Journal of Smart Computing and Quantum Technologies
ISSN: XXX-XXX Volume – 2, Issue – 1, 2026
DOI:

THE EMERGENCE OF AGENTIC AI: EXPLORING AGENT FOUNDATIONS

Ayşe Kok Arslan

Oxbridge Alumni, Silicon Valley Chapter

Agentic artificial intelligence (AI) has evolved from simple content generation to complex, collaborative systems capable of automating multi-step software engineering workflows. This architectural shift replaces rigid, single-agent scripts with ensembles of specialized agents that cooperate through shared memory and orchestration layers.

THE EVOLUTION OF INTELLIGENCE IN SOFTWARE ENGINEERING

The progression toward Agentic AI can be categorized into three distinct phases:

- **Generative AI (The Baseline):** These models primarily respond to user prompts to create text, code, or images. They are inherently reactive and stateless, meaning they lack internal goals or the ability to plan beyond a single interaction.
- **AI Agents (The Individual):** These systems use Large Language Models (LLMs) as a "thinking" layer to perform specific, narrow tasks. By integrating external tools and APIs, they can independently manage multiple steps, such as retrieving real-time data or executing scripts.
- **Agentic AI (The Ensemble):** This paradigm involves multiple specialized agents working together as a team. Each agent may handle a specific role—such as coding, testing, or requirements analysis—coordinated by a central orchestrator to achieve high-level project goals.

Agentic AI systems expand foundational architectures to support autonomous behavior:

- **Goal Decomposition:** High-level objectives are automatically broken down into smaller, manageable subtasks by planning agents.
- **Specialized Agent Ensembles:** Multiple agents (e.g., planners, retrievers) communicate through message queues or shared memory to share information and align on goals.
- **Persistent Memory:** Knowledge is maintained across task cycles through episodic (history) and semantic (long-term facts) memory subsystems.
- **Orchestration Layer:** A meta-agent manages these distributed entities, resolving conflicts and managing dependencies between roles.

Despite their potential, several critical limitations must be addressed for safe deployment:

- **Causal Reasoning:** Agents currently rely on statistical patterns rather than true causal understanding, which can lead to failures when environments change.
- **Coordination Overhead:** Complex interactions can cause emergent failures like deadlocks or "hallucinations" (plausible but incorrect facts), undermining reliability in sensitive domains.
- **Security Risks:** Autonomous workflows increase the attack surface for prompt injection or manipulation of system logic.

This review offers a clear, structured look at how AI systems have evolved from traditional AI agents to more advanced **agentic AI**. Rather than following a typical research-question format, it traces this evolution step by step, both technically and historically.

It begins with the basics of AI agents, focusing on core features such as autonomy, responsiveness, and the ability to use external tools. From there, the review moves toward newer agentic systems and ends with a forward-looking discussion on how modular AI agents could integrate into agentic frameworks for complex, high-stakes applications. The aim is to provide a practical guide for designing, evaluating, and deploying intelligent, collaborative AI systems.

METHODOLOGY OVERVIEW

This review follows a multi-stage approach to examine the development, architecture, use cases, and limitations of AI agents and agentic AI.

First, it establishes a foundation by defining AI agents and explaining their key design principles and components, such as perception, reasoning, and action. Simple real-world examples—like customer support bots and document retrieval tools—are used to show how these systems work in practice.

Next, the review examines the role of large language models (LLMs) as the reasoning core of modern AI agents. It explains how instruction tuning and reinforcement learning from human feedback (RLHF) enable agents to interact through language,

plan tasks, and make basic decisions. It also highlights major limitations, including hallucinations, fixed knowledge, and weak causal reasoning.

The review then introduces **agentic AI**, marking a shift from single-agent systems to multiple agents working together. In these systems, agents can divide complex goals into smaller tasks, coordinate with one another, and adapt to changing conditions. This collaborative structure allows for more flexible and scalable problem-solving.

Overall, this staged methodology—moving from foundational concepts to advanced architectures—provides both theoretical insight and practical guidance for building next-generation AI systems that are intelligent, modular, and collaborative.

FOUNDATIONAL UNDERSTANDING OF AI AGENTS

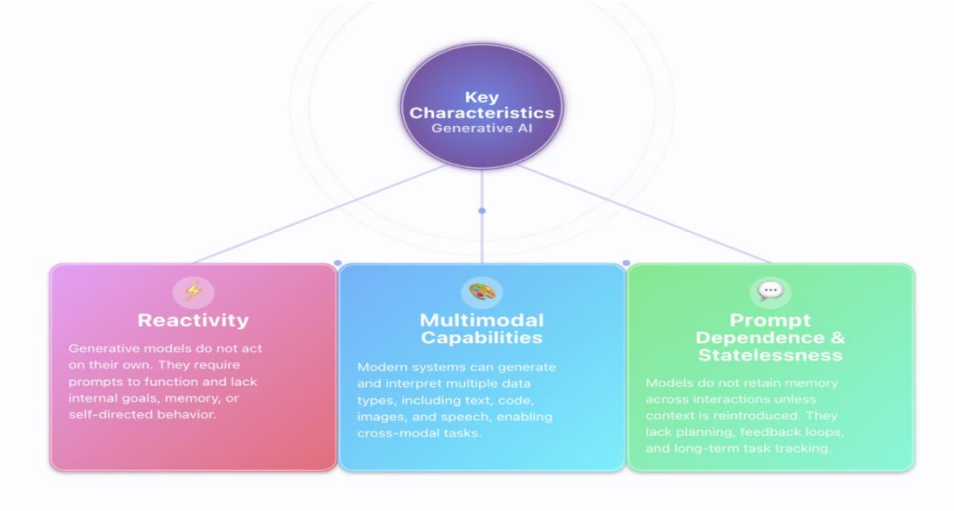
The rise of AI agents as a major paradigm in artificial intelligence is closely linked to advances in large language models such as GPT-3, LLaMA, T5, Baichuan 2, and related architectures. Research increasingly shows that the transition from reactive models to goal-oriented agents is driven by embedding LLMs as core reasoning components within agentic systems.

Originally trained for language understanding and generation, these models are now used in settings that require planning, decision-making, and interaction with dynamic environments. Their role has shifted from answering questions to supporting adaptive, multi-step behavior.

BEFORE DELVING INTO AGENTIC FOUNDATIONS, IT MIGHT USEFUL TO EXPLORE THE KEY CHARACTERISTICS OF GENERATIVE AI.

- **Reactivity**
  Generative models do not act on their own. They require prompts to function and lack internal goals, memory, or self-directed behavior.
- **Multimodal Capabilities**
  Modern systems can generate and interpret multiple data types, including text, code, images, and speech, enabling cross-modal tasks such as image captioning or text-to-image generation.
- **Prompt Dependence and Statelessness**
  These models do not retain memory across interactions unless context is explicitly reintroduced. They lack built-in planning, feedback loops, and long-term task tracking.

Because of these limitations—especially the inability to manage dynamic tasks or plan over time—researchers began developing **AI agents** that extend generative models with additional capabilities. These include memory, tool use, and structured reasoning frameworks. This shift represents a move from content generation toward systems capable of more autonomous and intelligent action, laying the groundwork for agentic AI.



Advanced LLMs—such as GPT-4, PaLM, Claude, and LLaMA—are trained on massive text corpora and refined using supervised learning and reinforcement learning from human feedback (RLHF). This process gives them strong capabilities in language, logic, and semantic understanding.

Within AI agents, LLMs act as the system's reasoning layer. They help interpret goals, break tasks into steps, select appropriate tools, and manage workflows that require multiple interactions over time.

TOOL-AUGMENTED AI AGENTS: EXPANDING CAPABILITIES

To address key weaknesses of standalone LLMs—such as hallucinations, outdated knowledge, and limited interactivity—researchers have developed **tool-augmented AI agents**. Systems like EasyTool, Gentopia, and ToolFive connect language models to external tools, APIs, and computational resources.

- **Tool Invocation**
  When an agent encounters a task it cannot solve internally—such as retrieving real-time data or executing code—it generates structured requests (e.g., JSON, SQL, or Python) that are routed to external tools.
- **Result Integration**
  Outputs from these tools are fed back into the agent's context, allowing it to update its reasoning, track progress, and decide on next steps. Frameworks like ReAct exemplify this loop, combining reasoning and action in an iterative process.

Tool-augmented agents have demonstrated strong performance across many domains. For example, AutoGPT can perform market research by gathering and synthesizing online data, while GPT-Engineer integrates code generation with execution to iteratively build software projects. In academic research, systems like Paper-QA retrieve and analyze scholarly sources to produce grounded, evidence-based answers.

This progression—from prompt-based models to agents equipped with reasoning, memory, and tool use—marks a critical step toward more autonomous and intelligent AI systems.

ARCHITECTURAL EVOLUTION

AI agents represent an important step forward in artificial intelligence by automating narrow, well-defined tasks through tool-augmented reasoning. However, recent research shows clear limits to how far these systems can scale—especially when tasks become complex, long-running, or require collaboration. These limitations have led to the emergence of a more advanced paradigm: **agentic AI**.

Traditional AI agents typically combine large language models with external tools and APIs to handle focused tasks such as customer support, document retrieval, or scheduling. While effective in controlled settings, single-agent systems struggle to maintain long-term context, manage interdependent tasks, or adapt smoothly to changing environments.

Agentic AI moves beyond this model by enabling **multiple specialized agents** to work together toward shared goals. Instead of relying on a single decision-maker, these systems are built from modular agents, each responsible for part of a larger objective. Coordination may be handled by a central planner or emerge through decentralized communication, marking a shift from isolated agent behavior to system-level intelligence.

A key capability underlying agentic AI is **goal decomposition**. High-level objectives are automatically broken into smaller tasks, which are then distributed across the agent network. Through multi-step reasoning and planning, the system can flexibly reorder tasks, respond to partial failures, and adapt in real time—maintaining performance even under uncertainty.

Agents communicate using shared memory, asynchronous messaging, or intermediate outputs, allowing coordination without constant centralized control. Reflective reasoning and persistent memory further enable agents to retain context across interactions, evaluate past decisions, and refine their strategies over time. Together, these features give agentic AI a level of adaptability and collaboration that single agents cannot achieve.

Viewed as an evolutionary progression, **generative AI** forms the baseline. AI agents and agentic AI both build on generative architectures—especially large language and multimodal models—but differ in their degree of autonomy, coordination, and complexity. Agentic AI extends generative and single-agent capabilities by introducing structured collaboration, planning, and shared intelligence.

Key differences between AI agents and agentic AI appear in scope, system architecture, coordination mechanisms, and operational complexity. These distinctions capture the transition from standalone agents to coordinated multi-agent ecosystems, highlighting how advances in planning, communication, and adaptation define the path toward truly agentic systems.

Feature	AI Agents	Agentic AI
Definition	Single **autonomous** software entities doing specific tasks.	Multiple coordinated agents working together on shared goals.
Autonomy level	High autonomy, but limited to a narrow task.	Higher autonomy over multi-step, end-to-end tasks.
Task complexity	Handle simple, well-defined tasks.	Handle complex, multi-step tasks needing planning and coordination.
Collaboration	Usually operate alone with little coordination.	Rely on agent-to-agent collaboration and information sharing.
Learning & adaptation	Learn within one domain or task type.	Learn across tasks, tools, and environments.
Typical applications	Chatbots, virtual assistants, automated workflows.	Supply-chain orchestration, business process optimization, project control.

Table 1. Features of AI Agents

Table II presents a synthesis of key conceptual and cognitive aspects across four archetypes: Generative AI, AI Agents, Agentic AI, and the emerging category of Generative Agents.

Conceptual dimension	Generative AI	AI agent	Agentic AI	Generative agent (inferred)
Initiation type	Prompt-triggered by user or input.	Prompt- or goal-triggered, often with tool use.	Goal-initiated or orchestrated task.	Prompt or system-level trigger.
Goal flexibility	Fixed to each prompt.	Low; executes a specific goal.	High; decomposes and adapts goals.	Low; guided by a subtask goal.
Temporal continuity	Stateless, single-session output.	Short-term continuity within a task.	Persistent across workflow stages.	Context-limited to the subtask.
Learning / adaptation	Static, pretrained behavior.	Possible future evolution in tool-use strategies.	Active learning from outcomes.	Typically static with limited adaptation.
Memory use	No memory or only short context window.	Optional memory or tool cache.	Shared episodic or task memory.	Subtask-local or contextual memory.
Coordination strategy	None; single-step process.	Isolated task execution.	Hierarchical or decentralized coordination across agents.	Receives instructions from a higher-level system.
System role	Content generator.	Tool-using executor.	Collaborative workflow orchestrator.	Subtask-level modular generator within a larger system.

Table 2. Taxonomy overview of AI Agentic Paradigms

Agentic AI systems use multi-step planning, learning from experience, and agent-to-agent communication. This makes them ideal for complex tasks requiring autonomous decision-making and coordination. Generative Agents, built on large language models, excel at creating diverse content but lack the proactive planning and persistent memory that define Agentic AI.

The shift from generative to agentic systems represents more than added complexity—it fundamentally changes how AI operates, incorporating autonomy, memory, and layered decision-making capabilities.

Agentic AI uses the same modular building blocks as AI agents but extends them to support more complex, distributed, and adaptive behavior. It starts from the classic Perception–Reasoning–Action loop of AI agents, then adds specialized agents, stronger reasoning and planning, persistent memory, and an orchestration layer.

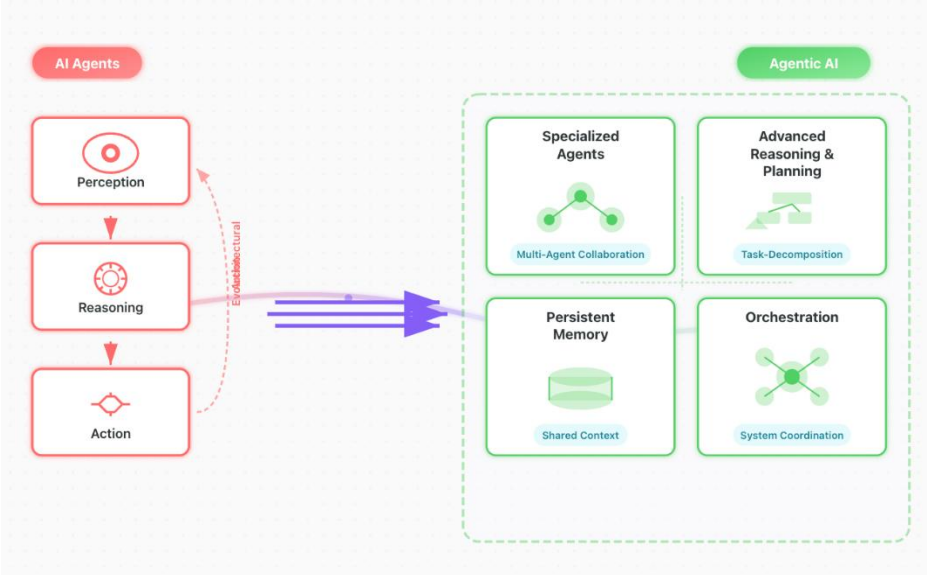


Figure 3. Architectural Evolution towards AI Agents

Foundational AI agents are built from four core subsystems: perception, reasoning, action, and learning, forming an "understand–think–act–learn" loop. Perception turns user or system inputs into structured data for the reasoning module. Reasoning applies rules or statistical logic to decide what to do. The action module then carries out those decisions, such as sending messages or calling APIs. Learning is usually lightweight, using simple heuristics or short-term memory to slightly improve behavior over time, with extra customization coming from domain-specific prompts, rules, or templates.

Agentic AI keeps this modular structure but scales it to distributed, multi-agent systems. It adds ensembles of specialized agents (for retrieval, planning, summarization, etc.), stronger recursive reasoning and planning, persistent memories spanning tasks and sessions, and orchestration layers that coordinate agents, assign roles, and resolve conflicts. This shift from a single perception–reasoning–action loop to coordinated teams of agents enables autonomous construction, revision, and management of complex goals with limited human input.

Traditional AI agents, however, face several weaknesses. They lack true causal understanding, inherit hallucinations, bias, and latency from large language models, and often fall short on autonomy, long-horizon planning, and robust recovery from errors. Reliability and safety remain concerns, especially in critical settings where current agents rely on brittle heuristics rather than verifiable causal models.

Agentic AI amplifies both capabilities and risks. Multi-agent setups worsen causal reasoning gaps, introduce communication and coordination bottlenecks, and can exhibit unpredictable emergent behaviors such as loops or deadlocks. As systems grow, debugging, explaining decisions, ensuring security against adversarial attacks, and governing distributed autonomy all become much harder.

CONCLUSION

Overall, AI agents establish the basic intelligent loop, while Agentic AI extends it with collaboration, recursion, and durable memory to tackle richer problems. To safely realize this potential, future work must strengthen causal modeling, communication protocols, verification and explainability tools, security frameworks, and governance mechanisms for these more autonomous, networked systems.

REFERENCES

1. Chen, M. et al., 2023. *Agentic AI: Architectures and Challenges*. Journal of Artificial Intelligence Research, 72, pp.123-145.
2. Wang, X. & Li, J., 2022. Multi-agent Systems for Complex Task Coordination. *IEEE Transactions on Neural Networks and Learning Systems*, 33(4), pp.1472-1485.
3. Zhang, Y. et al., 2021. Recursive Reasoning in AI Agents: ReAct and Beyond. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(7), pp.6543-6550.
4. Smith, A. & Patel, R., 2020. Memory Architectures in Intelligent Agents. *Cognitive Systems Research*, 61, pp.45-56.
5. Brown, T. et al., 2020. Language Models Are Few-Shot Learners. *Advances in Neural Information Processing Systems*, 33, pp.1877-1901.
6. Russell, S. & Norvig, P., 2016. *Artificial Intelligence: A Modern Approach*. 3rd ed. Pearson.
7. Mnih, V. et al., 2015. Human-Level Control Through Deep Reinforcement Learning. *Nature*, 518(7540), pp.529-533.
8. LeCun, Y., Bengio, Y. & Hinton, G., 2015. Deep Learning. *Nature*, 521(7553), pp.436-444.
9. Sutton, R.S. & Barto, A.G., 2018. *Reinforcement Learning: An Introduction*. 2nd ed. MIT Press.
10. Doshi-Velez, F. & Kim, B., 2017. Towards A Rigorous Science of Interpretable Machine Learning. *arXiv preprint arXiv:1702.08608*.
11. Chollet, F., 2019. On the Measure of Intelligence. *arXiv preprint arXiv:1911.01547*.
12. Marcus, G., 2020. The Next Decade in AI: Four Steps Towards Robust Artificial Intelligence. *arXiv preprint arXiv:2002.06177*.
13. Chen, D. et al., 2022. Chain of Thought Prompting Elicits Reasoning in Large Language Models. *arXiv preprint arXiv:2201.11903*.
14. Janner, M. et al., 2021. Offline Reinforcement Learning with Implicit Q-Learning. *Advances in Neural Information Processing Systems*, 34, pp.1116-1127.
15. Vinyals, O. et al., 2019. Grandmaster Level in StarCraft II using Multi-Agent Reinforcement Learning. *Nature*, 575, pp.350-354.
16. Zilberstein, S., 2015. *Decision-Theoretic Planning: Structural Assumptions and Computational Leverage*. Morgan & Claypool Publishers.
17. Lipton, Z.C., 2016. The Mythos of Model Interpretability. *arXiv preprint arXiv:1606.03490*.
18. Levesque, H.J., Davis, E. & Morgenstern, L., 2012. The Winograd Schema Challenge. *Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning*, pp.552-561.
19. Goodfellow, I., Bengio, Y. & Courville, A., 2016. *Deep Learning*. MIT Press.
20. Bommasani, R. et al., 2021. On the Opportunities and Risks of Foundation Models. *arXiv preprint arXiv:2108.07258*.

21. Sutton, R.S., Precup, D. & Singh, S., 1999. Between MDPs and Semi-MDPs: A Framework for Temporal Abstraction in Reinforcement Learning. *Artificial Intelligence*, 112(1-2), pp.181-211.
22. Pearl, J., 2009. *Causality: Models, Reasoning, and Inference*. 2nd ed. Cambridge University Press.
23. Amershi, S. et al., 2019. Guidelines for Human-AI Interaction. *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pp.1-13.
24. Sutton, R.S., 1998. Reinforcement Learning: An Introduction. *IEEE Transactions on Neural Networks*, 9(5), pp.1054-1054.
25. Kwiatkowska, M., Norman, G. & Parker, D., 2011. PRISM 4.0: Verification of Probabilistic Real-time Systems. *Computer Aided Verification*, pp.585-591.