What are agentic workflows?

Tutorial: Agentic workflows

What are agentic workflows?

Agentic workflows are AI-driven processes where autonomous AI agents make decisions, take actions and coordinate tasks with minimal human intervention. These workflows leverage core components of intelligent agents such as reasoning, planning and tool use to execute complex tasks efficiently. Traditional automation such as robotic process automation (RPA), follow predefined rules and design patterns. This approach can be sufficient for repetitive tasks that follow a standard structure. Agentic workflows are dynamic, offering more flexibility by adapting to real-time data and unexpected conditions. AI Agentic workflows approach complex problems in a multistep, iterative way, enabling AI agents to break down business processes, adapt dynamically and refine their actions over time.

By enabling generative AI to handle intricate workflows, organizations benefit from improved operational efficiency, scalability and informed decision-making. As we continue to see advancements in machine learning and natural language processing (NLP), AI technology is becoming more common in industries seeking to automate and optimize processes while reducing reliance on human oversight. The impacts of evolving AI models not only affect software development but also industries such as healthcare, finance, human resources and much more.

Your subscription will be delivered in English. You will find an unsubscribe link in every newsletter. You can manage your subscriptions or unsubscribe here. Refer to our IBM Privacy Statement for more

information.

How do agentic workflows work?

Imagine a company has an IT support chatbot that follows a rule-based automation system. When an employee reports an issue (for example, "My wifi isn't working"), the chatbot runs through static decision trees and provides predefined responses. If the problem isn't resolved, the chatbot simply escalates to human support. This approach is efficient for basic, well-defined issues but struggles with complex, multistep troubleshooting that requires adaptability.

With an agentic workflow, the IT assistant approaches troubleshooting as a multistep, iterative process. If an employee reports a wifi issue, the agent follows a dynamic step-by-step process for breaking down the workflow:

1. **Understanding the problem**: The AI agent gathers detailed information from the employee, asking clarifying questions like, "Are other devices connected to the network?" or "Did this start after a recent update?"

2. **Executing diagnostic steps**: Based on the user's responses, the AI selects and runs different problem-solving steps. It might ping the router, check network logs or suggest specific settings changes, retrieving and summarizing this information for the user.

3. **Adaptive tool use**: If the AI detects a server-side issue, it can call an internal monitoring tool API to check for outages. If the issue is device-specific, it can retrieve driver update suggestions or run a script to reset network settings.

4. **Iterating based on results**: If an action doesn't resolve the problem, the AI adjusts its approach dynamically. It might cross-check related issues, reattempt diagnostics or suggest a different solution instead of just escalating immediately.

5. **Finalizing and learning**: If the issue is fixed, the AI logs the solution for future cases, improving its troubleshooting efficiency over time. If unresolved, it escalates with a detailed report, saving IT staff time by summarizing attempted fixes.

Components of agentic workflows

The core components of agentic workflows consist of:

- **AI agents** - In artificial intelligence (AI), a workflow is not agentic if it does not consist of an AI agent. An AI agent refers to a system or program that is capable of autonomously performing tasks on behalf of a user or another system by designing its workflow and utilizing available tools.

- **Large language models (LLMs)** - At the core of AI agents are large language models. LLMs are crucial for processing and generating natural language. The adjustment of LLM parameters like temperature will also result in varying output quality.

- **Tools** - For an LLM to acquire information beyond the data used in training the model, we must provide tools. Examples of commonly used tools include external datasets, web searches and application programming interfaces (APIs). We can use tools to tailor an AI agent to specific use cases beyond routine tasks.

- **Feedback mechanisms** - Feedback mechanisms such as a human-in-the-loop (HITL) or even other agents can be valuable in facilitating the AI agent's decision-making process and steering the agent output.

- **Prompt engineering** - Agentic workflow performance is heavily dependent on the quality of provided prompts. Prompt engineering helps generative AI models better comprehend and respond to a wide range of queries, from the simple to the highly technical. Common prompt engineering techniques include chain of thought (CoT), one-shot, zero-shot and self-reflection.

- **Multiagent collaboration** - Communication and distributed problem-solving within multiagent systems (MASs) are key for complex use cases. Each agent within a MAS can be designated a set of tools, algorithms and a domain of "expertise" so that agents are not all relearning the same information. Instead, agents are sharing their learned information with the rest of the MAS.

- **Integrations** - To streamline existing processes, agentic workflows need to be integrated with the existing infrastructure. This synergy is dependent on the requirements and goals of the agentic workflow. Data integration, the process of consolidating data into a central database for the agent to access, is often the first step. Other forms of integrations include agent frameworks such as LangChain, LangGraph, crewAI and IBM's BeeAI. These agent orchestration frameworks can serve as providers for achieving greater scale and performance. The integration of context-specific tools is also key to achieving relevant outputs.

The impact of agentic workflows

A personal anecdote from Andrew Ng, a leader in AI, highlights the adaptability of agentic workflows. Andrew recalls his demonstration of building AI agents, in which one of the many AI tools, a web search API, failed. The AI system was able to quickly handle the dependency failure by using an available Wikipedia search tool instead. The system completed the task and remained adaptable to the changing environment. The lessening need for human oversight might allow for our effort to be spent less on mundane, repetitive tasks and more on intricate work requiring human intelligence.

Andrew also explains that agentic workflows are meaningful not only for task execution but also for training the next generation of LLMs. In traditional, nonagentic workflows, using the output of one LLM to train another has not been found to lead to effective results. However, using an agentic workflow that produces high-quality data leads to useful training.