

0001年1月1日 2 分钟阅读

多智能体系统：原理、类型与比较研究

来自智谱清言的“沉思”，prompt是“调查并比较不同的人工智能多智能体系统”

引言

多智能体系统(MAS, Multi-Agent Systems)是人工智能领域中的重要研究方向，它由多个具有自主性的智能体组成，这些智能体能够感知环境并通过交互与协作来实现共同目标。随着人工智能技术的快速发展，多智能体系统在游戏开发、机器人技术、经济模拟和人工智能研究等多个领域展现出广泛的应用前景。本报告将深入探讨多智能体系统的基本概念、分类、关键特征及其在不同应用场景中的表现，并对主流框架进行全面比较，以期对相关领域的研究者和开发者提供有价值的参考。

多智能体系统的基本特征

多智能体系统是由多个自主智能体组成的集合，这些智能体能够感知环境并通过交互与协作来实现共同目标。与传统的单一智能体相比，MAS的核心特征可以归纳为以下几点：

自治性

每个智能体都具备独立决策能力，能够在复杂动态的环境中高效运作。这种自治性使智能体能够独立处理其职责范围内的任务，而无需中央控制系统的持续干预。

分布性

多智能体系统通常采用分布式架构，系统功能由多个智能体共同承担，而非依赖于单一的中心节点。这种分布特性增强了系统的灵活性和容错能力，使得系统能够更好地适应复杂环境中的变化[15]。

目录

文章信息

字数

阅读时间

发布时间

协作性

通过智能体之间的信息交流与协作，系统能够实现资源优化与任务高效执行。在动态环境中，智能体通过共享信息和协调行动，可以更有效地应对复杂问题[15]。

多智能体系统与传统智能体理论的对比

传统的智能体理论往往集中在单一系统的行为控制上，强调反馈、稳定性和可控性等核心概念。控制方法主要包括PID控制、状态反馈与最优控制等，这些方法适用于模型精确、结构明确的系统。相比之下，多智能体系统的控制目标不仅涉及单个智能体的行为控制，更重要的是系统级的性能优化。在MAS中，智能体之间的协作与竞争关系使得控制策略的设计变得更加复杂。控制策略通常是分布式的，智能体根据自身感知的信息和与其他智能体的交互来决策，这加强了MAS在应对动态环境中的灵活性[15]。在信息处理方面，传统的智能体理论强调集中处理的信息和决策，而多智能体系统则更注重局部信息的共享与融合。每个智能体只能感知部分信息，从而通过信息融合技术(如卡尔曼滤波)降低不确定性，提高决策的准确性[15]。

多智能体系统的类型

按交互类型分类

多智能体系统可以根据智能体之间的交互类型进行分类：

- 协作型多智能体系统：**智能体之间通过协作共同完成任务，例如分布式智能电网系统[2]。
- 竞争型多智能体系统：**智能体之间存在竞争关系，例如金融市场模拟系统，每个智能体代表一个交易者。
- 中立型多智能体系统：**智能体独立工作，没有直接的协作或竞争关系，例如分布式传感器网络。

按系统结构分类

多智能体系统也可以根据其结构进行分类：

- 集中式多智能体系统：**有一个中央控制器协调所有智能体的活动，例如某些机器人协调系统。
- 分布式多智能体系统：**没有中央控制器，智能体独立决策并进行局部通信，例如点对点网络系统。

按智能水平分类

根据智能体的智能水平，多智能体系统可分为：

单一高级智能体系统：系统中有一个智能体拥有与其他人不同的高级智能，负责整体协调和决策。

同质智能体系统：所有智能体都具有类似的智能水平，通过协作完成任务。

按应用领域分类

多智能体系统在不同应用领域有其特定的形式和特点：

游戏开发领域：用于创建具有复杂行为的NPC（非玩家角色）。

机器人技术领域：用于编队控制和协作任务执行。

经济模拟领域：用于建模市场行为和经济动态。

人工智能研究领域：用于研究学习和协作的涌现行为。

多智能体系统的关键组成部分

一个典型的多智能体系统通常包括以下几个关键组成部分：

智能体(Agent)

智能体是多智能体系统的基本单元，它可以被定义为“为了完成某个目标，能够接收输入并执行特定操作的系统”^[31]。智能体可以是软件程序、机器人、传感器等，它们各自具备一定的智能和自主性，并处理各自擅长的领域和事情。

交互机制

智能体之间的交互是多智能体系统的核心。交互机制决定了智能体如何通信、共享信息和协调行动。交互机制可以是直接的点对点通信，也可以通过中央控制器进行。

协作机制

协作机制在多智能体系统中的应用尤为关键。通过设置有效的协商协议和博弈策略，智能体能够优化资源分配和任务执行。这在智能交通管理和智能电网等领域表现得尤为明显。在这些环境中，智能体通过互相协作来提高效率，实现整体性能的最优化^[20]。

决策机制

决策机制决定了智能体如何根据环境信息和系统目标做出决策。决策机制可以是基于规则的，也可以是基于学习的，或者两者的结合。

环境模型

环境模型描述了智能体所处的环境特征和变化规律。智能体通过感知环境并根据环境模型做出决策，以适应环境变化。

多智能体系统的工作原理

多智能体系统的工作原理可以概括为以下步骤：

环境感知：智能体通过传感器或输入通道感知环境状态。

信息处理：智能体对感知到的信息进行处理和理解，形成对环境的认知。

决策制定：智能体根据环境认知和系统目标，制定行动决策。

行动执行：智能体通过执行器或输出通道执行决策，与环境进行交互。

反馈学习：智能体根据行动结果获取反馈，并通过学习机制改进决策策略。在实际应用中，这些步骤是循环进行的，系统通过不断感知、处理、决策、执行和学习，逐步适应环境变化并优化系统性能。

多智能体系统的设计与实现

系统架构设计

多智能体系统的设计需要考虑系统的整体架构。从架构上，可以将智能体系统分为两类：

workflow系统(Workflows)：人做整体规划的决策，LLM是链路的一个节点。LLM和各类工具通过预定义的代码路径进行编排。这种架构提供可预测性和一致性，适用于明确定义的任务。

智能体系统(Agents)：LLM做决策，决定任务要怎么做。LLM能够动态指导自己的过程和工具使用，保持对任务完成方式的控制。这种架构适用于需要灵活性和模型驱动决策的场景[14]。

开发实现策略

在开发多智能体系统时，可以考虑遵循以下策略：

简单优先：从最简单的解决方案开始，根据需要增加复杂度，避免过度工程。

渐进式发展：先优化单一LLM调用，添加检索和上下文示例。

必要时使用智能体系统：从Workflow到Agents，Workflow为明确定义的任务提供可预测性和一致性，而当大规模需要灵活性和模型驱动的决策时，可以考虑Agents[14]。

多智能体系统中的通信协议

在多智能体系统中，智能体之间的通信是系统正常运行的关键。常用的通信协议包括：

FIPA-ACL：基于帧的通信协议，支持多种通信模式，如请求-响应、告知-确认等。

JADE协议：Java Agent Development Framework (JADE) 提供的通信协议，支持智能体之间的消息传递和协商。

XMPP：可扩展的消息和表示协议，可用于智能体之间的实时通信。

RESTful API：基于HTTP的通信协议，适用于分布式系统中的智能体通信。

主流多智能体框架比较

随着多智能体系统研究的深入，各种框架和工具不断涌现。以下是对当前流行的多智能体框架的比较分析：

AutoGen (微软)

AutoGen是由微软推出的开创性框架，主要应用于软件开发领域。其核心由两种智能体构成：用户智能体(User-Agent)和助手智能体(Assistant-Agent)。用户智能体负责传达指令和需求，而助手智能体则负责代码的生成与执行，并将成果反馈给用户或其他智能体[1]。 **核心特色：**

卓越的多智能体协调能力，尤其在应对编程任务时表现尤为突出

允许在智能体互动过程中进行人工干预，增加了开发流程的灵活性和可控性

强大的社区支持，来自微软 **不足：**

不适合非编程用户

配置过程较为复杂

在非软件开发领域的表现稍显逊色

CrewAI

CrewAI是一个轻量级的开源Python框架，专门用于构建多模态AI应用。其操作界面直观，即使是没有技术背景的用户也能迅速上手。通过编写提示词，用户可以在几分钟内创建出众多智能体[11]。 **核心特色：**

- 操作界面直观，上手容易
- 支持快速创建智能体
- 适合快速搭建任务演示 **不足：**
- 在灵活性和复杂任务处理上有所局限
- 对技术深度需求不高的项目更为适合
- 文档资料相对匮乏

LangGraph (基于LangChain)

LangGraph是一个高灵活性框架，基于LangChain开发，提供了高度的定制性，几乎能够满足所有多智能体协作应用需求。它采用有向循环图结构，支持高度的定制化，特别适合需要自定义逻辑的高级用户[11]。 **核心特色：**

- 基于LangChain开发，提供高度的定制性
- 采用有向循环图结构，支持复杂的工作流
- 几乎能够满足所有多智能体协作应用需求 **不足：**
- 学习曲线对初学者可能显得较陡
- 文档资料相对匮乏
- 对计算资源要求较高

OpenAI的Swarm框架

Swarm框架专为新手设计，旨在简化智能体的创建和上下文切换流程。用户可以方便地构建演示应用，但其仅支持OpenAI API，这在一定程度上限制了其应用场景[11]。 **核心特色：**

- 简化了智能体的创建过程
- 适合快速开发简单应用
- 用户友好的界面和流程 **不足：**
- 仅支持OpenAI API，应用场景受限
- 技术社区支持不足
- 用户反馈和疑问解决的渠道相对有限

Magnetic-One (微软)

Magnetic-One是微软新推出的框架，旨在简化AutoGen功能，提供更友好的用户体验。它预设了五个智能体，通过便捷的操作支持普通用户。但在灵活性和开源模型支持方面，Magnetic-One仍显得不够成熟，技术社区支持也需改进[11]。 **核心特色：**

- 简化了AutoGen功能
- 提供更友好的用户体验
- 预设五个智能体，操作便捷
- 不足：**
- 灵活性和开源模型支持不足
- 技术社区支持有待改进
- 仍处于发展初期，功能不完善

MetaGPT

MetaGPT是由Meta公司开发的多智能体框架，专注于构建具有协作能力的智能体系统。它采用层次化的智能体架构，支持多种智能体角色和交互模式，适用于复杂的协作任务[13]。 **核心特色：**

- 层次化的智能体架构
- 多种智能体角色和交互模式
- 专注于协作能力的智能体系统

Amazon的Multi-Agent Orchestrator

全球云计算巨头亚马逊云(AWS)开源了5级通用AI Agent框架——Multi-Agent Orchestrator。该框架能通过多层级AI Agent来处理各种复杂任务，将用户的请求找到最适合的Agent来执行，并在整个交互过程中保持上下文的一致性和连贯性。最大亮点之一便是对Python和TypeScript的双语言支持，极大提升了Agent的部署和应用环境，满足不同类型的开发者需求[8]。Multi-Agent主要由5层级组成：

- Orchestrator，核心协调者，管理信息流动处理用户输入，编排响应生成并处理错误场景
- Classifier，检查用户输入、Agent描述和对话历史，选择最适合的Agent来处理请求
- 执行层，负责具体任务的执行
- 监控层，监控系统运行状态
- 学习层，通过反馈优化系统性能

性能比较

以下是不同框架在性能方面的比较：

框架	适用场景	灵活性	易用性	社区支持	性能表现
AutoGen	软件开发	中	低	高	优秀
CrewAI	快速搭建	低	高	中	中等
LangGraph	复杂任务	高	低	中	优秀
OpenAI Swarm	初学者	中	高	低	中等
Magnetic-One	普通用户	低	高	中	中等
MetaGPT	协作任务	高	中	中	优秀
Amazon Multi-Agent Orchestrator	复杂任务	高	中	中	优秀

多智能体系统的应用场景

游戏开发

在游戏开发中，多智能体系统被广泛应用于创建具有复杂行为的NPC（非玩家角色）。通过赋予NPC以智能体特性，可以使游戏中的AI角色表现出更加自然和多样的行为模式，从而增强游戏的沉浸感和可玩性。例如，在多人在线游戏中，可以使用多智能体系统来管理游戏中的各种NPC角色，每个NPC可以具有不同的行为模式和决策逻辑，使游戏世界更加生动和动态。

机器人技术

在机器人技术领域，多智能体系统被应用于机器人编队控制和协作任务执行。多个机器人可以通过智能体之间的通信和协作，共同完成复杂的任务，如搜索与救援、环境监测、工业自动化等。例如，在仓储物流中，可以使用多智能体系统来协调多个机器人完成货物的搬运和分拣任务，通过智能体之间的协作和优化，提高工作效率和资源利用率。

智能交通管理

在智能交通管理中，多智能体系统可以用于模拟和优化交通流量，预测和缓解交通拥堵，提高道路使用效率。每个智能体可以代表一个车辆、一个交通信号灯或一个交通管理者，通过智能体之间的协作，实现交通系统的整体优化。例如，通过智能体之间的信息共享和协调，可以实现自适应交通信号控制，根据实时交通流量动态调整信号灯的时序，减少交通拥堵。

智能医疗

在智能医疗领域，多智能体系统可以用于疾病诊断、治疗方案优化和医疗资源分配。多个智能体可以协同工作，从不同角度分析患者的病情，提供全面的诊断和治疗建议。例如，四川大学华西医院、华西生物医学大数据中心、浙江大学医学院、北京邮电大学等团队，分别基于GPT-3.5和GPT-4，开发了多智能体对话(MAC)框架。该框架由Admin Agent、Supervisor Agent和多个Doctor Agent组成，共同参与患者病情分析。MAC的最佳配置是采用GPT-4作为基础模型，并由4名Doctor Agent和1名Supervisor Agent组成[31]。

经济模拟

在经济模拟中，多智能体系统可以用于建模和分析复杂的经济系统，研究市场行为和经济动态。每个智能体可以代表一个经济主体，如消费者、企业、银行或政府机构，通过智能体之间的交互，模拟经济系统的运行和演化。例如，可以通过多智能体系统模拟股票市场的交易行为，研究不同交易策略对市场波动的影响，为投资决策提供参考。

多智能体系统的挑战与未来发展趋势

当前挑战

多智能体系统在实际应用中面临诸多挑战：

- 通信效率：**随着智能体数量的增加，系统通信开销呈指数级增长，影响系统性能。
- 协调复杂性：**多个智能体之间的协调和协作变得越来越复杂，需要高效的协作机制和决策算法。
- 可扩展性：**如何使系统能够适应不同规模的应用场景，是一个重要的挑战。
- 可解释性：**多智能体系统的决策过程往往缺乏透明度，难以解释和验证。

安全性：多智能体系统面临更多的安全威胁，如智能体被攻击或操纵的风险。

未来发展趋势

多智能体系统未来的发展趋势包括：

自主学习与适应：未来的多智能体系统将更加注重自主学习和适应能力，使系统能够根据环境变化自动调整策略和行为。

边缘计算与多智能体系统结合：边缘计算技术与多智能体系统的结合，将使系统能够在资源受限的环境中高效运行，适用于物联网和嵌入式系统等应用场景。

混合智能体系统：结合基于规则和基于学习的方法，构建更加灵活和高效的混合智能体系统，充分发挥两种方法的优势。

人机协作：未来的多智能体系统将更加注重人机协作，使人类能够与智能体系统无缝交互，共同完成复杂任务。

跨平台与跨语言支持：多智能体系统将提供更好的跨平台和跨语言支持，使开发者能够更灵活地选择开发环境和工具。

结论

多智能体系统作为一种新兴的人工智能技术，正在各个领域展现出广泛的应用前景。通过多个智能体之间的协作与协调，多智能体系统能够更有效地解决复杂问题，实现单个智能体无法完成的复杂目标。本报告对多智能体系统的基本概念、分类、关键特征和应用场景进行了全面的分析，并对主流的多智能体框架进行了比较。研究表明，不同的多智能体系统在结构、功能和性能方面存在显著差异，选择合适的系统需要根据具体的应用需求和场景特点。随着人工智能技术的不断发展，多智能体系统将面临更多的挑战和机遇。未来的多智能体系统将更加注重自主学习、边缘计算、混合方法、人机协作和跨平台支持，为各行业提供更加智能、高效和灵活的解决方案。

参考文献

[2] IDAPS multi-agent system collaborative diagram. https://www.researchgate.net/figure/IDAPS-multi-agent-system-collaborative-diagram_fig4_224408084. [8] 亚马逊开源5级通用AI Agent: 自动查询航班、天气，超4000颗星. <https://new.qq.com/rain/a/20250322A01F5200>. [11] 五大热门 Multi-Agent 智能体开发框架，谁更出色？深度剖析. <https://blog.csdn.net/musicml/article/details/144152618>. [13] 【AI多智能体框架】个人整理的几款AI多智能体框架 - 腾讯云开发者社

区 - 腾讯云. <https://cloud.tencent.com/developer/news/1524339>.
[14] 主 流 多 智 能 体 框 架 设 计 原 理 .
<https://new.qq.com/rain/a/20250314A01GAE00>. [15] 多智能体系统
与传统智能体理论 的比较与发展_应用_决策_环境 .
https://www.sohu.com/a/853404772_121798711. [20] 多智能体系统
与传统智能体理论 的比较与发展_应用_决策_环境 .
https://www.sohu.com/a/853404772_121798711. [31] 模拟医生会
诊,四川大学华西医院团队开发多智能体对话框架助力疾病诊断.
<https://new.qq.com/rain/a/20250402A03S8I00>.

分
享
这
篇
文
章

