



AGENTIC AI: THE AI THAT TALKS AND ACTS

Ms. Hole S.S.¹, Kartiki Jadhav²

¹ Faculty Guide & Supervisor | ² Department of Computer Engineering, India

Corresponding author: kartikijadhav2004@gmail.com

ARTICLE ID IJREK-2026-00170	ARTICLE TYPE Research Article	RESEARCH AREA Artificial Intelligence	PUBLICATION Volume 1, Issue Issue 1 (2026)
RECEIVED 26 October 2026	REVISED 29 October 2026	ACCEPTED 28 October 2026	PUBLISHED 29 October 2026

Abstract

Most AI tools only talk: they answer a question and stop. Agentic AI systems go further, because they plan steps, call tools and act on other software. This is useful, but a wrong answer from a chatbot is only text, while a wrong action from an agent can change a database or move money. This paper reviews how agentic systems are built, where they are used and how well they perform in published tests. We reviewed 29 sources (research papers, open standards and official rules) under six themes: reasoning, tool use, memory, protocols, safety and business use. Acting helps—ReAct beat earlier methods by 34% and 10% on two benchmarks—yet success on realistic tasks is still low: the best web agent completed 14.41% of tasks against 78.24% for humans. From the gaps found we propose a five-layer framework with a human-approval gateway and check it analytically against published cases. No software was built, and all numbers are quoted from the original authors. We conclude that agents work best when humans keep control of important decisions.

Keywords: Agentic AI; AI agents; large language models; ReAct; human-in-the-loop; Model Context Protocol

I. INTRODUCTION

Most people first met AI through chatbots: you type a question, the model writes an answer, and its work is over. Agentic AI is the next step. An agent is given a goal instead of a single question, and it decides which steps to take, which tools to use and when to stop. Russell and Norvig describe an agent as something that senses its environment and acts on it [1], an idea studied since the 1990s [2]. What changed recently is that large language models (LLMs) [3] became good enough to act as the “brain” of such an agent. Put simply, generative AI asks “what should I create?”, while agentic AI asks “what should I do next?”. An LLM alone has limits: a finite context window, no knowledge of real-time events, and no way to work inside other applications unless software connects it. When an AI starts to act, mistakes also become more serious. A wrong chatbot answer is ignored; a wrong agent action may already have changed a database, sent an email or paid an invoice. Organizations therefore need clear designs, safety rules and controls. This paper (i) reviews 29 sources on LLM agents and collects the evidence in one table, (ii) draws three research gaps from the review, (iii) proposes a five-layer framework with a human-approval gateway, and (iv) checks the framework analytically against published cases. This is a literature-based study; we built and tested no software, so every performance number is quoted from the original authors. Section II describes the method, Section III reviews the literature, Section IV presents the framework, Section V evaluates it, and Sections VI and VII give limitations, future scope and conclusions.

II. REVIEW METHOD

Agentic test. The word “agent” is used loosely, so we apply a simple test: a system is agentic when it meets at least four of five points—(1) it works toward a goal, not a single prompt; (2) it plans ordered steps; (3) it calls tools, APIs or databases itself; (4) it keeps short- and long-term memory; and (5) it has rules or human approval for risky steps. A plain chatbot meets only the first, whereas a coding assistant that plans, runs tests and asks approval before merging meets all five. Sources and analysis. We used a structured narrative review, following the transparency ideas of PRISMA 2020 [6] (stating sources, inclusion rules and analysis) without a full

systematic review, so no database hit counts are reported. Material came from peer-reviewed papers, preprints from recognised groups, technical documentation of open standards and official regulations, found using terms such as agentic AI, LLM agents, ReAct, tool use, Model Context Protocol, human-in-the-loop and AI governance. A source was included if it reported evidence or a clear technical design, concerned LLM-based agents or their governance, and had an identifiable author. Marketing material and opinion pieces were excluded. Of the 29 sources, 15 are journal or conference papers, 4 are arXiv preprints, 3 are industry documents, 6 are standards, laws or reports, and 1 is a textbook. Each source was assigned to one of six themes, and the goal, method, result and limits of key studies were recorded (Table I). The gaps found were turned into design requirements for the framework.

III. LITERATURE REVIEW

A. Reasoning and Acting A model that only thinks cannot check its thoughts against the world, so it can invent facts and carry mistakes forward. Yao et al. [7] proposed ReAct, in which the model writes a “thought”, takes an “action” such as a search, and reads the “observation” before thinking again (Fig. 1). On ALFWorld and WebShop it beat imitation and reinforcement learning methods by an absolute 34% and 10% in success rate, as reported by the authors. The written thoughts also let a person read and check what the agent did.

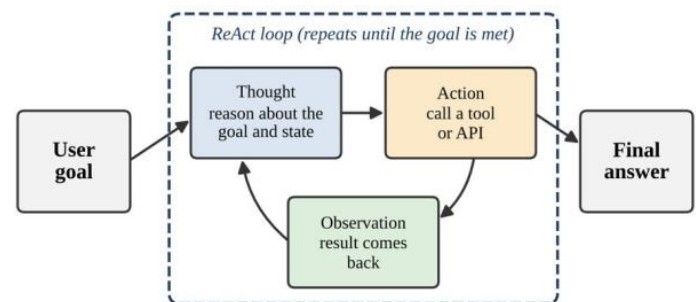


Fig. 1. The ReAct loop (thought, action, observation).

TABLE I Evidence Map of Key Studies (As Reported) Ref. Topic Reported finding Limit [7] Reason + act +34% / +10% over IL / RL Benchmarks only [8] Tool use Better zero-shot results No tool chains [12] Architecture One structure fits many agents No experiments [14] Memory 91% vs 80% (GPT-4), HumanEval Needs good tests [19], [20] GitHub issues 1.96% best, then 12.5% Python only [21] Web tasks 14.41% agent vs 78.24% human Simulated sites [23] Security Hidden text can steer an app Defences open

E. Safety, Security and Regulation Greshake et al. [23] showed indirect prompt injection: instructions hidden in a web page or document can steer an LLM application that reads it, without the user typing anything harmful. OWASP [24] lists “excessive agency”—giving a system more power than it needs—as a top risk. The NIST AI Risk Management Framework [25] treats governance as a function spanning the whole AI life cycle, the EU AI Act [26] requires human oversight of high-risk AI systems, and India's Digital Personal Data Protection Act, 2023 [27] applies when agents handle personal data. ISO/IEC 42001 [28] sets requirements for an AI management system inside an organization. F. Research Gaps Three gaps emerge. (1) Reliability and cost: agents are far below human level on realistic tasks, and cost is rarely reported. (2) Real deployments: most evidence comes from benchmarks and prototypes, with few rigorous studies in real company workflows. (3) Governance: the risks are known, but few practical designs show how approval policies, confidence thresholds, audit logs and human checkpoints fit into one architecture.

IV. PROPOSED FRAMEWORK

A. Five-Layer Architecture Each gap was turned into requirements and placed in the architecture of Fig. 2. Layer 1 (interface and goal intake) takes the goal and shows a plan preview. Layer 2 (planning and orchestration) splits the goal into tasks and routes them to agents. Layer 3 (agent core) holds the reasoning model, memory and tool selector. Layer 4 (tools and integration) contains MCP servers, APIs, databases and workflow tools such as n8n. Layer 5 (governance and safety) holds the policy gateway, human approval and audit log. Every tool call from Layer 3 must pass Layer 5 before reaching Layer 4.

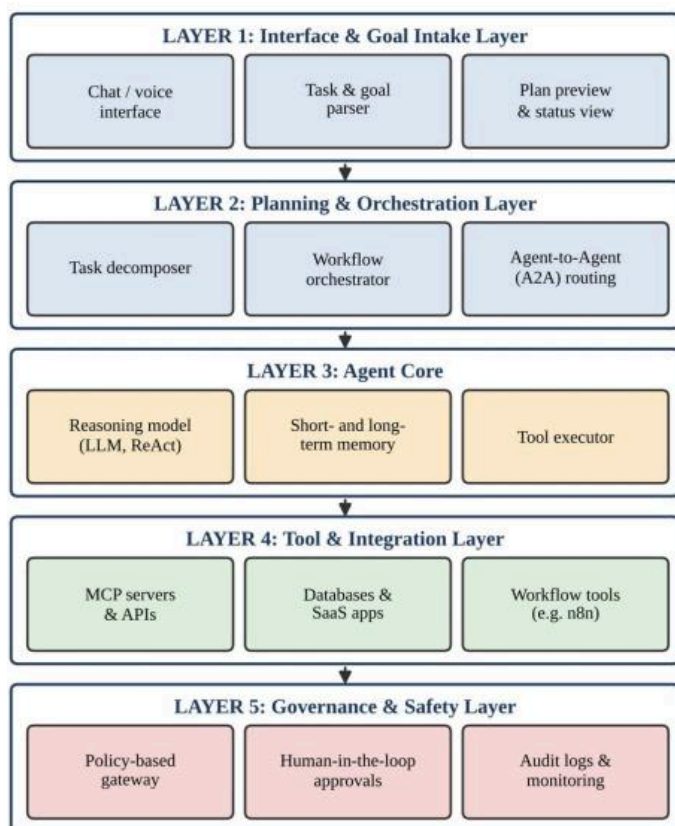


Fig. 2. Proposed five-layer architecture.

$\text{review}(a) = [\text{r}(a) = \text{high}] \text{ OR } [\text{c}(a) < \tau]$ (1) Otherwise it runs automatically. A reviewer can approve, reject or edit the action, every decision is logged, and a

rejection sends the agent back to re-plan. A high τ means more reviews and more safety; a low τ means more autonomy. Example. An accounts-payable agent is given the goal “process this supplier invoice”. Reading the invoice and comparing it with the purchase order are low-risk and run alone. Releasing a payment or changing a vendor's bank details is hard to undo, so the rule marks it high-risk and a human sees a short summary first. A blurry scan gives low confidence, so that case also goes to a person. Every step is logged with the approver's name.

V. APPLICATIONS AND ANALYTICAL EVALUATION

A. Applications and Cross-Case Insights The sources show three main application areas: business workflows (onboarding, supplier payment checks, IT incidents), software engineering (task splitting, code and test generation, with engineers reviewing merges) and digital automation (monitoring email or tickets, summarising documents, linking systems with tools such as n8n). Four published cases show where agents work and where they fail: ReAct [7] showed that acting reduces invented facts; Generative Agents [13] showed what memory adds, though in a simulated town; Reflexion [14] showed that selfreflection raises coding results, although people still write the tests; and SWE-bench and SWE-agent [19], [20] showed that tool interface design matters as much as the model and that real success rates are modest. Across all cases, a human seeds the goal, writes the tests or reviews the output. Jobs reports expect agents to take over some tasks within a job rather than remove whole jobs [29]. B. Why Small Errors Matter Because the framework is not implemented, we evaluate it on paper. Suppose an agent completes each step correctly with probability p , independently. A task of n steps is then fully correct with probability

$P(\text{success}) = p^n$ (2) This is plain arithmetic, not a measured result, and real steps are not fully independent. It still explains the gap in Section III-D: even 95% accuracy per step gives only about 59.9% for ten steps and 21.5% for thirty. If a human checks only the risky steps, unchecked errors are limited to low-risk steps that can be undone, which is why Eq. (1) lets the agent work alone where mistakes are cheap and keeps a person in charge where they are not. C. Framework Against the Criteria Five criteria were taken from the gaps. Goal-to-action flow is provided by Layers 1–2 and the approve-or-re-plan loop, as in ReAct [7]. Risk control comes from the policy gateway and Eq. (1), which answer prompt injection and excessive agency [23], [24]. Reliability and cost are addressed by a step limit and budget in Layer 2, in response to the weak SWE-bench and WebArena

RESULTS

[19], [21]. Interoperability comes from MCP and A2A in Layer 4 [9], [10]. Auditability comes from the Layer 5 log, which supports the NIST and EU requirements [25], [26]. Following [15], the simplest approach that solves the problem (chat assistant, fixed workflow, single agent, then multi-agent team) should be chosen, adding autonomy only where flexibility is needed.

VI. LIMITATIONS AND FUTURE SCOPE

Advantages. Agents finish multi-step work, reducing time on routine steps; written thoughts and logs make their path easy to check; open standards let tools change without redesign; and the approval gateway keeps humans in charge of costly or irreversible actions. Limitations. Success on realistic tasks is still low and small errors add up. Cost and delay can grow because agents may loop or retry. Agents read untrusted content while holding real permissions, which allows prompt injection. Too many approvals remove the benefit of autonomy, and too few allow harmful errors. Our findings also depend on how the original authors reported their data, most evidence comes from benchmarks rather than long-running company systems, the framework itself is untested, and the field changes fast.

FUTURE SCOPE. (I) PILOT THE FRAMEWORK IN A REAL

organization and report success, cost and humanintervention rate; (ii) study approval between agents in A2A teams; (iii) add and test prompt-injection defences in the gateway; (iv) measure how well LLM confidence separates right from wrong actions; and (v) develop shared standards for audit logs so governance can be checked across vendors.

VII. CONCLUSION

Agentic AI moves AI from answering to acting. Our review of 29 sources showed that acting, memory and good tool interfaces make agents more capable, but success on realistic tasks remains far below human level and small errors compound over many steps. We identified three gaps—reliability and cost, real-deployment evidence, and governance—and proposed a five-layer framework with a policy-based approval gateway, checked analytically against published cases. All performance numbers are quoted from the original authors and the framework is not yet tested. An AI that talks and acts is most useful when humans stay in control of the important decisions; a real-world pilot is the next step.

REFERENCES

- [1] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed. Hoboken, NJ, USA: Pearson, 2021.
- [2] M. Wooldridge and N. R. Jennings, "Intelligent agents: Theory and practice," *Knowl. Eng. Rev.*, vol. 10, no. 2, pp. 115–152, 1995.
- [3] A. Vaswani et al., "Attention is all you need," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 30, 2017.
- [4] L. Ouyang et al., "Training language models to follow instructions with human feedback," in *NeurIPS*, vol. 35, 2022.
- [5] J. Wei et al., "Chain-of-thought prompting elicits reasoning in large language models," in *NeurIPS*, vol. 35, 2022.
- [6] M. J. Page et al., "The PRISMA 2020 statement: An updated guideline for reporting systematic reviews," *BMJ*, vol. 372, art. n71, 2021.
- [7] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafraan, K. Narasimhan, and Y. Cao, "ReAct: Synergizing reasoning and acting in language models," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2023, arXiv:2210.03629.
- [8] T. Schick et al., "Toolformer: Language models can teach themselves to use tools," in *NeurIPS*, vol. 36, 2023, arXiv:2302.04761.
- [9] Anthropic, "Introducing the Model Context Protocol," Nov. 2024.
- [10] Google, "Announcing the Agent2Agent Protocol (A2A)," Google Developers Blog, Apr. 2025.
- [11] P. Lewis et al., "Retrieval-augmented generation for knowledgeintensive NLP tasks," in *NeurIPS*, vol. 33, 2020.
- [12] T. R. Sumers, S. Yao, K. Narasimhan, and T. L. Griffiths, "Cognitive architectures for language agents," *Trans. Mach. Learn. Res.*, 2024, arXiv:2309.02427.
- [13] J. S. Park, J. C. O'Brien, C. J. Cai, M. R. Morris, P. Liang, and M. S. Bernstein, "Generative agents: Interactive simulacra of human behavior," in *Proc. ACM UIST*, 2023, arXiv:2304.03442.
- [14] N. Shinn, F. Cassano, E. Berman, A. Gopinath, K. Narasimhan, and S. Yao, "Reflexion: Language agents with verbal reinforcement learning," in *NeurIPS*, vol. 36, 2023, arXiv:2303.11366.
- [15] Anthropic, "Building effective agents," Dec. 2024.
- [16] Q. Wu et al., "AutoGen: Enabling next-gen LLM applications via multi-agent conversation," arXiv:2308.08155, 2023.
- [17] L. Wang et al., "A survey on large language model based autonomous agents," *Front. Comput. Sci.*, vol. 18, no. 6, art. 186345, 2024, arXiv:2308.11432.
- [18] Z. Xi et al., "The rise and potential of large language model based agents: A survey," arXiv:2309.07864, 2023.
- [19] C. E. Jimenez et al., "SWE-bench: Can language models resolve realworld GitHub issues?" in *Proc. ICLR*, 2024, arXiv:2310.06770.
- [20] J. Yang et al., "SWE-agent: Agent-computer interfaces enable automated software engineering," in *NeurIPS*, vol. 37, 2024, arXiv:2405.15793.
- [21] S. Zhou et al., "WebArena: A realistic web environment for building autonomous agents," in *Proc. ICLR*, 2024, arXiv:2307.13854.
- [22] S. Kapoor, B. Stoebl, Z. S. Siegel, N. Nadgir, and A. Narayanan, "AI agents that matter," arXiv:2407.01502, 2024.
- [23] K. Greshake, S. Abdelnabi, S. Mishra, C. Endres, T. Holz, and M. Fritz, "Not what you've signed up for: Compromising real-world LLM-integrated applications with indirect prompt injection," arXiv:2302.12173, 2023.
- [24] OWASP Foundation, "OWASP Top 10 for LLM Applications 2025," 2025.
- [25] National Institute of Standards and Technology, "Artificial Intelligence Risk Management Framework (AI RMF 1.0)," NIST AI 100-1, 2023.
- [26] European Union, "Regulation (EU) 2024/1689 of the European Parliament and of the Council laying down harmonised rules on artificial intelligence (Artificial Intelligence Act)," *Off. J. Eur. Union*, 2024.
- [27] Government of India, "The Digital Personal Data Protection Act, 2023 (Act No. 22 of 2023)," Ministry of Law and Justice, 2023.
- [28] International Organization for Standardization and International Electrotechnical Commission, "ISO/IEC 42001:2023 Information technology—Artificial intelligence—Management system," ISO, 2023.
- [29] World Economic Forum, "The Future of Jobs Report 2025," 2025.

Publication archival note: Official scholarly contribution published under CC BY 4.0 International License. Permanent record on <https://www.ijrek.org> | Article ID: IJREK-2026-00170