, supervises them, aggregates the results of all agents, and makes final decisions. The other agents in the system are subordinates, performing their subtasks and reporting to this central agent.

This structure is easy to design and implement, and efficient for resource allocation. It provides consistent decision-making, making it suitable for small or medium systems. If the central node fails, the entire system might collapse.

Decentralise or Distributed Topology

In a distributed or Decentralised topology, agents communicate and collaborate directly, without relying on a central controller. Each agent works autonomously, inherent in their nature and contributes to the shared goal. This makes it ideal for complex systems where complex problem-solving and data interpretation are required, such as multimodal inputs.

This topology offers robustness and fault tolerance, scalability, flexibility, and better cooperation. Although this topology employs hard synchronisation with all agents. Agents might disagree on conclusions, which could lead to inconsistency, and require more iteration to agree, which leads to longer convergence time.

Heirarchical Topology

This topology mirrors real-world organisation, with higher-level agents responsible for making strategic decisions, assigning tasks, or coordinating multiple low-level agents. On the other hand, Lower-level agents perform specific tasks or execute subtasks assigned by higher-level agents. In simpler terms, this structure organises agents in terms of authority. Hierarchy helps divide complexity, streamline collaboration, and maintain control in multi-agent systems.

Chapter 4

Project Analogy

The CellSense Project involves multiple specialised agents analysing different data modalities(text, images, tabular data) to collectively interpret biomaterials. Each agent contributes to automatically replicating and explaining the research process for cell growth on biomaterials.

Based on my understanding of the referenced paper and the project objectives, I believe the multi-agent system for the CellSense project will likely include 4-5 specialised LLM agents. Each of the agents is briefly described with their role and task in the next section 4.1. These agents are expected to collaborate following a specific type, structure, and strategy, which are elaborated later in this chapter, along with the rationale and illustrative examples justifying each design choice.

4.1 Proposed Agents and Roles

1. **Text Agent:**

   Role: Research Analyst

   Task: Extracts information from scholarly articles, including experiment setup, materials used, and conditions under which cell growth is studied.

2. **Image Agent:**

   Role: Microscopic image Analyst

   Task: Processes and interprets microscopic images to identify and measure cell growth or proliferation on different materials.

3. **Data Agent:**

   Role: Data Analyst

Task: Analyse tabular data extracted from experiments (e.g., growth rate, cell count, surface type) and perform basic statistical comparisons.

4. **Reasoning Agent (Coordinator):**

   Role: Team Leader

   Task: Acts as the central coordinator that integrates outputs from other agents, validates consistency, and generates a comprehensive explanation or summary of findings.

5. **Evaluator Agent:**

   Role: Scientific Reviewer

   Task: Evaluates and provides feedback on the reasoning or conclusions generated by the other agents to ensure accuracy and completeness.

4.2 Collaboration Type —> Cooperative

All agents are working towards the same goal, identifying which material supports better cell growth by replicating the research process. There is no scope for competing with each other to solve the task; instead, they work together and share their interpretation to conclude which material is more suitable.

For example, the Text Agent extracts the experimental conditions from papers, the Image Agent quantifies cell density, Data Agent performs statistical analysis. All this information is shared in the environment, and the reasoning agent integrates it and summarises it.

4.3 Collaboration Structure —> Hybrid (Centralized + Decentralized)

Agents, such as the Data agent, the Image agent, and the text agents, work independently, but we need a coordinator to assign a role to agents. If we make it centralised, then the coordinator will likely be overloaded, and other agents will be too dependent. On the contrary, we cannot keep it completely distributed as we still need one coordinator to gather the data from the environment coming from the different agents and summarise the final output. This indicates the **Hybrid structure** is a good choice.

The coordinator assigns tasks to agents in the system. Each agent performs their task that they are specialised in independently. Once they are done with the given task, they send their outputs back to the coordinator. The coordinator reviews, integrates, and generates the final explanation/report.

4.4 Collaboration Strategy —> Role-Based

Each agent has a clear role and defined responsibility, leading to less confusion and overlap. This strategy reflects how actual research teams operate. Each agent follows its Standard Operating Procedure(SOP) encoded in the prompts.

The Proposed mechanism ensures efficiency as the tasks are distributed properly, Interpretability, where each agent's reasoning is transparent, and Scalability, allowing for the addition of more agents.

Chapter 5

Suitable Frameworks for CellSense MAS

For the CellSense project, choosing a framework is a challenging task. The framework should support:

- Multiple interacting agents
- Role Specialisation
- Data exchange between agents
- Possibly tool use
- Integration with external models or APIs

The referenced paper suggested a few frameworks, including OpenAI Swarm, Microsoft Magentic-One, IBM Bee Agent Framework, and Langchain agents.

5.1 OpenAI Swarm

This framework uses an approach to orchestrating multiple agents through the concepts of routine and handoffs.

"In this framework, an agent is defined as an entity that encompasses specific instructions and tools that are capable of transferring an active conversation to another agent, a process termed a 'handoff.'"

The above-mentioned line is from the referenced paper gives an idea to use this framework to hand off the task of agents to the coordinator.

5.2 Microsoft's Magentic-One System

"At its core is the Orchestrator agent, which is responsible for high-level planning, progress tracking, and dynamic re-planning to recover from errors. The Orchestrator delegates specific tasks to specialised agents."

From the statement above, the structure (Orchestrator + specialised agents) fits CellSense perfectly. We could design: Orchestrator = Coordinator Agent; Specialised = Text, Image, Data agents. Great for modular design and dynamic task allocation.

5.3 IBM's Bee Agent Framework

"It emphasises modularity, extensibility, and production-level control to create sophisticated MASs for a wide range of applications, with future development aimed at enhanced multi-agent orchestration."

Bee supports multi-agent collaboration, state saving (pause/resume experiments), and integration with various LLMs. Perfect for a research workflow simulation like ours.

5.4 LangChain Agents

"These agents are designed to interact with their environment, using tools to process information. LangChain provides a suite of tools and integrations that facilitate the creation of agents capable of complex reasoning and decision-making processes."

Good for building individual agents and chaining logic, but not ideal for large-scale coordination of multiple agents. We can use it for a prototype or internal agent logic, but not for orchestration.