Agentic AI: Architectures, Types, Capabilities, Mathematical Equations and Governance in the Era of Autonomous Intelligence

Dr. Nitin S. Shrirao
nshrirao@gmail.com
Director

Mr. Dnyaneshwar S. Jadhav
dnyaneshwarj473@gmail.com
Assistant Professor

Mrs. Sarita B Patil
sarita.sica91@gmail.com
Assistant Professor

Siddhant Institute of Computer Application

Abstract: Agentic Artificial Intelligence (Agentic AI) represents a major advancement in the evolution of intelligent systems by enabling autonomous planning, decision-making, and action execution. Unlike traditional AI models, which are primarily reactive and designed to respond to predefined inputs, Agentic AI systems possess capabilities such as memory, reasoning, goal-oriented planning, tool integration, and dynamic adaptation to changing environments. These characteristics allow them to perform complex, multi-step tasks with minimal human intervention, making them suitable for applications across healthcare, finance, cybersecurity, robotics, education, and enterprise automation. This paper explores the conceptual foundations, architectural principles, and practical applications of Agentic AI while examining the key differences between conventional AI and autonomous agent-based systems. It presents a taxonomy that distinguishes reactive and agentic models, discusses single-agent and multi-agent orchestration frameworks, and analyzes the role of memory, planning, perception, and external tool usage in intelligent decision-making. The study also highlights the ethical, security, governance, and accountability challenges associated with deploying autonomous AI, including issues related to transparency, privacy, bias, reliability, and human oversight. As AI systems become increasingly capable of operating independently within complex environments, establishing robust governance and regulatory frameworks is essential. This paper proposes a comprehensive framework for the design, evaluation, and responsible deployment of Agentic AI, emphasizing safety, explainability, human-in-the-loop supervision, and ethical compliance. By integrating technical innovation with effective governance strategies, the proposed approach aims to maximize the benefits of Agentic AI while minimizing potential risks. The findings contribute to the growing body of research on trustworthy autonomous systems and provide practical guidance for developing secure, reliable, and human-centered Agentic AI solutions.

Keyword: *Artificial Intelligence(AI), large language models (LLMs), Reinforcement learning (RL), Markov Decision Process (MDP), Multi-Agent System (MAS), Hierarchical Multi-Agent System(HMAS), Robot Operating System (ROS), Non-Player Characters (NPCs), Human-in-the-Loop (HITL)*

I. Introduction

Artificial Intelligence (AI) has evolved from rule-based expert systems to data-driven machine learning and deep learning models. Most traditional AI systems function in a reactive manner, responding to predefined inputs without long-term autonomy or goal persistence. Recent advances in large language models, reinforcement learning, and tool-integrated AI have enabled the emergence of **agentic AI systems**—intelligent agents capable of autonomous goal formulation, planning, execution, and adaptation [1].

As AI systems gain increased autonomy, they begin to resemble decision-making entities operating within dynamic environments. This evolution introduces new opportunities for automation and optimization, but also raises concerns regarding safety, accountability, ethics, and governance. This

paper explores agentic AI as a distinct paradigm, examining its foundations, architectures, applications, and regulatory challenges [2].

Recent advances in large language models (LLMs) have enabled the rise of Agentic AI—intelligent systems that not only respond to queries but autonomously pursue goals, coordinate sub-tasks, and interact with environments. Tools like Auto-GPT, Baby AGI, and Lang Chain exemplify this evolution, showing how LLMs can act as cognitive engines within agentic frameworks [3].

II. Architectures of Agentic AI

Agentic AI architecture emphasizes modularity, adaptability, and decision-making. Common designs include:

1. **Reinforcement Learning (RL) Frameworks:** Reinforcement learning (RL) is a broad framework in which an agent learns how to behave within an environment in order to maximize cumulative rewards. It consists of two primary elements: the environment, which defines the task or problem, and the agent, which embodies the learning algorithm that makes decisions and improves through interaction.

   The agent and the environment continuously interact with each other. At each step, the agent acts on the environment according to its policy $\pi(a_t \mid s_t)$, where s_t is the current observation of the environment, and it receives a reward $r_t + 1$ and a further observation $s_t + 1$ from the environment. The goal is to improve the policy so as to maximize the sum of rewards (return).

   Agents learn via trial-and-error, maximizing rewards. Architectures like Deep RL (e.g., DQN or PPO algorithms) enable goal-directed behavior. For instance, OpenAI's Gym environments train agents for tasks like robotic manipulation [4].

   Basic Mathematical Representation:

   a) Environment Model: Markov Decision Process (MDP):

      Reinforcement learning problems are mathematically modeled using a **Markov Decision Process (MDP)**

      $$M = (S, A, P, R, \gamma)$$

      Where:

      - S - Set of states (environment situations)
      - A - Set of actions available to the agent
      - $P(s' \mid s, a)$ – Transition probability from state s to s' after action a
      - $R(s, a)$ – Reward received after taking action a in state s
      - $\Upsilon \in [0,1]$ – Discount Factor (Importance of future rewards)

   b) Policy (Agent Behavior)

      A **policy** defines how the agent chooses actions:

      $$\pi(a \mid s) = \Pr(A_t = a \mid S_t = s)$$

- Deterministic policy: $a = \pi(s)$
- Stochastic policy: probability distribution over actions [5].

c) Return (Total Reward)

The goal of RL is to maximize the **expected cumulative reward** (called return):

$$G_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots$$

Or in summation form:

$$G_t = \sum_{k=0}^{\infty} \gamma^k \, r_{t+k+1}$$

2. **Multi-Agent Systems (MAS):** A Multi-Agent System (MAS) is a computational framework in which several independent AI agents operate within a common environment, working together or competing to address complex, large-scale challenges that would be difficult for a single system to handle alone. These agents rely on decentralized control, distinct capabilities, and structured communication to enhance decision-making, adaptability, efficiency, and scalability in dynamic settings [6].

3. A Multi-Agent System (MAS) is made up of several interacting agents that operate and make decisions within a common environment. The mathematical model extends single-agent MDPs to multi-agent settings [7].

a) MAS as a Multi- Agent Markov Decision Process(MMDP)

A MAS can be defined as:

$$M = (S, \{A_i\}^N_{i=1}, P, \{R_i\}^N_{i=1}, \gamma)$$

Where:

- S - Global state space of the environment
- N - Number of agents
- A_i - Action space of agent i
- $P(s'|s,a_1,\ldots a_N)$ - State transition function
- R_i $(s, a_1,\ldots,a_N)$ – Reward function for agent i
- γ – Discount factor

Each agent selects actions simultaneously, influencing the shared environment [8].

b) Joint Action Space

The **joint action** of all agents is:

$$\mathrm{a} = (a_1, a_2,\ldots,a_N) \in A_1 \times A_2 \times \cdots \times A_N$$

The environment evolves according to:

$$S_{t+1} \sim P(s_{t+1} | \, s_t, a_t)$$

c) **Agent Policies**

Each agent has its own policy:

$$\pi_i(a_i \mid s) = \Pr(A_i = a_i \mid S = s)$$

The **joint policy** is:

$$\pi(a|s) = \prod^{N}{}_{i=1} \pi_i(a_i \mid s)$$

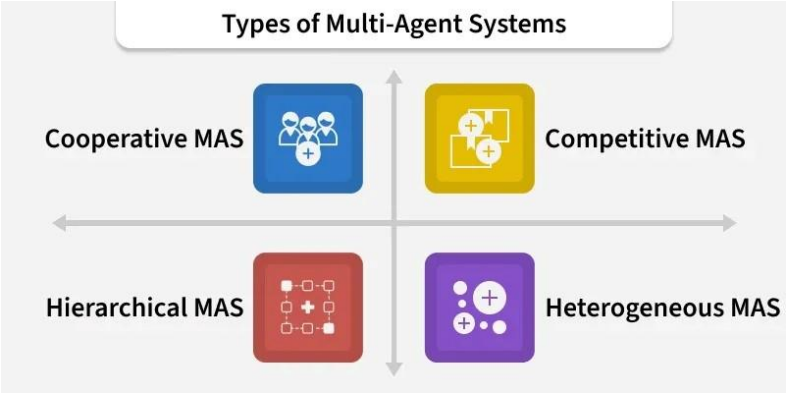

Fig. 2.2. Types of Multi-Agent Systems

i. Cooperative MAS

Cooperative Multi-Agent Systems (MAS) are systems in which several autonomous agents—whether software-based or physical—work together and interact to accomplish common objectives or tackle complex tasks, often achieving results beyond what a single agent can do alone. These systems play a key role in contemporary artificial intelligence by focusing on coordination, communication, and frequently collaborative decision-making to enhance overall effectiveness [9].

- Agents within these systems collaborate to accomplish a shared objective.
- They exchange information and resources to perform tasks that would be difficult for a single agent to handle alone.
- Several drones working together during a search-and-rescue operation [10].

ii. Competitive MAS

Competitive Multi-Agent Systems (MAS) refer to a type of Artificial Intelligence framework where multiple autonomous agents operate within a shared environment, often with conflicting goals. Unlike cooperative MAS, where agents work together, competitive MAS

involve agents that act in their own self-interest, often aiming to maximize their own utility at the expense of others, frequently resulting in a zero-sum game.

- Agents have conflicting goals and compete for limited resources.
- Example: In competitive gaming, players (agents) compete to win [11].

iii. Hierarchical MAS

A Hierarchical Multi-Agent System (Hierarchical MAS or HMAS) is an organizational paradigm in artificial intelligence where intelligent agents are structured across multiple, layered levels. Unlike "flat" MAS, where all agents operate at the same level, hierarchical systems utilize a tiered structure where higher-level agents (managers/supervisors) oversee, coordinate, or instruct lower-level agents (workers/executors) [12].

- These systems are arranged in a clear hierarchy, with agents operating at various levels.
- Agents at the top levels oversee and direct those at lower levels.
- For example, mission control systems used in space exploration.

iv. Heterogeneous MAS

A heterogeneous Multi-Agent System (MAS) is a system composed of agents with different dynamics, parameters, or models, as opposed to a homogeneous system where all agents are identical. This type of system is characterized by its ability to incorporate specialized, diverse components to achieve a common goal, often resulting in improved performance and efficiency in complex tasks [13].

- In these systems, agents possess varied capabilities or responsibilities, allowing the system to be more versatile and responsive to change.
- Example: Hybrid robot teams combining aerial drones with ground-based robots.

Systems made up of interconnected agents that interact with one another, frequently applying game-theoretic models or consensus-based methods. Examples include platforms such as JADE or the Foundation for Intelligent Physical Agents (FIPA) standards, where agents engage in negotiation within simulations like traffic control or supply chain management [14].

4. **Hierarchical Architectures:** Hierarchical architecture is a system design where components are organized into levels or layers, creating a top-down structure with increasing detail, from high-level plans to low-level commands, used in software (like OS or network protocols), computer systems (memory, databases), and physical buildings to manage complexity, improve scalability, and define clear responsibilities, often using principles like layers (access, distribution, core) or main-subroutine models [15].

   Inspired by human cognition, these layer planning (e.g., HTN planning) with execution. Systems like ROS (Robot Operating System) integrate perception, reasoning, and actuation for autonomous robots.

   Hierarchical Architectures decompose a complex decision problem into **multiple levels of abstraction**, where higher levels set goals and lower levels execute actions.

This improves scalability, interpretability, and long-horizon planning [16].

a) Hierarchical Model Structure

A hierarchical system is represented as a tuple:

$$H = (S, A, G, \Pi, P, R, \gamma)$$

Where:

- S – State space
- A – Primitive action space
- G – Set of subgoals (high – level decisions)
- Π – $\{\pi^H, \pi^L\}$ – Policies at different levels
- P – State transition model
- R – Reward function
- γ – Discount Factor

b) Two-Level Hierarchy (Manager–Worker Model)

i. High-Level Policy (Manager)

Selects subgoals instead of actions:

$$g_t \sim \pi^H (g | s_t)$$

Where $g_t \in G$ is a **temporal abstraction** lasting multiple steps.

ii. Low-Level Policy (Worker)

Executes primitive actions to achieve the subgoal:

$$a_t \sim \pi^L(a \mid s_t, g_t)$$

5. Transformer-Based Models with Agency:

Transformer-based models with agency, or Agent Transformers, extend Large Language Models (LLMs) from passive text generators into active agents capable of planning, decision-making, and tool interaction to accomplish multi-step tasks. They leverage self-attention mechanisms to process complex environmental data, with applications ranging from reinforcement learning policies to autonomous software assistants [17].

Advanced LLMs like GPT-4, augmented with tools (e.g., via Lang Chain), enable agentic behavior through prompt engineering and external APIs, allowing dynamic task decomposition.

These architectures often incorporate safety layers, such as value alignment (e.g., via inverse reinforcement learning), to ensure actions align with human values.

Transformer-based agents extend standard Transformer models by embedding them into a **decision-making loop** (perception → reasoning → action → feedback). Mathematically, this combines **sequence modeling + reinforcement learning + planning**.

a) Environment Formulation (Agent Setting)

An agent engages with an environment that is represented as a Partially Observable Markov Decision Process (POMDP):

$$\mathcal{E} = (S, A, O, P, R, \gamma)$$

Where:

- *S:* Hidden environment states
- *O:* Observations (text, images, tool outputs)
- *A:* Actions (tokens, API calls, decisions)
- $P(s' \mid s, a)$: State transition
- $R(s, a)$:Reward signal
- γ: Discount factor

b) Sequence-Based World Representation

The Transformer encodes the interaction history as a sequence:

$$x_{1:t} = (o_1, a_1, o_2, a_2, \ldots, o_t)$$

This sequence acts as the agent's **implicit belief state:**

$$b_t \approx f\theta(x_{1:t})$$

where $f\theta$ is the Transformer parameterized by θ.

III. Types of Agentic AI Agents

AI agents are categorized according to their degree of intelligence, the way they make decisions, and how they engage with their surroundings to accomplish specific goals. Some function strictly according to preset rules, whereas others rely on learning algorithms to adapt and enhance their performance over time [18].

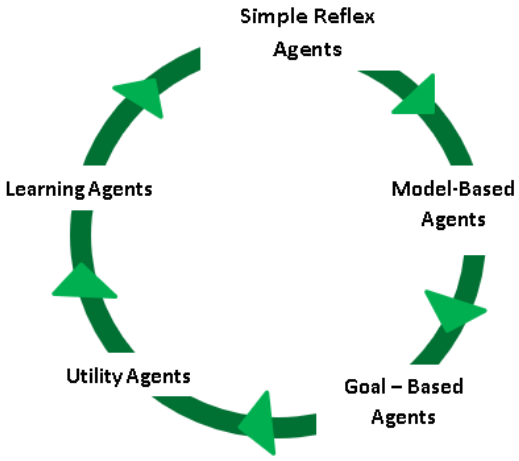

Fig. 3. Types of Agentic AI Agents

AI agents are generally categorized into five primary types: simple reflex agents, model-based reflex agents, goal-oriented agents, utility-driven agents, and learning agents. Each category features unique traits and serves different purposes, spanning from straightforward automated processes to advanced, adaptive AI systems [19].

1. Simple Reflex Agents

A **simple reflex agent** is one of the most basic forms of AI agents, designed to act solely on the **current percept** without considering past states or predicting future outcomes. It operates on **condition–action rules** — essentially an *if-this-then-that* logic — making it fast, predictable, and easy to implement.

These agents do not retain memory and depend solely on sensors to interpret their surroundings and actuators to execute actions. They work best in completely observable, stable environments where the rules remain relatively constant.

Core Components

- Sensors: Gather data from the surroundings (such as temperature sensors or motion detectors).
- Condition–Action Rules: Pre-established links that specify which action to take when certain conditions are met.
- Actuators: Carry out the selected response (for example, activating motors or flipping switches).

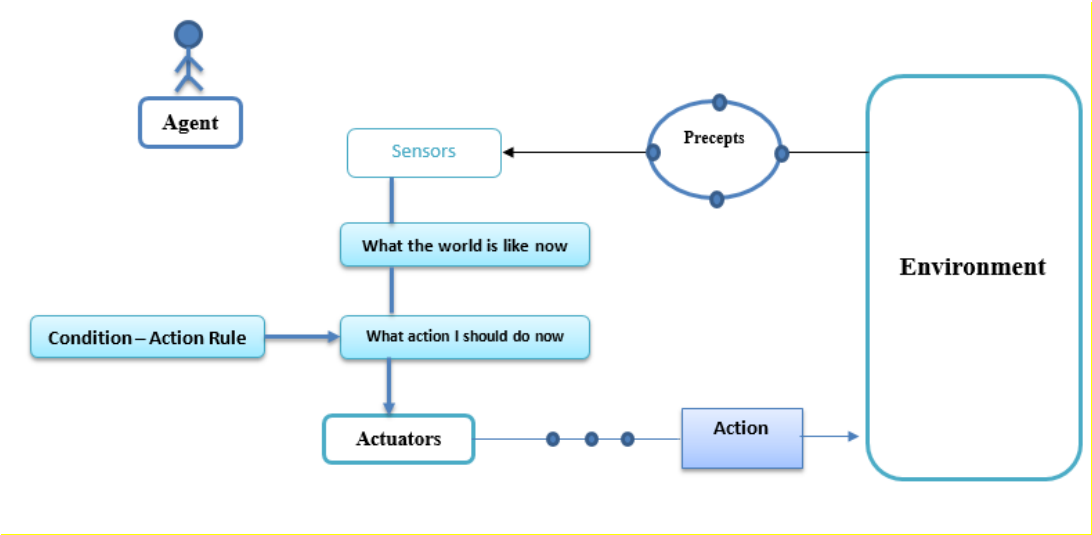

Fig. 3.1. simple reflex agent

Basic Mathematical Representation:

Let:

- Pt = percept at time t
- At= action at time t
- f = condition-action function

Then the agent function is:

At = f (Pt)

This means:

Action = Condition – Action Rule (Current Percept)

2. Model-Based Agents

Model-based agents are intelligent systems in artificial intelligence that keep an internal model or representation of their environment. In contrast to simple reflex agents, which act only on immediate sensory input, model-based agents rely on this internal model to understand how the environment changes and to make better-informed decisions. As a result, they perform especially well in environments that are dynamic or only partially observable [20].

Key Components

- **Sensors**: Collect data about the current state of the environment, whether through physical devices (like cameras) or digital sources (such as APIs).
- **Internal Model**: Captures the agent's representation of the environment, including its rules, behavior patterns, and the possible results of different actions.
- **Reasoning Component**: Combines sensory information with the internal model to assess situations, anticipate consequences, and determine appropriate actions.
- **Actuators**: Carry out the selected actions, allowing the agent to interact with and influence its surroundings.

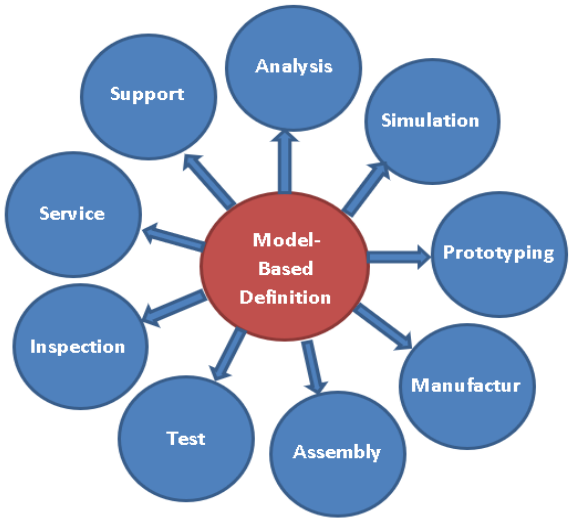

Fig. 3.2. Model-based agents

Basic Mathematical Representation:

Let:

- P_t = percept at time t
- A_t= action at time t
- S_t = internal state at time t
- f = action function
- μ = state update function

Then the agent function is:

$$S_t = \mu (S_{t-1}, P_t)$$

This means:

New State = Update Function (Previous State, Current Percept)

3. Goal-based AI Agents

Goal-based agents in AI are built to accomplish defined objectives by planning, carrying out tasks, and adapting their behavior as needed. They are commonly applied in areas that require high levels of flexibility and adaptability [21].

Robotics: Goal-based agents are extensively used in robotics for navigation, task accomplishment, and human-robot interaction. For example, an automated warehouse robot uses such agents to optimize routes, avoid obstacles, and efficiently perform tasks like picking and placing items [22].

Game AI: In video games, goal-based agents control non-player characters (NPCs) that exhibit intelligent behavior. These agents enable NPCs to strategize, adapt to player actions, and create a realistic and challenging gameplay experience.

Autonomous Vehicles: Autonomous vehicles depend on goal-driven agents to travel along roads, steer clear of hazards, and follow traffic regulations. These agents ensure safe and efficient operation by dynamically adjusting to changing road conditions and traffic patterns.

Resource Management: In industries such as logistics and energy, goal-oriented agents are used to allocate resources properly. For instance, in supply chain management, these agents assist with organizing delivery routes, lowering expenses, and making sure shipments arrive on time.

Healthcare: In the healthcare sector, goal-driven agents play a key role in supporting diagnosis, developing treatment strategies, and monitoring patients. For example, AI-powered systems can examine patient information to suggest customized treatment options or continuously monitor vital signs to notify medical professionals during critical situations. These cases highlight the flexibility and effectiveness of goal-oriented agents in addressing complex challenges across different domains [23].

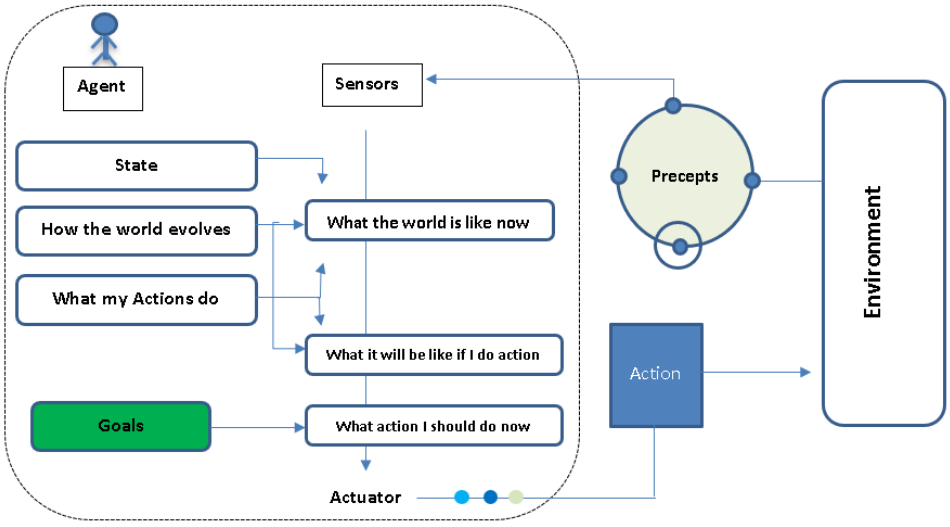

Fig. 3.3. Goal-based AI agents

Basic Mathematical Representation:

Let:

- S_t = internal state at time t
- A= set of possible actions
- G = goal state
- Result (S, a) = transition function
- Π = plan (Sequence of actions)

Then the agent function is:

$$S_{t+1} = \text{Result}(S_t, A_t)$$

This means:

Next State = Transition Function (Current State, Action)

4. Utility-Based Agents

Utility-based agents are AI systems that go beyond achieving a simple goal by selecting the best action to maximize a calculated "utility" score, representing happiness or performance. They evaluate multiple options, handling trade-offs, uncertainty, and, for example, balancing speed and safety. These agents use a "utility function" to map states to real numbers, ensuring optimal decision-making in complex environments [24].

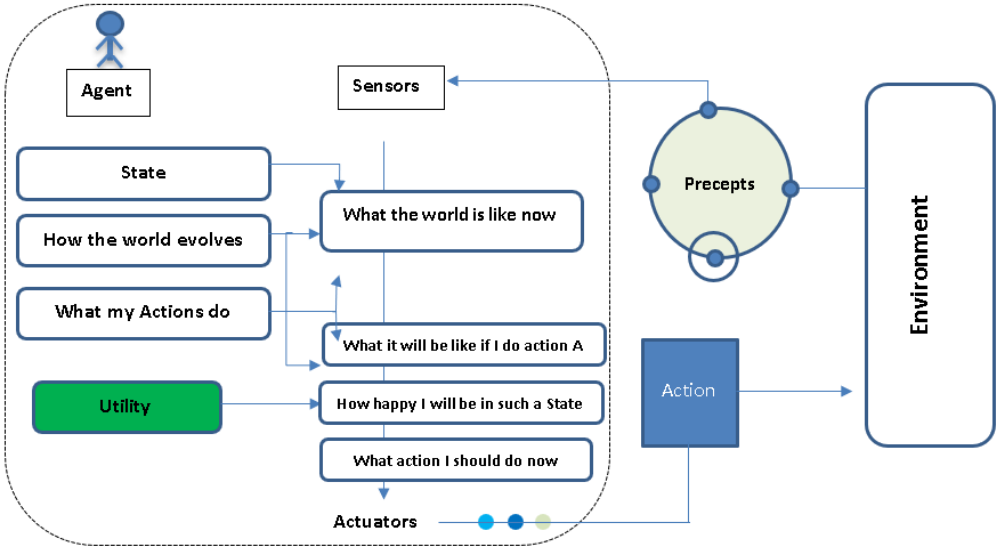

Fig. 3.4. Utility-based AI agents

Basic Mathematical Representations:

1. **Utility Function:**

Let:

- S = set of states
- U (S) = utility of states S

The utility function is defined as:

$$U: S \rightarrow R$$

This means each state is assigned a real number representing its desirability.

2. **State Transition Model :**

Let:

- P (S′| S, a)) = probability of reaching next state S′ from state S after action a

$$S_{t+1} = \text{Result}(S_t, A_t)$$

3. **Expected Utility (EU) :**

For a given action a, the expected utility is:

$$EU(a \mid S_t) = \sum_{ś} P(Ś \mid S_t, a)\, U(Ś)$$

Where:

- Ś = possible next states
- $P(Ś \mid S_t, a)$ = probability of each next state
- U (Ś) = utility of next state

5. Learning Agents

Learning agents are AI systems that autonomously interact with environments, acquiring knowledge from experiences to improve performance over time, rather than relying solely on pre-programmed rules. They adapt to new situations using machine learning (reinforcement/imitation learning), making them ideal for dynamic, uncertain tasks [25].

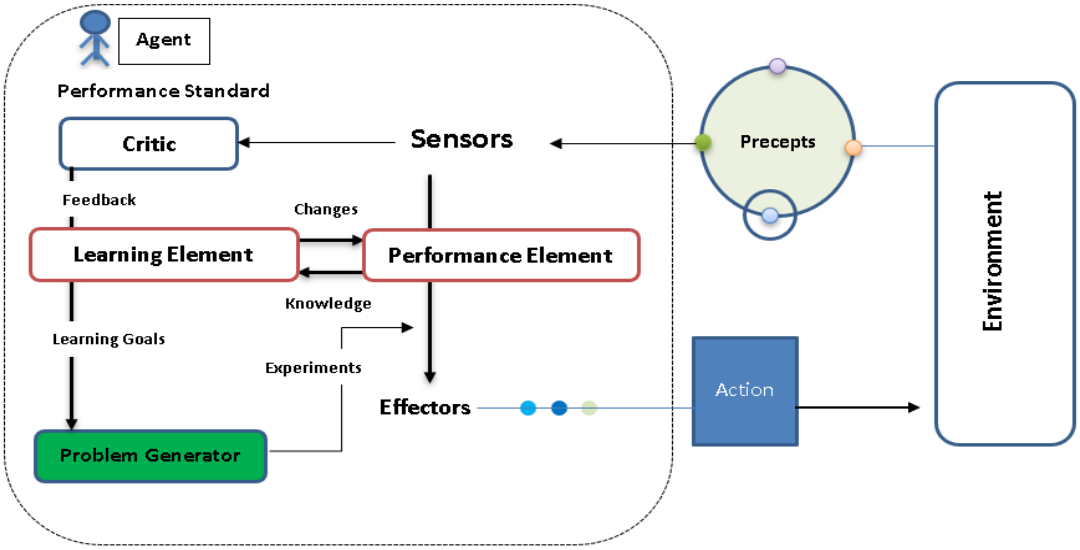

Fig. 3.5. Learning AI agents

Basic Mathematical Representations:
Let:

- S_t = state at time t
- A_t = action at time t
- R_t = reward at time t
- π = policy (mapping from state to action)
- θ = parameters of the agent

Policy Function

The agents select actions using a policy:

$$A_t = \pi (S_t; \theta)$$

Where:

θ are learnable parameters.

Reward Function

$$R_t = R (S_t, At)$$

Reward indicates how good the action is.

3. Capabilities of Agentic AI

3.1 Autonomous Task Completion

Unlike generative AI tools (e.g., chatbots), agentic AI can independently:

- Identify tasks from context or prompts.
- Create subgoals.
- Decide the order of operations.
- Execute steps with monitoring and correction.

Example: An agent receives an instruction to "organize a marketing campaign." It searches trends, drafts content, schedules posts, and analyzes engagement—without step-by-step input [26].

5.2 Use Cases

Sector	Application	Description
Enterprise	Workflow Automation	Agents plan meetings, manage inboxes, optimize resources [6]
Cybersecurity	Autonomous Incident Response	AI triages alerts, isolates threats, generates reports [7]
Healthcare	Patient Monitoring Agents	Track vitals, recommend actions, notify staff [8]
Finance	Investment Portfolio Agents	Monitor markets, rebalance assets, explain decisions [9]

3.3 Limitations and Open Challenges

- **World Model Incompleteness**: Agents often hallucinate or misrepresent context.
- **Goal Misalignment**: Misunderstanding human intent can lead to harmful or suboptimal actions.
- **Coordination Complexity**: In multi-agent systems, conflicting goals or dependencies introduce failures.

4. Governance in the Age of Autonomous Intelligence

4.1 Emerging Risks

Risk Type	Example
Security	Agents exploiting API keys or performing unauthorized actions [10]

Explainability	Inability to justify decisions due to opaque planning [11]
Ethical Violations	AI acting in ways that conflict with human rights or values
Regulatory Non-compliance	GDPR or AI Act violations due to unsupervised data access [12]

4.2 Governance Frameworks

- **SAGA**: Enforces registration, trust scoring, and scope-limiting. Critical for agent marketplaces and decentralized systems [3].
- **TRiSM**: Focuses on trust, risk, and security management across the agent lifecycle. Includes performance benchmarks and explainability metrics [4].
- **MI9**: Provides dynamic monitoring, anomaly detection, and runtime policy enforcement [5].

These frameworks propose:

- Agent "passports" for identity and behavior constraints.
- Sandboxed execution environments.
- Governance agents that monitor and intervene when anomalies arise.

4.3 Global and Regulatory Landscape

- **Council of Europe Framework Convention on AI (2024)**: Mandates transparency, risk assessment, and human oversight [13].
- **U.N. AI Governance Proposals**: Call for global standards, auditing mechanisms, and oversight bodies [14].
- **National Regulations**: Vary widely but often lag behind agentic system capabilities.

4.4 Enterprise Governance Strategies

- Identity & Role Management
- Policy-based Access Control
- Audit Trails & Logs
- Human-in-the-Loop (HITL) Safeguards
- Agent Transparency Dashboards

Enterprises adopting Agentic AI must align governance with business strategy, security, and compliance from the outset [27].

5. Conclusion

Agentic AI marks a significant shift in the evolution of artificial intelligence, moving beyond reactive systems toward autonomous, goal-driven agents capable of planning, reasoning, and executing complex tasks. By combining reinforcement learning, multi-agent systems, hierarchical planning, and transformer-based models, agentic AI enables adaptive decision-making and proactive task completion across dynamic environments. These capabilities offer transformative potential in sectors such as enterprise automation, cybersecurity, healthcare, and finance, where intelligent agents can improve efficiency, scalability, and responsiveness.

However, increased autonomy also introduces critical challenges. Risks related to goal misalignment, lack of transparency, coordination failures, security vulnerabilities, and regulatory non-compliance highlight the need for strong governance frameworks. Approaches such as SAGA, TRiSM, and MI9 emphasize identity management, monitoring, explainability, and policy enforcement to ensure responsible deployment. Emerging global regulations further stress transparency, accountability, and human oversight.

In the end, the effectiveness of agentic AI will rely not just on technological progress, but also on strong ethical alignment, sound governance frameworks, and ongoing human oversight. A balanced approach that integrates innovation with responsibility is essential to ensure that autonomous intelligent systems serve societal interests safely and effectively.

References

1. Russell S, Norvig P. *Artificial intelligence: a modern approach*. 4th ed. Hoboken (NJ): Pearson; 2021.
2. Wooldridge M. *An introduction to multiagent systems*. 2nd ed. Chichester (UK): John Wiley & Sons; 2009.
3. Sutton RS, Barto AG. *Reinforcement learning: an introduction*. 2nd ed. Cambridge (MA): MIT Press; 2018.
4. Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, et al. Human-level control through deep reinforcement learning. *Nature*. 2015;518:529-533.
5. Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O. Proximal policy optimization algorithms [Internet]. arXiv:1707.06347; 2017 [cited 2026 Jul 22]. Available from: https://arxiv.org/abs/1707.06347
6. OpenAI. GPT-4 technical report [Internet]. arXiv:2303.08774; 2023 [cited 2026 Jul 22]. Available from: https://arxiv.org/abs/2303.08774
7. Yao S, Zhao J, Yu D, Du N, Shafran I, Narasimhan K, et al. ReAct: synergizing reasoning and acting in language models [Internet]. arXiv:2210.03629; 2022 [cited 2026 Jul 22]. Available from: https://arxiv.org/abs/2210.03629

8. Schick T, Dwivedi-Yu J, Dessì R, Raileanu R, Lomeli M, Hambro E, et al. Toolformer: language models can teach themselves to use tools [Internet]. arXiv:2302.04761; 2023 [cited 2026 Jul 22]. Available from: https://arxiv.org/abs/2302.04761

9. Nakajima Y. *Auto-GPT: an autonomous GPT-4 experiment* [Internet]. GitHub; 2023 [cited 2026 Jul 22]. Available from: https://github.com/Significant-Gravitas/Auto-GPT

10. Richards T. *BabyAGI: task-driven autonomous agent framework* [Internet]. GitHub; 2023 [cited 2026 Jul 22]. Available from: https://github.com/yoheinakajima/babyagi

11. Chase H. *LangChain: building applications with large language models* [Internet]. 2023 [cited 2026 Jul 22]. Available from: https://www.langchain.com/

12. Stone P, Veloso M. Multiagent systems: a survey from a machine learning perspective. *Auton Robots*. 2000;8(3):345-383.

13. Busoniu L, Babuska R, De Schutter B. A comprehensive survey of multiagent reinforcement learning. *IEEE Trans Syst Man Cybern C Appl Rev*. 2008;38(2):156-172.

14. Jennings NR. An agent-based approach for building complex software systems. *Commun ACM*. 2001;44(4):35-41.

15. Amodei D, Olah C, Steinhardt J, Christiano P, Schulman J, Mané D. Concrete problems in AI safety [Internet]. arXiv:1606.06565; 2016 [cited 2026 Jul 22]. Available from: https://arxiv.org/abs/1606.06565

16. Christiano PF, Leike J, Brown T, Martic M, Legg S, Amodei D. Deep reinforcement learning from human preferences. In: *Advances in Neural Information Processing Systems (NeurIPS)*; 2017.

17. Hadfield-Menell D, Milli S, Abbeel P, Russell S, Dragan A. The off-switch game. In: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*; 2017.

18. Brundage M, Avin S, Clark J, Toner H, Eckersley P, Garfinkel B, et al. *The malicious use of artificial intelligence: forecasting, prevention, and mitigation*; 2018.

19. European Parliament. *Artificial Intelligence Act (EU AI Act)*. Brussels (Belgium): European Union; 2024.

20. Council of Europe. *Framework Convention on Artificial Intelligence and Human Rights, Democracy and the Rule of Law*. Strasbourg (France): Council of Europe; 2024.

21. United Nations. *Governing AI for humanity report*. New York (NY): United Nations; 2025.

22. Doshi-Velez F, Kim B. Towards a rigorous science of interpretable machine learning [Internet]. arXiv:1702.08608; 2017 [cited 2026 Jul 22]. Available from: https://arxiv.org/abs/1702.08608

23. Ribeiro MT, Singh S, Guestrin C. "Why should I trust you?": explaining the predictions of any classifier. In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*; 2016.

24. National Institute of Standards and Technology. *AI Risk Management Framework (AI RMF 1.0)*. Gaithersburg (MD): NIST; 2023.

25. IBM. *AI governance framework* [Internet]. IBM; 2024 [cited 2026 Jul 22]. Available from: https://www.ibm.com/

26. Gartner. *Trust, Risk and Security Management (TRiSM) framework* [Internet]. Gartner; 2024 [cited 2026 Jul 22]. Available from: https://www.gartner.com/

27. OpenAI. *Planning and tool use in LLM agents* [Internet]. OpenAI Technical Blog; 2024 [cited 2026 Jul 22]. Available from: https://openai.com/