

# REPORT

## Understanding Agentic AI

---

### Introduction

Before we dive into Agentic AI, let's take a moment to understand what Generative AI is all about.

#### **Gen AI ( Generative AI ) :**

Gen AI is an AI technique used to create new artifacts that is creating new content such as text, music, audio, and videos based on existing artifacts.

- Focus on generating creative outputs or mimicking human-like expressions.
- It is a subfield of AI under ML, leveraging models like deep learning (e.g., generative adversarial networks (GANs), transformers).
- It is widely used in creative tasks (e.g., generating images, writing text).
- **Examples:** OpenAI's GPT models, DALL·E, and GANs, Codex (code generation).
- **Applications:** Content creation, text-to-image generation, summarization, data augmentation, etc.
- **Characteristics:** It is versatile, creative, scalable, adaptive, and enhances human collaboration but poses ethical challenges.
- Generative AI creates new and realistic content by learning patterns from data.

Now let us discuss about

#### **Agentic AI -**

---

---

The term "agentic" comes from the concept of agency, which is the ability of an entity to act independently and make decisions that affect its environment. Agentic AI refers to a type of AI that exhibits autonomous, goal-oriented behavior, and decision-making capabilities similar to those of a human agent.

- Agentic AI systems are designed to be proactive, self-directed, and able to adapt to changing situations.
- Agentic AI might not always generate new content like generative AI.
- It spans subfields like reinforcement learning, robotics, and systems engineering.
- **Examples:** Self-driving cars, or robotic systems, alphaGo.
- **Applications:** Autonomous vehicles, robotics, AI assistants for dynamic workflows, etc.
- **Focus:** Problem-solving, decision-making, and goal-oriented actions.
- **Characteristics:**
  1. Autonomy: Agentic AI systems can operate independently, making decisions without direct human intervention.
  2. Goal-oriented behavior: Agentic AI systems have clear goals and objectives, and they work to achieve them through their actions.
  3. Self-awareness: Agentic AI systems have some level of self-awareness, which enables them to understand their own strengths, weaknesses, and limitations.
  4. Adaptability: Agentic AI systems can adapt to changing situations and environments, and they can learn from experience.
  5. Reasoning and decision-making: Agentic AI systems can reason about their goals and environment, and they can make decisions that are in line with their objectives.

---

## Relationship between Generative AI and Agentic AI

Generative AI and Agentic AI are not mutually exclusive. For instance, an **agentic AI system** could use **generative AI** as a component.

➤ For example:

1. A self-driving car (agentic AI) could generate scenarios to simulate traffic situations (generative AI).
2. Both are different approaches or types of AI tailored to specific use cases.

## Agentic AI and Traditional AI

### Key Difference -

Traditional AI is rule-based, task-specific and follows strict programming or predefined instructions.

In contrast Agentic AI exhibits autonomy, adaptability, and goal-oriented behavior in dynamic environments.

## Agents of Agentic AI

In Agentic AI, agents are classified based on their capabilities and the level of autonomy they exhibit. Here are the main types :

### 1. Simple Reflex Agents -

- Respond to specific inputs with pre-programmed actions.

- 
- Example: A thermostat adjusting temperature based on readings.

## 2. Model-based Reflex Agents -

- Maintain an internal model of the world to handle more complex situations and remember past states.

- Example: Navigation systems that adjust routes based on traffic conditions.

## 3. Goal-based Agents -

- Take actions to achieve specific goals or objectives, using problem-solving strategies.

- Example: A robot navigating to a target location/destination.

## 4. Utility-based Agents -

- Maximize a utility function to make the best decisions, considering trade-offs between different goals.

- Example: Financial trading algorithms optimizing for profit.

## 5. Learning Agents -

- Improve their performance based on experience and feedback from the environment.

- Example: AI in video games that adapts strategies based on player behavior.

## 6. Autonomous Agents -

- Fully autonomous and capable of independent decision-making and goal management.

- 
- Example: Self-driving cars making decisions on the road.

These agents vary in complexity, from simple rule-based systems to advanced ones capable of learning and adapting to dynamic environments.

## **UserProxy Agent, and Assistant Agent**

User Agent, Proxy Agent, and Assistant Agent are not typically considered fundamental types of agents in the framework of Agentic AI. Generally they describe **functional roles or applications** of agents rather than their core decision-making architectures.

### **1. UserProxy Agent**

---

A UserProxy Agent acts as an intermediary between a user and a system or service. Its primary role is to represent the user, manage their requests, and handle communication with other agents or systems on behalf of the user.

## **2. Assistant Agent**

An Assistant Agent is designed to help users directly by providing services, support, or information. It provides end-user functionality and may also interact with other agents to gather or process information. Unlike a UserProxy Agent, which primarily interfaces between the user and other systems, an Assistant Agent.

## **Advantages of Agentic AI**

- Increased Efficiency and Automation.
- 24/7 Availability.
- Scalability.
- Accuracy and Precision.
- Personalization.

## **Disadvantages of Agentic AI**

- 
- Lack of Human Judgment and Empathy.
  - Ethical Concerns.
  - Job Displacement.
  - Accountability Issues.
  - Security Risks.
  - Over-Reliance on AI.

## **Problem Statement - Implementing a Simple Web Task Assistant with Agentic AI**

---

### **Objective:**

The goal is to create an intelligent system that can handle simple web tasks, such as checking the current weather in a city, using an open-source agentic framework.

---

### **Framework and Tools:**

#### **1. Framework:**

- **AutoGen Framework:** An open-source tool to design agents that can perform specific tasks.
-

---

## 2. Tools:

- **Web Search Tool:**

- Use **BeautifulSoup** to extract data from web pages.
- Alternatively, rely on a search engine API (e.g., DuckDuckGo Search API) using a library like **duckduckgo-search** for faster and more accurate results.

---

## Agent Design:

### 1. UserProxyAgent:

- **Role:** Acts as the main point of interaction with the user.
- **Key Functions:**
  - Takes user queries, like "What's the weather in Tokyo?"
  - Passes these requests to the AssistantAgent.

### 2. AssistantAgent:

- **Role:** Handles the actual web search and retrieves the required data.
- **Key Functions:**
  - Uses the Web Search Tool to search online.
  - Extracts and processes data (e.g., weather information) before sending it to the UserProxyAgent.

---

## How It Works:

### 1. User Input:



- 
- The user interacts with the **UserProxyAgent**, asking a question
  - E.g : "What's the temperature in Paris?".
2. **Task Assignment:**
    - The **UserProxyAgent** forwards the query to the **AssistantAgent**.
  3. **Data Collection:**
    - The **AssistantAgent** uses tools like **BeautifulSoup** or **duckduckgo-search** to gather relevant information.
  4. **Processing Results:**
    - The **AssistantAgent** organizes the retrieved information into a clear response.
  5. **Response Delivery:**
    - The **UserProxyAgent** delivers the final response to the user.

## Implementation Plan:

1. **Set Up Environment:**
  - Install libraries like **BeautifulSoup** and **duckduckgo-search**.
  - Configure the AutoGen framework for creating agents.
2. **Develop Agents:**
  - **UserProxyAgent:** Create a system to manage user inputs and outputs.
  - **AssistantAgent:** Build the logic for web searches and data processing.
3. **Integrate Agents:**
  - Ensure seamless communication between the **UserProxyAgent** and **AssistantAgent**.
4. **Test the System:**
  - Test the workflow with sample queries

- 
- E.g : "Find the weather in New York".
  - Check for response accuracy and speed.

#### **5. Deploy:**

- Package the system for end-users with a simple, intuitive interface.
- 

### **Future Improvements:**

#### **1. Better Language Understanding:**

- Integrate advanced NLP models to make the system more conversational.

#### **2. Support for More APIs:**

- Add services like OpenWeatherMap or Google Search API to improve accuracy and functionality.

#### **3. Error Handling:**

- Build safeguards to manage API failures, invalid inputs, and unexpected scenarios.

#### **4. Broader Capabilities:**

- Expand to handle other web tasks, like fetching news, stock prices, or travel information.
- 

### **Conclusion:**

The Agentic AI - Simple Web Task Assistant lays a strong foundation for building intelligent systems that can automate web-based tasks. Using open-source tools ensures cost-effectiveness and flexibility, while planned enhancements can make the system more robust and versatile for a wide range of applications.

---

---

---

## References

1. Research Papers.
2. Websites.
3. Articles and Blogs.
4. General Reference.
5. GitHub.