2

Principles of Agentic Systems

In the previous chapter, we introduced the basics of generative AI, learned about types of generative AI models, and briefly discussed LLM-powered AI agents. In this chapter, we will discuss the basic principles of agentic systems, starting with a brief discussion about the concept of agency and autonomy, followed by a discussion about intelligent agents and their characteristics. We will also discuss various agentic system architectures as well as multi-agent systems through the lens of the travel booking assistant example that we discussed in the previous chapter.

The main topics discussed in this chapter are as follows:

- Understanding self-governance, agency, and autonomy
- Reviewing intelligent agents and their characteristics
- Exploring the architecture of agentic systems
- Understanding multi-agent systems

By the end of this chapter, you will have an overview of the basics of intelligent agents and the most critical aspects of agentic system architecture that must be considered while building an intelligent agentic system.

Technical requirements

You can find the code files for this chapter at `https://github.com/PacktPublishing/Building-Agentic-AI-Systems` and follow the README file in the repository to set up your development environment.

Understanding self-governance, agency, and autonomy

The captivating aspect of **agentic systems** lies in the intricate decision-making processes they employ, which provide valuable insights into how choices are optimized within specific contexts. These systems often challenge our conventional understanding of accountability and responsibility.

Agentic systems act as the driving force behind innovation and technological advancements in various fields, including robotics, AI, and systems engineering. The development and deployment of these systems have catalyzed the exploration and creation of new forms of automation and intelligent behaviors. Let's discuss some of the areas where agentic systems are making headway:

- **Robotics**: In the field of robotics, agentic systems have paved the way for the design and implementation of autonomous robots capable of navigating complex environments, performing intricate tasks, and adapting to changing conditions. These robots, equipped with decision-making capabilities and agency, have found applications in areas such as manufacturing, exploration, search and rescue operations, and healthcare. For instance, robots with agentic behavior can autonomously navigate through disaster zones, assess potential risks, and make decisions to assist in rescue efforts, demonstrating intelligent and adaptive behavior.
- **AI**: In AI, agentic systems have been instrumental in the development of intelligent agents and decision support systems. These systems leverage advanced algorithms, machine learning techniques, and knowledge representation methods to analyze data, reason about complex scenarios, and provide intelligent recommendations or automated decision-making capabilities. Agentic AI systems have been applied in domains such as finance, healthcare, transportation, and marketing, enabling more efficient and effective decision-making processes.
- **Systems engineering**: In the field of systems engineering, agentic systems have facilitated the design and implementation of complex, distributed, and adaptive systems. These systems often consist of multiple interacting components or subsystems, each exhibiting agentic behavior and decision-making capabilities. Such systems are found in areas such as power grids, transportation networks, and cyber-physical systems, where intelligent and autonomous decision-making is crucial for efficient operation, resource allocation, and fault tolerance.

Central to the idea of agentic systems are the concepts of self-governance, agency, and autonomy. Let's discuss each of these concepts in the subsequent sections to understand what they are and explore the critical role they play within an agentic system architecture.

Self-governance

Agentic systems are artificial and human systems that possess self-governance, adaptability, and interaction. Self-governance refers to the ability of a system or entity to govern or control itself autonomously, without external direction or control. In the context of agentic systems, self-governance implies that the system can make its own decisions, set its own goals, and regulate its behavior based on its internal rules, models, and decision-making algorithms. Basically, they operate according to their rules and internal states, and execute a change of behavior, if necessary, due to a change of the environment or the objectives. Such systems interact with the environment or other systems in a way that is meaningful to them and, through these interactions, become influenced.

Some key aspects of self-governance in agentic systems are as follows:

- **Self-organization**: The ability to organize and structure its own internal processes, resources, and behavior without external intervention
- **Self-regulation**: The capability to monitor and adjust its own actions and outputs based on feedback from the environment or internal states, to ensure it operates within desired parameters or constraints
- **Self-adaptation**: The ability to modify its behavior, strategies, or decision-making processes in response to changes in the environment or its own internal conditions, to achieve its goals more effectively
- **Self-optimization**: The ability to continuously improve its performance, efficiency, or decision-making capabilities through learning, experience, or evolutionary processes
- **Self-determination**: The ability to set its own objectives, priorities, and courses of action based on its internal decision-making processes, without being entirely controlled by external forces

Self-governance in agentic systems is often enabled by the integration of various technologies, frameworks, and methodologies such as machine learning, knowledge representation, reasoning, and decision-making algorithms. These allow the system to process information, learn from data and experiences, and make autonomous decisions based on its acquired knowledge and the current context.

Agency

Agency is described as the capability of an individual, or any other entity, to act independently and make choices. In the context of human and artificial systems, agency encompasses the following key elements:

- **Decisional authority**: This refers to the power or ability to act and perform actions according to a chosen alternative or course of action. Systems with agency possess the autonomy to evaluate different options and select the most appropriate action based on their internal decision-making processes, rather than being solely driven by external forces or predetermined rules.

- **Intentionality**: Agency implies the existence of intentions, goals, or objectives that guide the actions and behavior of the system. Agentic systems are not merely reactive; they have a sense of purpose and can pursue specific objectives, adjusting their actions and strategies as necessary to achieve those goals.
- **Responsibility**: Agency is closely tied to the concept of responsibility, which is the answerability or accountability for the outcomes and consequences of one's actions. Systems with agency are considered responsible for their decisions and the impact of their actions on the environment or other entities they interact with.

In most cases, AI agency involves the system's ability to make decisions autonomously based on its internal programs, models, and the data it processes. These decisions can have a significant impact on the functioning of the system itself or its interactions with the environment.

Let's go back to our travel booking assistant example from the previous chapter, which is responsible for booking flight tickets and perhaps even making hotel reservations. In this case, the system would exhibit agency by analyzing various factors, such as the availability of flights between two cities, the price of the tickets, and any other restrictions given by the user such as preferred seat class and so on, and then making decisions on how to optimally look for flights and hotels that meet those criteria so that it minimizes the overall cost of travel for the customer. The system would be responsible for the outcome of its decisions, which would impact the overall travel plan and the cost of travel for the customer.

Autonomy

Autonomy is closely related to the concept of agency but focuses more specifically on the degree of independence an entity or system possesses. It can be broken down into several aspects:

- **Operational autonomy**: This refers to the ability of a system to perform a specific task or set of tasks without direct human intervention or control. A system with operational autonomy can execute its functions independently, relying on its own internal processes, decision-making algorithms, and environmental sensing capabilities.
- **Functional autonomy**: This aspect of autonomy involves the system's ability to make choices and take action to achieve set targets or objectives, modulated by the environment or context in which it operates. Functionally autonomous systems can adapt their behavior and decision-making processes in response to changing conditions or stimuli, enabling them to pursue their goals more effectively.

- **Hierarchical autonomy**: This aspect relates to the amount of decisional authority or decision-making power awarded to a system within a larger framework or organizational structure. Systems with higher hierarchical autonomy have greater latitude in making decisions that impact their subsystems or broader operations, while systems with lower hierarchical autonomy may have more constraints or oversight from higher-level entities.

In AI and robotics, autonomy is a key concept that refers to the extent to which a system can perform tasks and make decisions without the need for continuous human intervention.

In our travel booking assistant example, the system would have operational autonomy to carry out tasks such as booking flights or hotels, managing reminders, and retrieving travel information with little or no human input. It would also possess functional autonomy, allowing it to interpret user commands, adapt to individual preferences, and make decisions that align with the user's goals and context. The level of hierarchical autonomy granted to such a system might depend on factors such as user privacy preferences or the system's access to sensitive data and resources.

Note that autonomy in AI and robotic systems does not necessarily imply a complete absence of human oversight or control. Often, these systems operate within well-defined boundaries and constraints set by their designers or operators, while still exhibiting a significant degree of autonomy. In our travel booking example, the chatbot requests additional information, such as travel dates, locations, name, and address, from the user to make flight bookings. It then cross-references this data with available flights, suggesting options that match the user's preferences. If any detail is missing or unclear, the chatbot prompts the user for clarification, ensuring accuracy while still operating autonomously within the boundaries set by its programming.

The concepts of self-governance, agency, and autonomy in AI systems are often accompanied by ethical considerations, particularly regarding the level of autonomy granted to these systems and the potential risks and implications of their decisions. As AI systems become more advanced and capable of independent decision-making, ensuring their alignment with human values and ethical principles becomes crucial.

Example of agency and autonomy in agents

Let’s illustrate the concept of agency and autonomy with a simple algorithm for the travel booking assistant. Note that this algorithm doesn’t necessarily use AI just yet, but it would help understand the concepts. Our travel booking assistant algorithm may look as follows.

```
Algorithm 1: Travel booking assistant algorithm with agency and autonomy
Require: Agent name N
Ensure: Initialized TravelAgent object A with agency and autonomy

Initialize A ← CreateTravelAgent(N)
Initialize A.goals ← empty list
Initialize A.knowledge_base ← empty dictionary

// Agency: Ability to act on behalf of a user
function SetGoal(G)
    A.goals.Append(G)  // Agency: Defining objectives
function UpdateKnowledge(K, V)
    A.knowledge_base[K] ← V  // Agency: Acquiring information from an API, and scoring

// Autonomy: Ability to operate independently
function MakeDecision(Options)
    best_option ← max(Options, key =score)  // Autonomy: Independent decision-making
    return best_option
function BookTravel(Departure, Destination)
    Output “Agent A.name is booking travel to Destination”

// Agency: Execute action on behalf of user
SetGoal(“Book flight from Departure to Destination”)
UpdateKnowledge({Departure, Destination})
// Autonomy: Book travel independently by finding best flight
MakeDecision()
// Implement booking logic here and store into A
Output A
```

Here is how the algorithm works, demonstrating the abilities of agency and autonomy:

1. We start by naming our agent; we call it `TripPlanner`.
2. Next, we initialize a new `TravelAgent` object with the name `N = "TripPlanner"`; this represents the creation of an entity capable of both agency and autonomy.
3. We then set up a list to store the goals for the agent. This relates to agency, as goals represent the intentions or desired outcomes the agent will work towards on behalf of the user. This is indicated by `A.goals ← empty list`.
4. Next, we initialize an empty dictionary (also known as a map or key-value pairs) to store the agent's knowledge. This is crucial for both agency (acting on behalf of users) and autonomy (independent operation), as it will contain information the agent uses to make decisions.
5. Steps *4* and *5* in the algorithm indicate the definition of a function that adds a new goal `G` to the agent's list of goals. This is akin to the agent taking on objectives on behalf of the user. This is indicated by `A.goals.Append(G)`. Think of this as the piece of code that will receive the user's chat message, such as *"Book me a flight from San Diego to Seattle."* Here, the goal is to book a flight from San Diego to Seattle.
6. Steps *6* and *7* in the algorithm indicate the definition of a function that updates the agent's knowledge base with a new key-value pair (map or dictionary). This represents agency through the acquisition of information that will be used to act on behalf of the user. It also supports autonomy by providing the agent with information it can use to make independent decisions. This operation is represented as `A.knowledge_base[K] ← V`. In our case, this function uses several travel-related APIs (in theory) to get flight options between two cities, thus forming the knowledge into a knowledge base. This is also a place where each of these flight options will be scored; for example, late flights get low scores, and early flights get higher scores.
7. Steps *8* through *10* define a function that does a few different things. It takes a list of options and selects the best one based on some scoring criteria. This is an example of autonomy in the algorithm since the agent independently evaluates options and makes a decision without direct human intervention.
8. Finally, steps *10* through *15* demonstrate how all of these components work together, starting from setting the goal of flight booking using the departure and destination cities, updating the knowledge base using a flight lookup API, and then scoring the available flights. Subsequently, it uses the `MakeDecision` function to find the best possible flight as per the highest score and performs the flight booking for the user.

A Python implementation of the `BookTravel` function from the algorithm is shown in the following code snippet:

```python
def book_travel(departure: str, destination: str):
    self.set_goal(f"Book flight from {departure} to {destination}")
    self.update_knowledge(departure, destination)

```

```
   try:
      best_flight = self.make_decision()
      booking_confirmation = f"BOOKING_#12345"
      self.knowledge_base['booking_confirmation'] = \
                       booking_confirmation
      print(f"Booking confirmed: {booking_confirmation}")
   except Exception as e:
      print(f"Booking failed: {str(e)}")

if __name__ == "__main__":
   agent = TravelAgent("TripPlanner")
   agent.book_travel("SAN", "SEA")
   print("\n----------- Final Agent State: -----------")
   print(f"Name: {agent.name}")
   print(f"Goals: {agent.goals}")
   if 'booking_confirmation' in agent.knowledge_base:
     print(f"Booking Confirmation: \
     {agent.knowledge_base['booking_confirmation']}")
```

The output of this code when the agent is initialized to book a flight from SAN (San Diego) to SEA (Seattle) looks as follows:

```
Agent TripPlanner is booking travel from SAN to SEA
Goal set: Book flight from SAN to SEA
Knowledge updated with 3 flight options
Decision made: Selected flight JetBlue
Booking confirmed: BOOK-JetBlue-TRIPPLANNER
----------- Final Agent State: -----------
Name: TripPlanner
Goals: ['Book flight from SAN to SEA']
Booking Confirmation: BOOK-JetBlue-TRIPPLANNER
```

For the full implementation of the trip planner agent, refer to the `Chapter_02.ipynb` Python notebook in the GitHub repository.

In this code snippet, the `book_travel` function takes a departure city code (such as SAN or SEA, which are airport codes) and subsequently calls other functions to set the goal, update its knowledge base, and then make a decision on which flight to choose and book that flight. Note that our agent, although has some functionality of agency and autonomy, is not intelligent. It cannot take plain text messages from a user and decipher what the user intends to do to set its goals, update its knowledge base, and then perform the actions; rather, it needs the airport codes. However, as we saw in our example, a user (or customer) may simply express their intentions in plain language such as "*Book me a flight from San Diego to Seattle*".

In its current form, given any such user input (message), the agent is incapable of determining what the departure and destination cities are, what the user is asking for, or even what the string of input text even means. This is where generative AI steps in, as we will see in the subsequent chapters. For now, let's continue with our discussion by looking at the characteristics of agents.

Reviewing intelligent agents and their characteristics

An intelligent agent is a complex, self-governed entity that perceives its environment and takes action to achieve certain goals or objectives. These agents can range from basic systems that strictly adhere to a predefined set of rules to highly advanced systems with the ability to learn and adapt from experience. Intelligent agents are characterized by several key attributes:

- **Reactivity**: Reactive agents respond to changes and events occurring in their environment in real time. They continuously monitor their surroundings and adjust their behavior accordingly. This reactivity allows agents to adapt to dynamic conditions and respond appropriately to stimuli, ensuring their actions remain relevant and effective.
- **Proactiveness**: An ideal intelligent agent should not merely react to events but also exhibit proactive behavior. Proactive agents anticipate future needs, challenges, or opportunities, and take the initiative to plan and act accordingly. They are goal-oriented and actively pursue strategies to achieve their objectives, rather than simply reacting to circumstances as they arise.
- **Social ability**: Many intelligent agents operate in multi-agent systems, where they interact and cooperate with other agents or humans to achieve common goals that require collaborative effort. Social ability encompasses communication, coordination, and negotiation skills, enabling agents to work together effectively and leverage collective intelligence or resources.

With these key characteristics, intelligent agents demonstrate remarkable versatility and efficiency across a wide spectrum of domains and scenarios. Their capabilities enable them to excel in tasks ranging from simple, automated processes to highly complex, dynamic decision-making situations that demand real-time adaptation and environmental responsiveness. In addition to these core characteristics, intelligent agents may possess other advanced capabilities:

- **Learning and adaptation**: Intelligent agents have the ability to learn from experience and adapt their behavior over time. They can acquire new knowledge, refine their decision-making processes, and improve their performance through techniques such as machine learning, reinforcement learning, or evolutionary algorithms.
- **Reasoning and planning**: Intelligent agents may employ reasoning and planning capabilities to analyze complex situations, formulate strategies, and make informed decisions. They can leverage techniques such as knowledge representation, logical inference, and planning algorithms to navigate through intricate problem spaces and determine optimal courses of action.