

面向智能软件开发的AI Agent：技术架构与智能行为特性深度解析

摘要

本文从技术架构演进与智能行为发展的角度，梳理了基于大语言模型的智能代理现状。归纳了基于LLM的“感知-认知-决策-执行”架构，并深度解析其核心组件、基于检索增强（RAG）的记忆机制。其次，从自主性、持续学习及人机交互等视角，剖析了Agent的行为特征。在此基础上，重点探讨AI Agent在软件开发全生命周期（如代码生成、多智能体协作）中的应用。最后，结合LLM推理能力、跨模态感知等前沿技术，总结了智能体在智能软件开发领域的发展趋势。

关键词： AI智能体；智能软件开发；多智能体协作；大语言模型；检索增强生成

AI Agents for Intelligent Software Development: A Deep Dive into Technical Architecture and Intelligent Behavioral Characteristics

Abstract

This paper surveys the state of large-language-model-based intelligent agents from the dual perspective of architectural evolution and emergent intelligent behavior. We first articulate the “perception–cognition–decision–execution” stack common to LLM-centric agents and dissect its core modules, with particular attention to a retrieval-augmented (RAG) memory subsystem. Next, we characterize the behavioral traits of these agents along three axes—autonomy, continual learning, and human–agent interaction. Building on this analysis, we examine how AI agents can be embedded across the full software-development life-cycle, spanning code generation to multi-agent team orchestration. Finally, we synthesize ongoing advances in LLM reasoning and cross-modal perception to outline future trajectories for agent-assisted intelligent software engineering.

Keywords: AI agents, intelligent software development, multi-agent collaboration, large language models, retrieval-augmented generation

1 引言：AI Agent的演进与智能软件开发新时代

1.1 AI Agent的崛起：从被动式工具到主动智能体

人工智能领域的发展正经历一场深刻的变迁，长期以来，学者们一直在追求与人类相当甚至超过人类水平的AGI，从传统的、依赖人工预设规则的专用工具，正在向具备自主性、强推理能力和环境适应性的通用智能体（AI Agent）演进(Xi et al., 2023; Wang et al., 2024)。传统深度学习模型本质上属于“被动式工具”，其工作模式遵循“输入-处理-输出”流程。在此模式下，用户必须承担“编排者”的角色，将复杂的软件工程任务手动分解为精确的原子指令(Wooldridge & Jennings, 1995)。这种人机交互范式在封闭、静态的场景中表现良好，但受限于预训练数据的分布和固定的行为规则，难以应对动态多变的开发环境或长周期的复杂任务。

与此同时，大型语言模型（LLMs）的出现为智能代理的进一步发展带来了希望。以大语言模型为认知核心的AI Agent 标志着AI应用从“辅助工具”向“合作伙伴”的根本性跃迁(Wei et al., 2022)。学术界通常将AI Agent定义为能够感知环境、进行推理并采取行动以实现目标的智能实体(Hong et al., 2024)。得益于LLM涌现出的思维链推理能力，Agent不再局限于执行单一指令，而是具备了高阶的代理权（Agency）：它能理解抽象的用户意图，将目标拆解为可执行

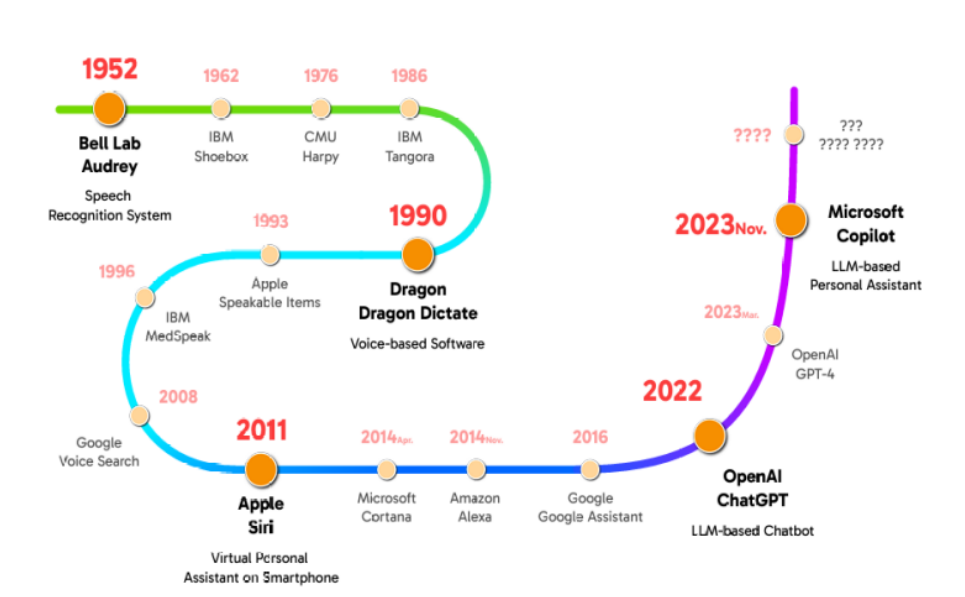


Figure 1: AI agent演进路线

的任务序列，并熟练调用外部工具（如编辑器、API、搜索引擎）来解决问题。这种从“指令驱动”到“意图驱动”的转变，使得Agent 在软件开发中展现出极强的角色特性，极大地重构了人机协作的边界。据预测，到2028 年，超过15% 的日常工作决策将由AI 智能体自主完成，这预示着智能体技术将成为继GUI 之后，重塑软件工程的关键技术。

1.2 智能软件开发的挑战与机遇

随着现代软件系统的发展，软件工程面临着前所未有的复杂性挑战(Storey et al., 2023)。传统的软件开发依赖开发者的经验，且伴随着高强度的负荷。在需求工程、代码实现、质量保障及运维等关键环节中，存在大量耗时且易错的重复性工作，导致显著的开发效率瓶颈和潜在的技术缺陷(Hou et al., 2024)。针对以上问题，具备自主感知与决策能力的AI agent为实现智能化软件开发提供了新的解决方式。与早期的辅助工具不同，AI agent能够深度嵌入软件开发的各个阶段，包括但不限于当前流行的：代码生成与辅助、自动化测试等领域。近年来广泛流行、同时也是我比较关注的是多智能体协作机制的引入，以ChatDev(Qian et al., 2023)和MetaGPT(Hong et al., 2024)为代表的框架证明，通过模拟人类开发团队的组织架构，让扮演不同角色的Agent交互，可以有效缓解单个LLM出现的“幻觉”问题，有望重构软件生产关系。

2 AI Agent的技术架构：智能体的系统

AI Agent 的自主性与适应性源于其独特的技术架构，这与传统软件基于确定性状态机和预设工作流的设计理念存在本质区别(Mialon et al., 2023)。它旨在模拟智能实体感知、思考、决策和行动的认知闭环。这一架构的核心驱动力是大语言模型（LLM），它作为Agent 的大脑或中央处理器，负责对从环境中感知到的信息进行推理和规划(Wang et al., 2024)。

拥有大模型加持的智能代理会是什么样？受到达尔文「适者生存」法则的启发，提出了基于大模型的智能代理通用框架。一个人如果想要在社会中生存，就必须学会适应环境，因此需要具有认知能力，并且能够感知、应对外界的变化。同样，智能代理的框架也由三个部分组成：控制端（Brain）、感知端（Perception）和行动端（Action）(Wang et al., 2024; Xi et al., 2023)。下图来源于复旦大学提出的早期架构模式：作者们用一个例子来说明来了LLM-based Agent 的工作流程：当人类询问是否会下雨时，感知端（Perception）将指令转换为LLMs 可以理解的表示。然后控制端（Brain）开始根据当前天气和互联网上的天气预报进行推理和行动规划。最后，行动端（Action）做出响应并将雨伞递给人类。

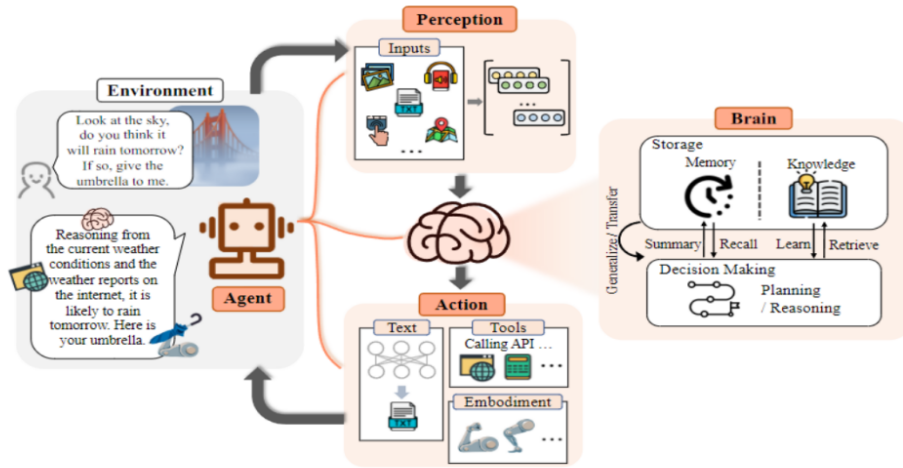


Figure 2: 早期“感知-规划-行动”闭环架构

2.1 感知端（Perception）：多模态环境信息的接收与预处理

通常由LLMs 构成，是智能代理的核心。它不仅可以存储记忆和知识，还承担着信息处理、决策等不可或缺的功能。它可以呈现推理和计划的过程，并很好地应对未知任务，反映出智能代理的泛化性和迁移性(Wang et al., 2024)。在软件开发中，Agent 需要通过此模块“读取”代码库、监控CI/CD 流水线日志，甚至分析由自动化测试失败时捕获的UI 截图(Yang et al., 2024)。这种丰富的环境感知能力是Agent 做出情境感知决策的前提。在此基础上，基于大语言模型的自主智能体统一架构有了更细的分类：

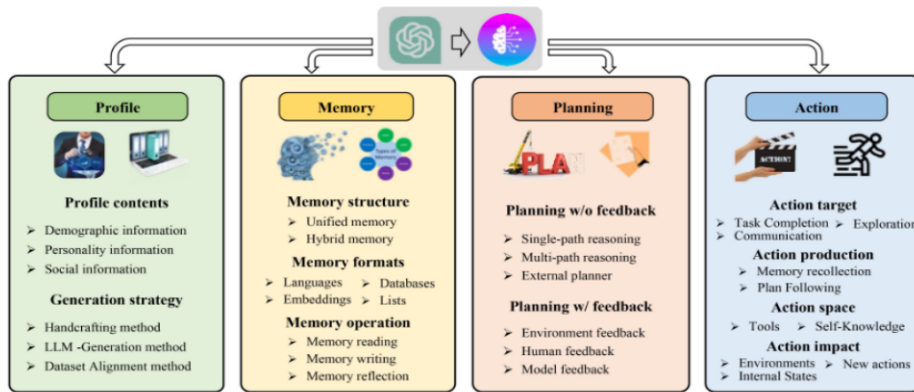


Fig. 2 A unified framework for the architecture design of LLM-based autonomous agent

Figure 3: “Profile - 记忆- 规划- 行动”架构

2.2 规划端(Planning): 认知推理的核心

面对复杂任务时，人类倾向于将其分解为更简单的子任务并逐一解决。规划端旨在赋予agent这种人类能力，使其行为更加合理、强大和可靠。(Wang et al., 2024)。推理能力 (Reasoning) 对于智能代理进行决策、分析等复杂任务而言至关重要。具体到LLMs 上，就是以思维链 (CoT) 为代表的一系列提示方法。而规划 (Planning) 则是面对大型挑战时常用的策略。它帮助代理组织思维、设定目标并确定实现这些目标的步骤(Wang et al., 2024)。在具体实现中，规划包含两个步骤：①计划制定：以ReAct框架为代表(Yao et al., 2023)，代理将复杂任务分解为更易于管理的子任务。例如：一次性分解再按顺序执行、逐步规划并执行、多路规划并选取最优路径等。②计划反思：在制定计划后，可以进行反思并评估其优劣。这种反思一般来自三个方面：借助内部反馈机制；与人类互动获得反馈；从环境中获得反馈(Wang et

al., 2024)。反思机制对于提升规划质量至关重要。Agent 会在任务执行后总结经验教训，将高层次的“洞见”存入记忆库，以避免在未来犯同样错误，这在长周期、复杂的软件项目中尤为关键(Shinn et al., 2023)。

2.3 行动端 (Action)：通过工具调用与环境交互

行动端是AI Agent 影响外部世界的执行器：除了常规的文本输出，还赋予其代理具身能力、使用工具的能力，使其能够更好地适应环境变化，通过反馈与环境交互，甚至能够塑造环境(Wang et al., 2024)。这是Agent 超越纯语言模型的关键，使其从普通的聊天机器人转变为有现实处理能力的工具。该模块的核心技术是工具使用 (Tools use)。LLM 通过函数调用或格式化输出来指定要调用的工具及其参数。这些工具的范围极广，包括：基础工具(搜索引擎、计算器、代码解释器)以及软件工程专用工具(如构建工具、API 客户端、以及与IDE 的集成接口(Hong et al., 2024))。

2.4 记忆端 (Memory)：持续学习的基础

基于大批量语料训练的LLMs，拥有了存储海量知识 (Knowledge) 的能力，而记忆模块 (Memory) 储存了代理过往的观察、思考和行动序列。通过特定的记忆机制，代理可以有效地反思并应用先前的策略，使其借鉴过去的经验来适应陌生的环境。(Wang et al., 2024)。使其从“一次性工具”演变为具备个性化和进化能力的智能实体(Xi et al., 2023)。受人类认知科学的启发，现代Agent 的记忆系统通常采用分层设计，主要包括短期记忆和长期记忆：①短期记忆：也称为工作记忆，它负责缓存当前任务的即时信息，其容量有限，但访问速度极快，是Agent 进行上下文感知交互的基础。②长期记忆：负责存储跨任务、跨会话的经验与知识。其实现通常依赖于向量数据库。主要包括情景记忆、语义记忆和程序记忆。

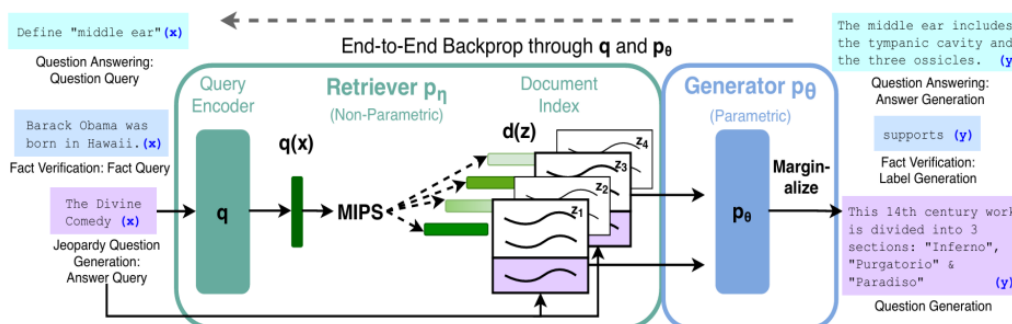


Figure 4: RAG的端到端训练框架

2.5 基于RAG 的记忆检索与利用机制

为了高效地从长期记忆中检索有用信息，检索增强生成 (RAG) 已成为标准技术(Lewis et al., 2020)。其工作流程如下图所示：当Agent 接收到一个新任务时，它首先将任务需求编码为查询向量，在向量数据库中进行语义相似度检索，找出最相关的记忆片段。随后，这些检索到的信息将作为“增强上下文”注入到Prompt 中，引导模型生成更精准、更具事实依据的规划或响应。这一技术在软件开发领域有着极为广泛的应用：设计不同的专用知识库来应对各种需求（如企业内部、医疗管理等）。

3 AI Agent 的智能行为特性：超越传统AI 的核心

AI Agent 的核心不仅在于其技术架构的复杂性，更在于其展现出的、超越传统被动式AI 模型的智能行为特性。这些特性使其能够作为主动的、具备“代理权”的实体在动态、不确定的环境中自主运作(Xi et al., 2023)。构成Agent 核心能力主要有四大行为特征：自主性、学习能力、交互性以及协作性（协作性将在第四章中分析）。

3.1 自主性：从被动执行到目标驱动

自主性是AI Agent 与传统自动化脚本的根本区别，它指的是Agent 在没有持续、微观人类干预的情况下，独立感知环境、更新内部状态、制定并执行行动以实现预设目标的能力(Franklin & Graesser, 1996)。在智能软件开发中，这意味着Agent 能够自主完成需求分析、代码修改、测试验证的全流程，将开发者从繁琐的实现细节中解放出来。LLM 强大的上下文理解与零样本推理能力，使得Agent 无需为每个特定任务硬编码其目标或效用函数，而是能够动态地、情境化地进行决策(Wang et al., 2024)。在感知的基础上，以Reflexion(Shinn et al., 2023)框架为代表的研究引入了自我反思机制，使得Agent 在行动失败后能够分析原因并动态调整后续计划。高度的自主纠错能力。是其实现真正自主性的核心。下图是该框架自我反思机制的示意图：

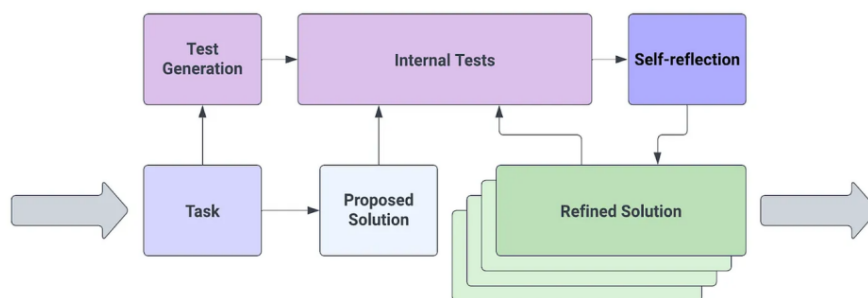


Figure 5: 自我反思架构

3.2 学习能力：从经验中持续进化

学习能力是AI Agent 实现智能和长期价值增长的核心。Agent 能够从环境交互、任务反馈及人类指令中持续学习，从而不断优化其行为策略并适应新的开发体系(Wang et al., 2024)。拥有世界知识的LLMs 赋予智能代理具备强大的迁移与泛化能力。一个好的代理不只是静态的知识库，还应具备动态的学习能力。随着模型规模与训练数据的增大，LLMs 在解决未知任务上涌现出了惊人的能力。通过指令微调的大模型在zero-shot 测试中表现良好，在许多任务上都取得了不亚于专家模型的成绩。(Brown et al., 2020)。这意味着Agent 无需针对新任务进行模型参数微调，仅通过在Prompt 中提供清晰的指令或少量示例，就能快速泛化到未见过的软件开发任务上。持续学习的主要挑战是灾难性遗忘，即当模型学习新任务时容易丢失过往任务中的知识。专有领域的智能代理应当尽量避免丢失通用领域的知识。(Parisi et al., 2019)。当模型为适应新任务而更新参数时，可能会破坏已学到的旧知识。现代Agent 架构通过外部记忆模块来有效缓解此问题。如检索增强生成（RAG）机制将知识和经验存储在外部向量数据库中，而非模型参数内部。当需要时，通过检索将相关知识注入上下文（见图4）。

3.3 交互性：数字世界中的“具身”与多模态

人类通过多模态的方式感知世界，所以研究者们对LLM-based Agents 抱有同样的期待。多模态感知能加深代理对工作环境的理解，显著提升了其通用性。(Wang et al., 2024)。而具身（Embodiment）是指代理与环境交互过程中，理解、改造环境并更新自身状态的能力。具身行动（Embodied Action）被视为虚拟智能与物理现实的互通桥梁(Wang et al., 2024)。现代Agent 不再仅仅处理文本，而是“具身”于特定的数字环境中，如操作系统、Web 浏览器或IDE (Xie et al., 2024)。此外，还可以为代理引入对更广阔的整体环境的感知：采用激光雷达、GPS、惯性测量单元等成熟的感知模块。这种数字具身性赋予了Agent 两大核心能力：情境化感知和可扩展的工具使用：主动分析外界环境并调用相关API进行操作。下图指明了交互性是解决“幻觉”问题的关键。

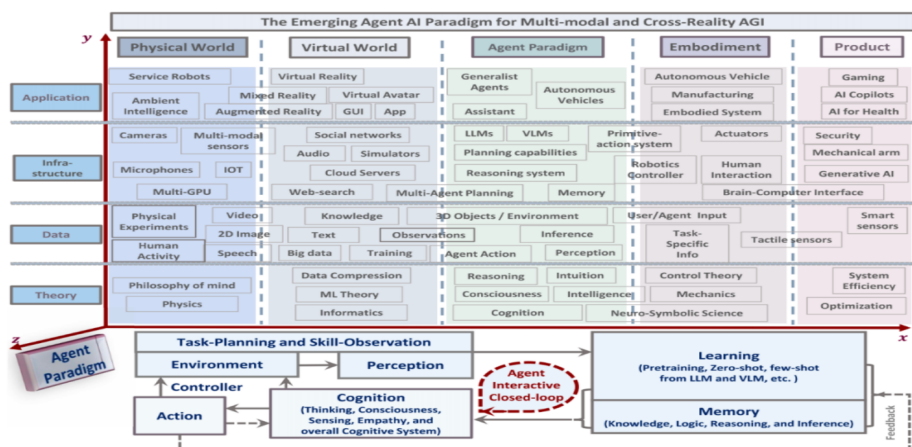


Figure 6: 智能体在多模态中的地位

4 AI Agent 在智能软件开发中的应用

在当前发展下，LLM-based Agents 已经展现出令人瞩目的多样性和强大性能。AutoGPT、MetaGPT、CAMEL 以及GPT Engineer 等耳熟能详的应用实例正在以前所未有的速度蓬勃发展(Wang et al., 2024)。

4.1 代码工程：从辅助编码到自主开发

代码工程是Agent 应用最成熟的领域，以GitHub Copilot等为代表的AI 编码辅助工具，被视为Agent 的前身。它们通过大规模代码预训练，具备了实现上下文感知的代码补全、函数生成等能力。虽然这些工具大多缺乏自主规划能力，但它们极大地提升了开发者的编码效率。同时，AI Agent 能够理解高阶的自然语言需求，自主完成端到端的应用开发。能够执行复杂的代码修改和重构任务。在代码维护方面，它们能够自动分析代码库，生成高质量的文档和注释。当代码库发生变更时，能够自动更新相关文档。

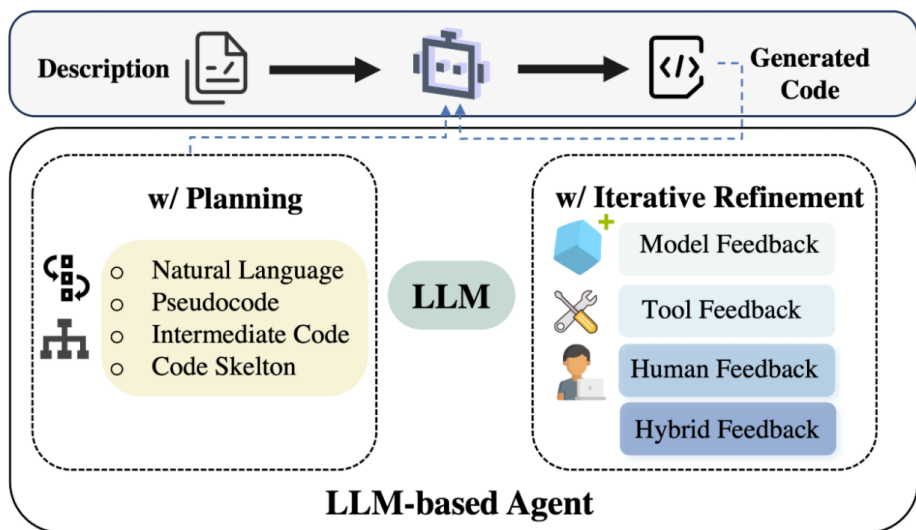


Figure 7: AI agent在代码工程的应用

4.2 质量保障：自动化测试与缺陷管理

软件质量保障是另一个受益于AI Agent 的关键领域。Agent 能够覆盖测试的全生命周期，实现从用例生成到缺陷修复的闭环。基于对需求文档和代码逻辑的深度理解，Agent 能

够生成比传统模糊测试更具针对性的测试用例。Test-Agent(Deng et al., 2023) 等研究表明, 通过让Agent 扮演“探索性测试员”的角色, 能够发现更多由复杂业务逻辑导致的深层Bug。当测试失败时, Agent 能够自动分析堆栈跟踪、错误日志和运行时状态, 快速定位缺陷根源。AgentFL(Zhang et al., 2023)展示了LLM 在项目级大规模故障定位中的潜力。SWE-agent(Yang et al., 2024) 通过模拟开发者使用命令行工具修复GitHub 仓库中的真实Bug, 验证了Agent 进行自主代码修复的可行性。

4.3 多智能体协作开发: 模拟社会化软件生产

早在1986 年, Marvin Minsky 就做出了具有前瞻性的预言。他在《心智社会》(The Society of Mind) 一书中提出了一种新颖的智力理论, 认为智力是在许多较小的、特定功能的代理的相互作用中产生的。例如, 一些代理可能负责识别模式, 而其他代理可能负责做出决策或生成解决方案。这一想法随着分布式人工智能的兴起得到了具体实践。多代理系统(Multi-Agent System) 作为其中主要的研究问题之一, 主要关注代理们如何有效地协调并协作解决问题(Wang et al., 2024)。

在我看来, 多智能体协作正在成为热门的发展方向。它并非简单地将任务分配给多个Agent, 而是通过模拟人类软件团队的社会化协作流程, 来应对复杂系统级开发任务的挑战。下图是“角色扮演”的一个典型例子:

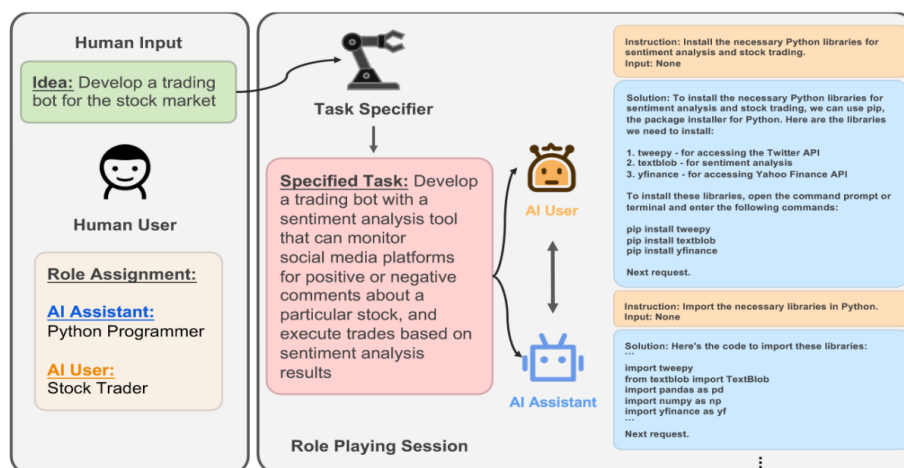


Figure 8: 多角色AI 智能体协作开发流程

最近的研究已经展示了利用基于LLM的多智能体解决各种任务的有希望的结果, 例如软件开发: ChatDev (Qian et al., 2023) 和 MetaGPT (Hong et al., 2024) 等。在这种背景下, 多个自治智能体协作参与规划、讨论和决策, 反映了人类群体工作在解决问题任务中的合作性质。这种方法利用了LLMs的沟通能力, 以及它们生成文本进行沟通并响应文本输入的能力。另一类多智能体则协作专注于构建高效的技术迭代闭环。以AgentCoder(Zha et al., 2023) 为例, 其协作方式可概括为“生成-测试-优化”。一个“程序员”Agent 负责生成代码, 一个“测试员”Agent 负责生成并执行测试用例, 一个“调试器”Agent 则根据测试失败的反馈对代码进行修改。通过持续的质量反馈来提升代码的正确性。

5 前沿技术趋势与未来展望

AI Agent技术正处于飞速发展的阶段, 其能力边界和应用场景在不断被拓宽。未来的AI Agent将不再是单一功能的工具, 而是具备更强通用性、自主性和协同能力的智能系统。大型语言模型(LLM) 的为这些智能体提供了强大的“大脑”, 多模态感知等技术, 使其能够更深入地理解和作用于现实世界。在智能软件开发领域, 这些前沿技术趋势预示着软件开发将迎来更深层次的变革, 向全流程、自主化、人机融合的智能化开发新时代迈进。

5.1 大型语言模型（LLM）作为通用智能体核心

LLM 作为Agent 的认知核心，其发展趋势是从特定领域的模型向通用解决工具演进。未来的LLM 将具备更长的上下文窗口、更强的多步推理能力和更低的幻觉率。通过与RAG、工具调用等机制的深度融合，以LLM 为核心的Agent 将能够自主处理从需求分析到部署运维的软件全生命周期任务。同时智能代理能否真正落地，还需要经过严谨的安全性评估，避免对真实世界带来危害。下图可以看到，基于LLM的智能体领域的论文数量有着显著的增长。

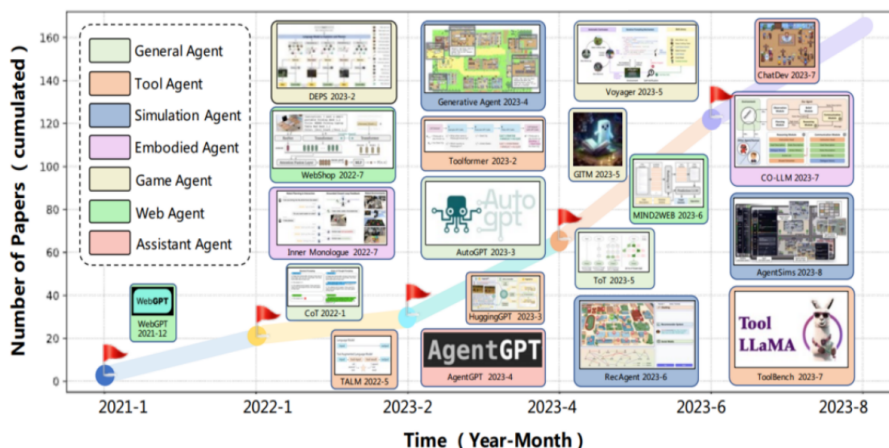


Figure 9: 2021–2023 年 LLM 自主智能体领域论文增长趋势与类别分布

5.2 跨模态感知与交互（图6所示）

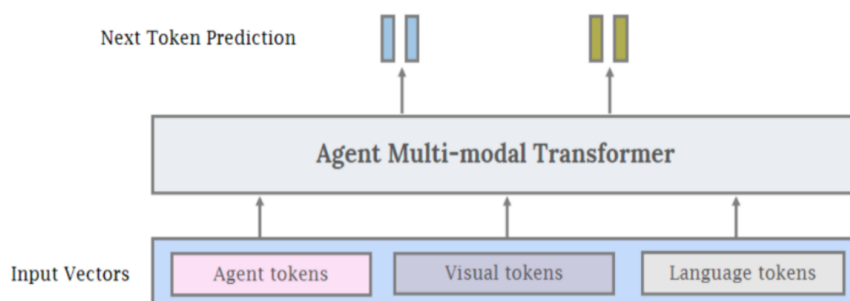


Figure 10: 智能体多模态Transformer 架构

传统的基于强化学习的Agent 在样本效率、泛化性和复杂问题推理等方面存在局限性，而LLM-based Agents 通过引入大模型丰富的内在知识，使得Embodied Agent 能够像人类一样主动感知、影响物理环境(Xi et al., 2023; Wang et al., 2024)。这种跨模态能力使其能够更全面地理解复杂的现实世界场景，并以更自然、更富表现力的方式与人类进行交流。智能体可以通过分析设计稿（视觉）和听取口头描述（听觉），来生成符合要求的UI代码。在协作开发中，开发者可以通过语音指令和屏幕共享，与AI Agent进行实时的、多模态的交互，共同调试代码或设计架构。Google的ADK已经内置了对双向音频和视频流的支持，预示着未来智能体交互将更加丰富。

5.3 自主进化的智能体系统

另一趋势是构建能够自主进化与自我完善的Agent 系统。这不仅是“终身学习”，更是向“自编程”的迈进：Agent 不仅学习解决任务，更能学习改进自身的代码、架构甚至学习算法(Chen et al., 2023)。这种能力最终可能催生出能够自主发现并应用全新算法或架构的、真正具有创造

性的AI。下方的智能体演进路线图也证明了这一点：多智能体自进化是当前发展趋势，也是人工智能领域进步的关键时期。

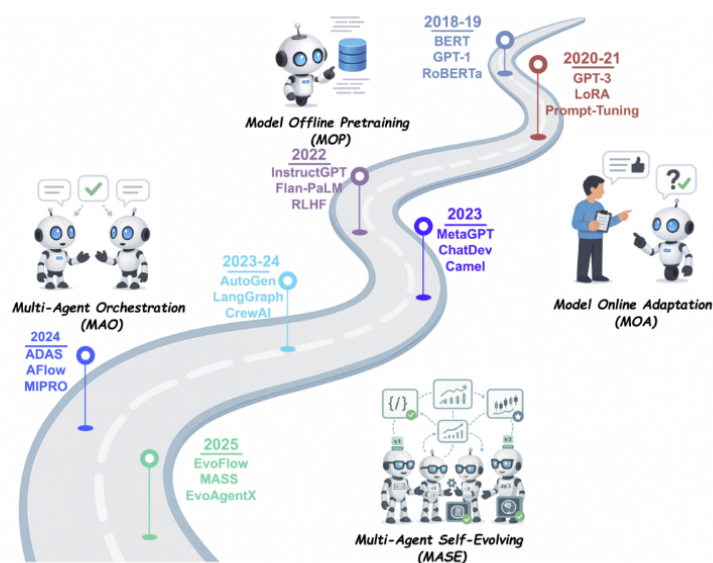


Figure 11: LLM 智能体技术演进路线图：从离线预训练到多智能体自进化

6 结论

本文通过分析AI Agent技术架构与智能行为特性，揭示了新一代智能系统的核心本质。以模块化设计为基础，通过感知、认知、决策、执行、记忆等核心组件的有机协同。其展现出的高级智能行为，如自主性、持续学习和多智能体协作，是其复杂技术架构在功能层面的集中体现。这种技术架构与智能行为的融合与统一，是AI Agent区别于以往普通AI系统的根本所在。在智能软件开发领域，AI Agent正引发一场深刻的变革。从被动的辅助工具，转变为主动的、能够深度参与软件全生命周期的智能伙伴。从智能编码、自动化测试到多智能体协作开发，AI Agent的应用正在将开发者从大量重复性、机械性的劳动中解放出来，使其能够专注于更具创造性的工作。这种变革不仅极大地提升了软件开发的效率和质量，更预示着未来软件开发模式将朝着更加智能化、自动化的方向发展。开发者与AI Agent的协作，将成为未来软件开发的主流形态。综上所述，AI Agent作为人工智能领域的前沿技术，正在以其独特的架构，为智能软件开发领域注入活力。随着技术的不断成熟和应用场景的不断拓展，AI Agent必将深刻地改变我们的工作方式，引领软件开发进入一个全新的智能化时代。

参考文献

- Brown, T., et al. 2020. *Language models are few-shot learners*. Advances in Neural Information Processing Systems, 33:1877–1901.
- Chen, Y., et al. 2023. *Self-Programming: A new paradigm for general-purpose AI*. arXiv preprint arXiv:2312.07450.
- Deng, G., et al. 2023. *Test-Agent: A General Framework for Building Test-Issue-Reproducing Agents*. arXiv preprint arXiv:2310.13800.
- Franklin, S., & Graesser, A. 1996. Is it an agent, or just a program?: A taxonomy for autonomous agents. *Proceedings of the First International Workshop on Agent Theories, Architectures, and Languages*, 81–95.
- Gao, Y., et al. 2024. *Agent AI: Surveying the Horizons of Multimodal Interaction*. arXiv preprint arXiv:2403.15003.

- Hong, S., et al. 2024. *MetaGPT: Meta-programming for a multi-agent collaborative framework*. arXiv preprint arXiv:2310.01888.
- Hou, X., et al. 2024. *Large Language Models for Software Engineering: A Survey*. arXiv preprint arXiv:2402.00825.
- Lewis, P., et al. 2020. *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks*. Advances in Neural Information Processing Systems, 33:9459–9474.
- Li, Y., et al. 2024. *Personal LLM Agents: Insights and Survey about the Capability, Efficiency and Security*. arXiv preprint arXiv:2405.17886.
- Mialon, G., et al. 2023. *GAIA: A benchmark for general AI assistants*. arXiv preprint arXiv:2311.15348.
- OpenAI. 2023. *GPT-4V(ision) System Card*. Technical Report.
- Parisi, G. I., et al. 2019. *Continual lifelong learning with neural networks: A review*. Neural Networks, 113:54–71.
- Qian, C., et al. 2023. *ChatDev: Communicative Agents for Software Development*. arXiv preprint arXiv:2307.07924.
- Shinn, N., et al. 2023. *Reflexion: An autonomous agent with dynamic memory and self-reflection*. arXiv preprint arXiv:2303.11366.
- Storey, M. A., et al. 2023. *The future of software engineering work*. IEEE Software, 40(5):58–64.
- Sutton, R. S., & Barto, A. G. 2018. *Reinforcement learning: An introduction*. MIT Press.
- Wang, L., et al. 2024. *A Survey on Large Language Model based Autonomous Agents*. arXiv preprint arXiv:2402.07867.
- Wei, J., et al. 2022. *Chain-of-thought prompting elicits reasoning in large language models*. Advances in Neural Information Processing Systems, 35:24824–24837.
- Wooldridge, M., & Jennings, N. R. 1995. *Intelligent agents: Theory and practice*. The Knowledge Engineering Review, 10(2):115–152.
- Xie, T., et al. 2024. *OSWorld: Benchmarking Multimodal Agents for Open-Ended Tasks in Real Computer Environments*. arXiv preprint arXiv:2403.08311.
- Xi, Z., et al. 2023. *The Rise and Potential of Large Language Model Based Agents: A Survey*. arXiv preprint arXiv:2306.08907.
- Yang, H., et al. 2024. *AppAgent: Multimodal Agents as Smartphone Users*. arXiv preprint arXiv:2402.06339.
- Yang, C., et al. 2024. *SWE-agent: Agent-Computer-Interface for Software Engineering*. arXiv preprint arXiv:2402.16158.
- Yao, S., et al. 2023a. *ReAct: Synergizing reasoning and acting in language models*. The Eleventh International Conference on Learning Representations.
- Yao, S., et al. 2023b. *Tree of Thoughts: Deliberate Problem Solving with Large Language Models*. arXiv preprint arXiv:2305.10601.
- Zha, K., et al. 2023. *AgentCoder: Multi-Agent-based Code Generation with Iterative Testing and Optimization*. arXiv preprint arXiv:2311.17337.
- Zhang, L., et al. 2023. *AgentFL: A Fault Localization Framework for AI Agents*. arXiv preprint arXiv:2312.02135.
- Zhou, Y., et al. 2024. *A Comprehensive Survey of Self-Evolving AI Agents: A New Paradigm Bridging Foundation Models and Lifelong Agentic Systems*. arXiv preprint arXiv:2402.07891.