Chapter 1

Intelligent Agents

Outline

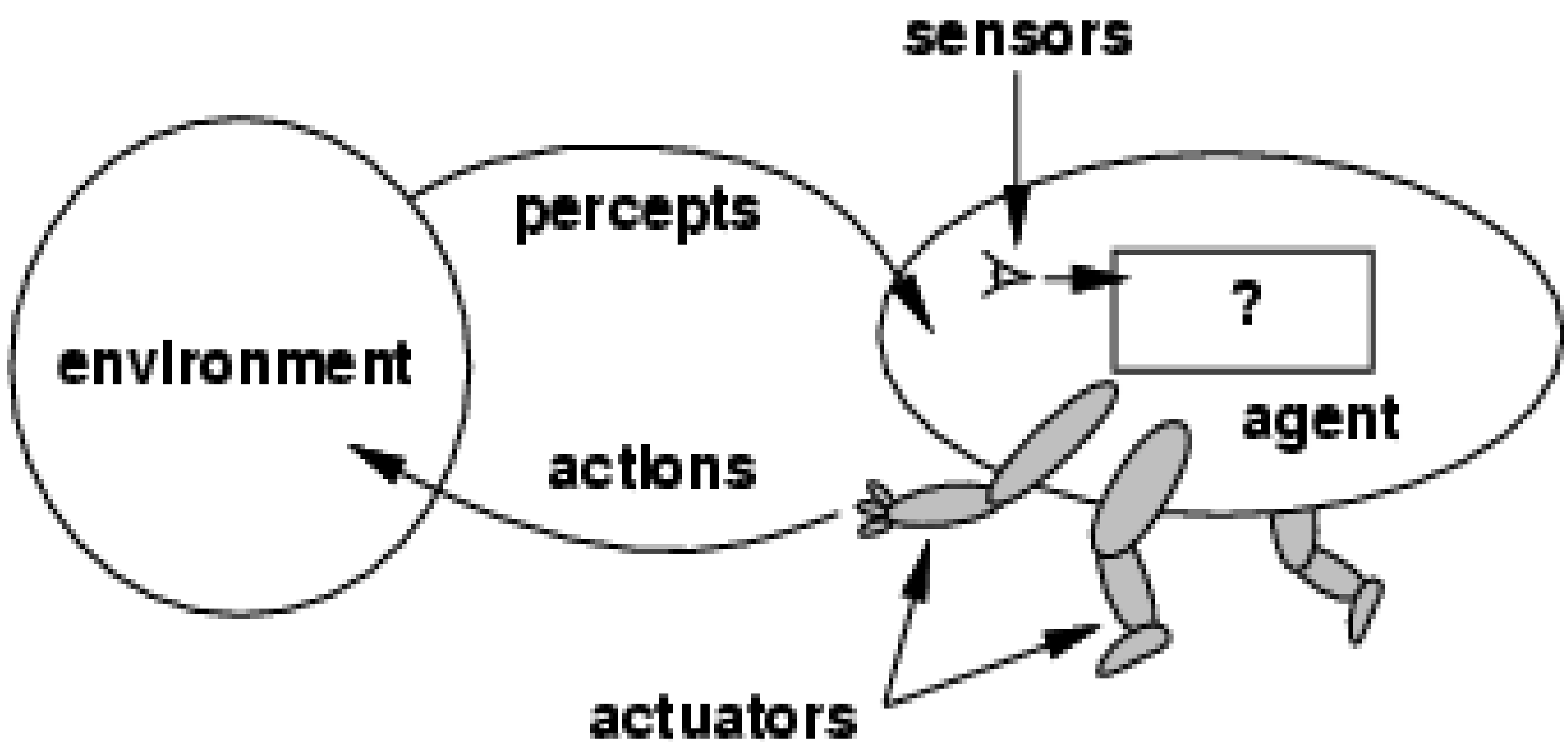
- Introduction to Intelligent Agents
- How agents should act
- Structure of Intelligent agents
- Environments
- Summary

Agent

An Agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors.

How Agent should Act?

- A Rational Agent is one that does the right thing.



Terms in AI

- **Performance Measure** : It is the criteria that determines how successful an agent is.
- **Omniscience** :An omniscient agent knows the actual outcome of its actions and can act accordingly.

Rational Agent

- Rational is depend on following things :
  - The performance measure that defines the degree of success.
  - Everything that agent perceived so far. Complete perceptual history called the percept sequence.
  - What agent knows about the environment
  - The actions that the agent can perform

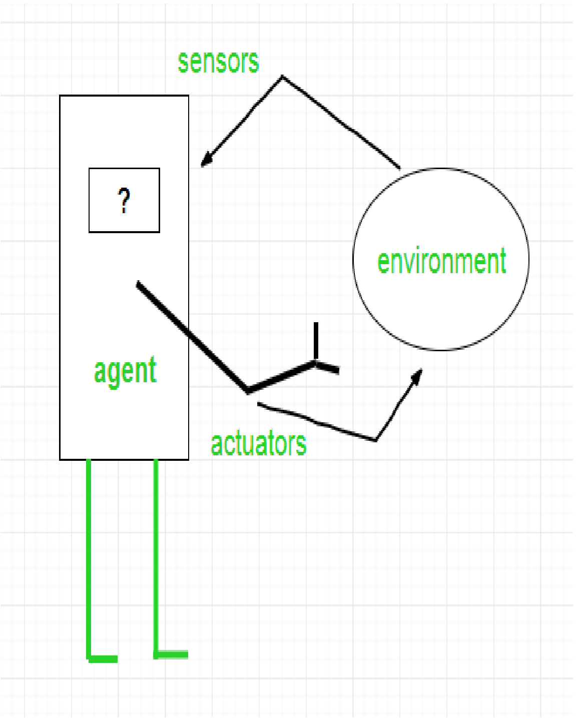
Ideal Rational Agent

- For each possible percept sequence , an ideal rational agent should do whatever action is expected to maximize its performance measure, on the basis of the evidence provided by the percept sequence and whatever built-in knowledge the agent has.

- Autonomy
  - If the agent ‘s actions are based completely on built-in knowledge , such that it need pay no attention to its percepts, the that agent lacks autonomy.

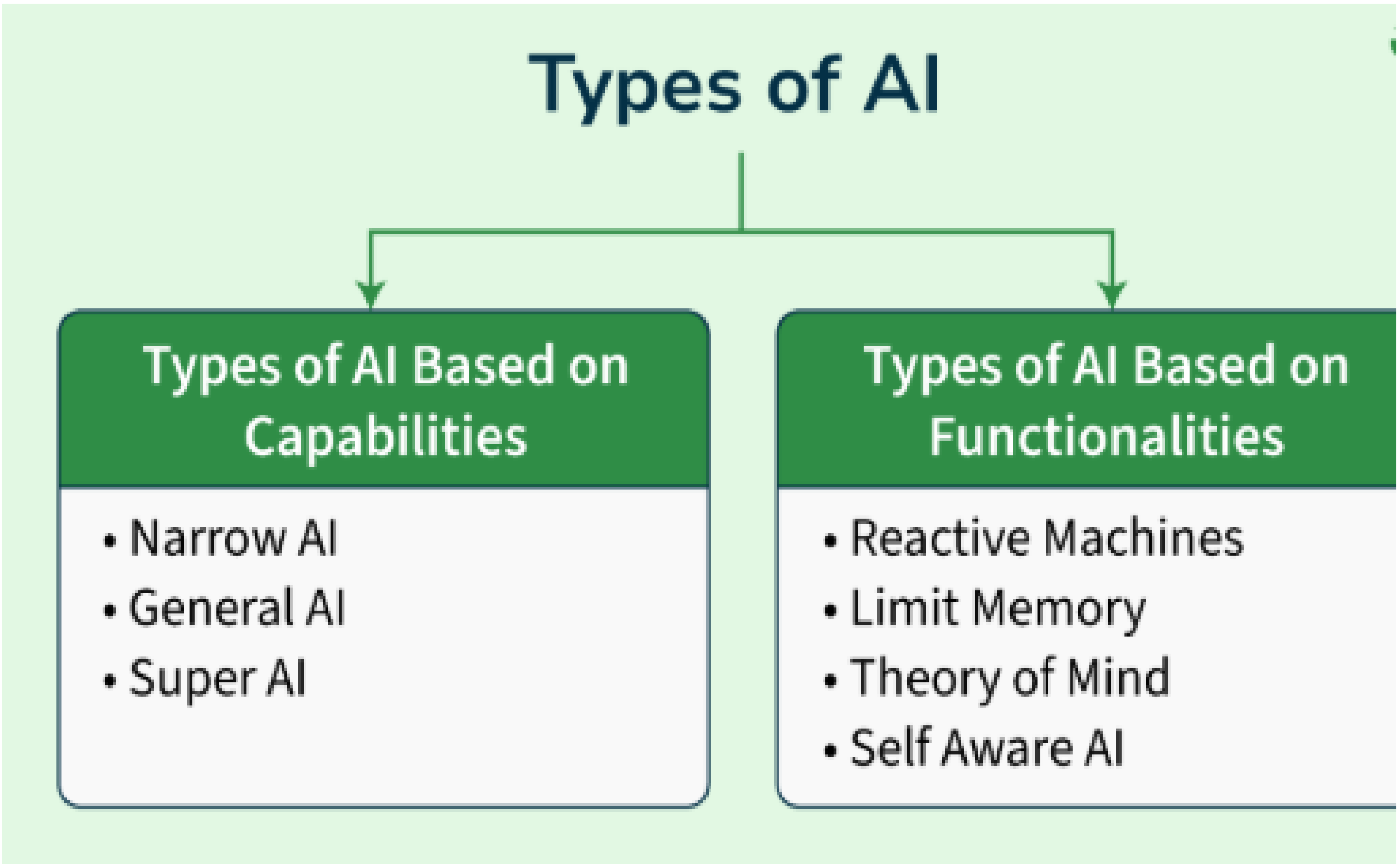
Structure of an Intelligent agents

- The job of AI is to design Agent Program: a function that implements the agent mapping from percepts to actions.



- Program will run on some device called as architecture.

- Agent = Architecture + Program


Types of AI
Types of AI Based on Capabilities
• Narrow AI
• General AI
• Super AI
Types of AI Based on Functionalities
• Reactive Machines
• Limit Memory
• Theory of Mind
• Self Aware AI

Narrow AI/Soft OR WEAK AI

- "Weak AI" refers to the use of software to study or accomplish specific problem solving or reasoning tasks that do not encompass the full range of human cognitive abilities.

- Example : a chess program such as "Deep Blue".

- "Weak AI" does not achieve
  - self-awareness, or
  - demonstrate a wide range of human-level cognitive abilities,
  - at best is merely an intelligent, a more specific problem-solver.

“HARD” OR “STRONG” AI

- “Strong AI” refers to a machine that approaches or supersedes human intelligence,
  - if it can do typically human tasks,
  - if it can apply a wide range of background knowledge and
  - if it has some degree of self-consciousness.

- “Strong AI” aims to build machines whose overall intellectual ability is indistinguishable from that of a human being.

Superintelligence (Super AI)

- Super AI **surpasses intelligence of human** in solving-problem, creativity, and overall abilities.
- Super AI develops emotions, desires, need and beliefs of their own.
- Able to make decisions of their own and solve problem of its own.
- Better than humans but also understand and interpret emotions and respond in a human-like manner.
- While Super AI remains speculative, it could revolutionize industries, scientific research, and problem-solving, possibly leading to unprecedented advancements.

Types of Artificial Intelligence Based on Functionalities

1. Reactive Machines

Reactive machines are the most basic form of AI. They operate purely based on the present data and do not store any previous experiences or learn from past actions. These systems respond to specific inputs with fixed outputs and are unable to adapt.

Examples:

- **IBM’s Deep Blue**, which defeated the world chess champion Garry Kasparov in 1997. It could identify the pieces on the board and make predictions but could not store any memories or learn from past games.
- **Google’s AlphaGo**, which played the board game Go using a similar approach of pattern recognition without learning from previous games.

2. Limited Memory in AI

Limited Memory AI can learn from past data to improve future responses. Most modern AI applications fall under this category. These systems use historical data to make decisions and predictions but do not have long-term memory. Machine learning models, particularly in autonomous systems and robotics, often rely on limited memory to perform better.

Examples:

- **Self-driving cars**: They observe the road, traffic signs, and movement of nearby cars, and make decisions based on past experiences and current conditions.
- **Chatbots** that can remember recent conversations to improve the flow and relevance of replies.

Theory of Mind

Theory of Mind AI aims to understand human emotions, beliefs, intentions, and desires. While this type of AI remains in development, it would allow machines to engage in more sophisticated interactions by perceiving emotions and adjusting behavior accordingly.

Potential Applications:

- **Human-robot interaction** where AI could detect emotions and adjust its responses to empathize with humans.
- **Collaborative robots** that work alongside humans in fields like healthcare, adapting their tasks based on the needs of the patients.

4. Self-Awareness AI

Self-Aware AI is an advanced stage of AI that possesses self-consciousness and awareness. This type of AI would have the ability to not only understand and react to emotions but also have its own consciousness, similar to human awareness.

While we are far from achieving self-aware AI, it remains the ultimate goal for AI development. It opens philosophical debates about consciousness, identity, and the rights of AI systems if they ever reach this level.

Potential Applications:

- Fully autonomous systems that can make moral and ethical decisions.
- AI systems that can independently pursue goals based on their understanding of the world around them.

PEAS

- PEAS: Performance measure, Environment, Actuators, Sensors
- Must first specify the setting for intelligent agent design

PEAS of Automated Taxi Driver

- Performance Measure: Safe, fast, legal, comfortable trip, maximize profits
- Environment: Roads, other traffic, pedestrians, customers
- Actuators: Steering wheel, accelerator, brake, signal, horn
- Sensors: Cameras, sonar, speedometer, GPS, odometer, engine sensors, keyboard

PEAS of Medical Diagnosis System

- Agent: Medical diagnosis system
- Performance measure: Healthy patient, minimize costs, lawsuits
- Environment: Patient, hospital, staff
- Actuators: Screen display (questions, tests, diagnoses, treatments, referrals)
- Sensors: Keyboard (entry of symptoms, findings, patient's answers)

PEAS of Part-Picking Robot

- Agent: Part-picking robot
- Performance measure: Percentage of parts in correct bins
- Environment: Conveyor belt with parts, bins
- Actuators: Jointed arm and hand
- Sensors: Camera, joint angle sensors

PEAS of Interactive English Tutor

- Agent: Interactive English tutor
- Performance measure: Maximize student's score on test
- Environment: Set of students
- Actuators: Screen display (exercises, suggestions, corrections)
- Sensors: Keyboard

Agent: Intelligent house

Percepts: signals from temperature sensor, movement sensor, clock, sound sensor

Actions: room heaters on/off, lights on/off

Goals: occupants warm, rooms light when occupied, house energy efficient

Environment: at various times, occupants enter and leave house, enter and leave rooms; daily variation in outside light and temperature

Agent: Car driver

Percepts: camera (array of pixels of various intensities), signals from GPS, speedometer, sonar

Actions: steer, accelerate, brake

Goals: safe, fast, legal trip

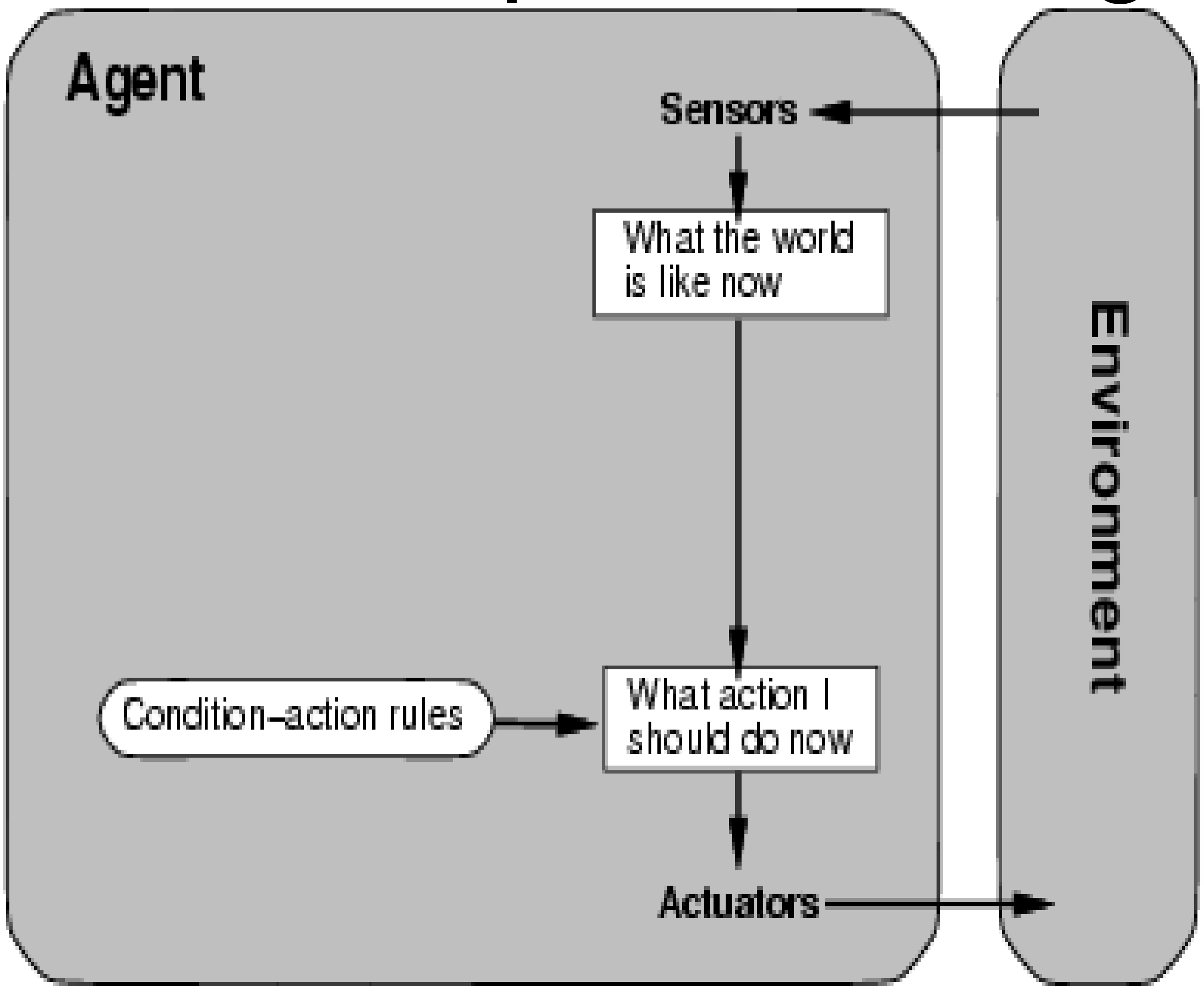
Environment: streets, intersections, traffic signals, traffic lights, other moving vehicles, pedestrians

Agent Type	Percepts	Actions	Goals	Environment
Medical diagnosis system	Symptoms, findings, patient's answers	Questions, tests, treatments	Healthy patient, minimize costs	Patient, hospital
Satellite image analysis system	Pixels of varying intensity, color	Print a categorization of scene	Correct categorization	Images from orbiting satellite
Part-picking robot	Pixels of varying intensity	Pick up parts and sort into bins	Place parts in correct bins	Conveyor belt with parts
Refinery controller	Temperature, pressure readings	Open, close valves; adjust temperature	Maximize purity, yield, safety	Refinery
Interactive English tutor	Typed words	Print exercises, suggestions, corrections	Maximize student's score on test	Set of students

Types of Agent programs

- Simple reflex Agents
- Agents that keep track of the world
- Goal-based Agents
- Utility based Agents

Simple Reflex Agents



How They Work

- **Condition-Action Rules**: These agents operate purely on a predefined set of rules of the form *if condition, then action.*
- **Stateless**: They do not maintain any memory or model of the world. Their actions are determined solely by the current percept.
- **Immediate Response**: The agent does not consider the history of its percepts or any future consequences of its actions.

Advantages

- Simple to design and implement.
- Effective in fully observable environments where all necessary information is available at any given time.

Disadvantages

- Limited adaptability; they fail in partially observable or dynamic environments.
- Cannot handle complex tasks requiring memory or reasoning.

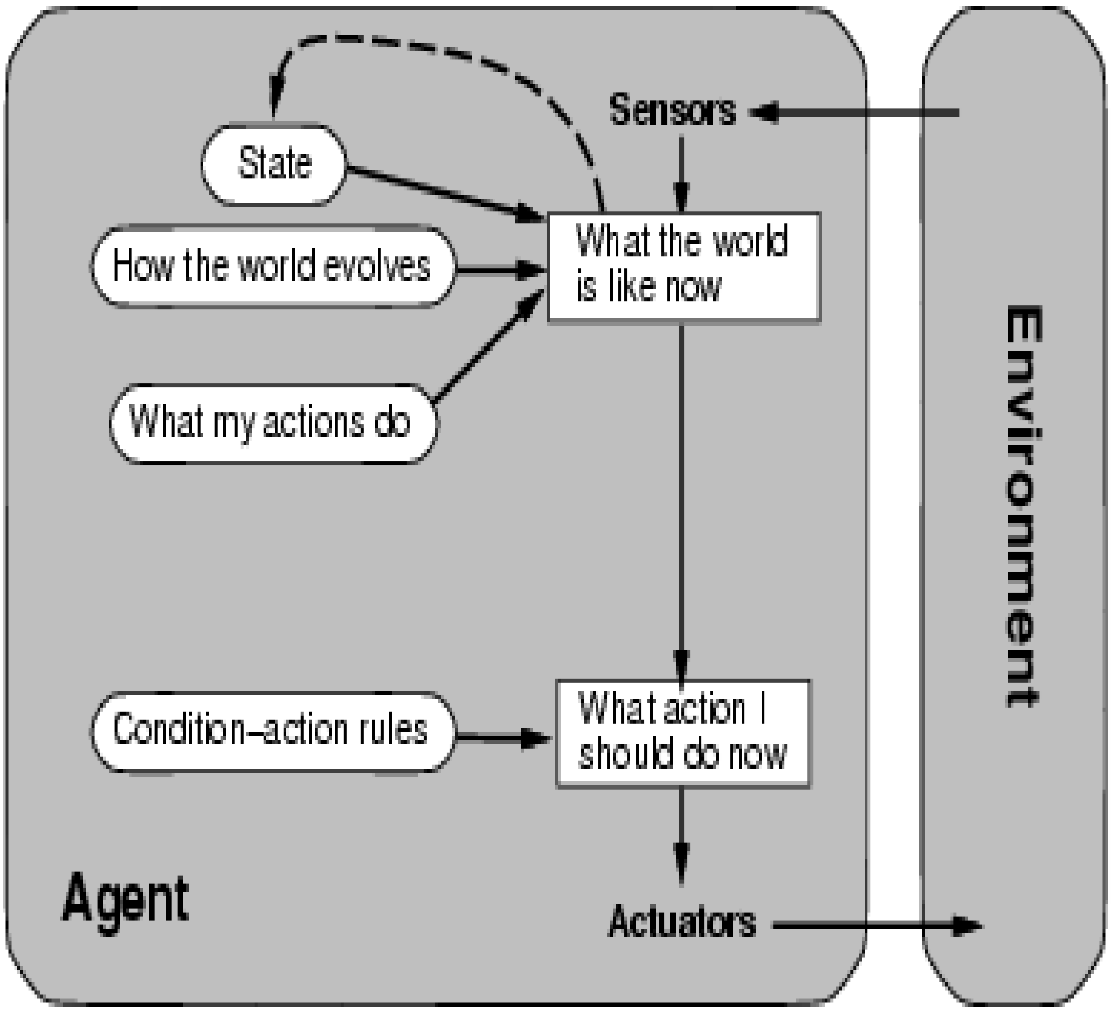
function SIMPLE-REFLEX-AGENT(*percept*)
returns *action*
    static: *rules*, a set of condition-action rules

    $state \leftarrow$ INTERPRET-INPUT(*percept*)
    $rule \leftarrow$ RULE-MATCH(*state*, *rules*)
    $action \leftarrow$ RULE-ACTION[*rule*]
    return *action*

Problems with Simple reflex agents are :

- Very limited intelligence.
- No knowledge of non-perceptual parts of the state.
- Usually too big to generate and store.
- If there occurs any change in the environment, then the collection of rules needs to be updated.

Agents that keep track of the world
Model-Based Agents



How They Work

- **Internal State**: These agents maintain an internal model of the world, which represents the current state of the environment.
- **World Evolution Rules**: They use rules that describe how the world changes over time (based on percepts and actions).
- **Partial Observability**: By maintaining an internal state, they can handle partially observable environments.

Advantages

- Can operate in more complex and dynamic environments by maintaining a model of the world.
- Better decision-making compared to simple reflex agents.

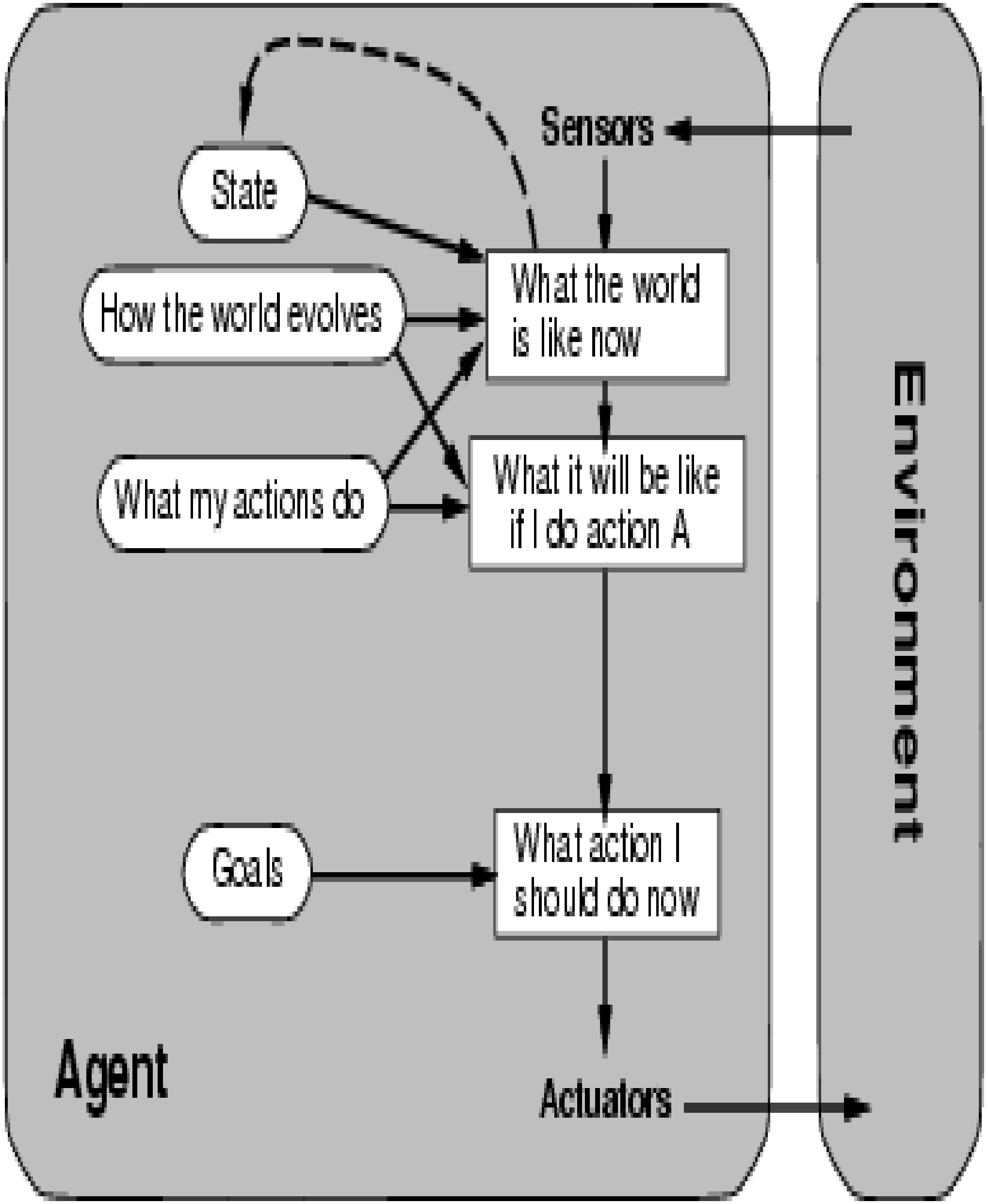
Disadvantages

- Increased computational and memory requirements.
- Accuracy depends on the quality of the model and its updates.

function REFLEX-AGT-WITH-MODEL(*prcpt*)
returns *action*
static: *model*, current world state (description)
rules, set of condition-action rules
K_{ev}, rules about how world evolves
K_{act}, description of *rules*

$model \leftarrow$ UPDATE-MODEL($model,\ prcpt,\ K_{ev},\ K_{act}$)
$rule \leftarrow$ RULE-MATCH($model,\ rules$)
$action \leftarrow$ RULE-ACTION[$rule$]
$model \leftarrow$ UPDATE-MODEL($model,\ action$)
return *action*

Goal Based Agents



How They Work

- **Goal-Driven Behavior**: These agents act to achieve specific goals, which define desirable states.
- **Search and Planning**: They evaluate possible actions and choose the ones that lead to achieving their goals.
- **Deliberation**: They may take longer to decide on actions as they analyze and plan based on the goal.

Advantages

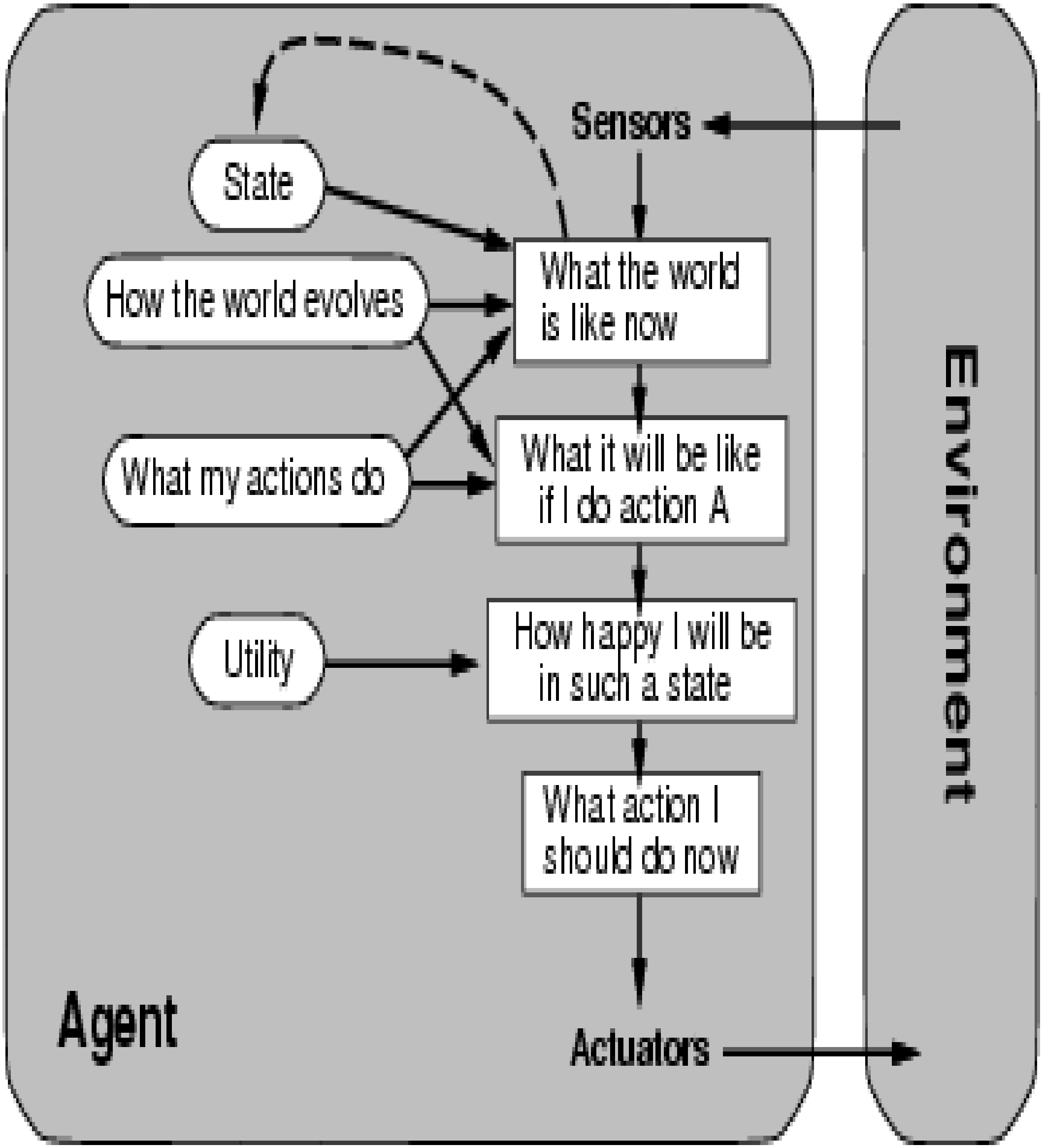
- Flexible and can adapt to different scenarios by changing goals.
- Can handle complex tasks that require reasoning and planning.

Disadvantages

- Computationally expensive; planning and goal evaluation may take significant time and resources.
- May not always find an optimal path to the goal in large or complex environments.

Utility based Agents

delivery robot in a warehouse



How They Work

- **Utility Function**: These agents use a utility function to evaluate the "desirability" of different states or outcomes.
- **Optimization**: They aim to maximize their utility by selecting actions that lead to the highest utility.
- **Trade-offs**: They can balance between conflicting goals or priorities based on the utility values.

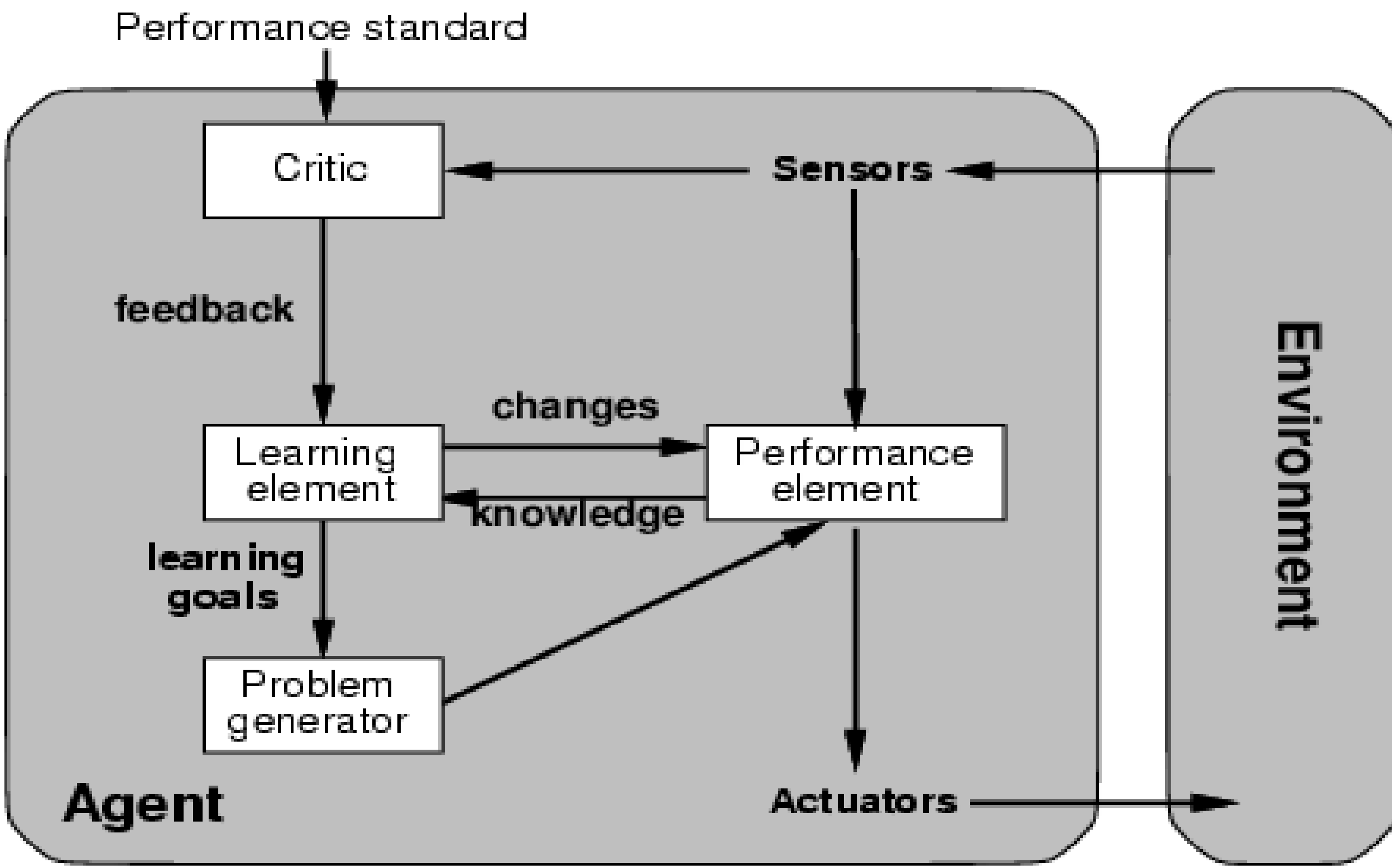
Advantages

- Highly flexible and adaptable to changing conditions.
- Can handle situations where multiple goals exist and trade-offs are required.

Disadvantages

- Designing a utility function that accurately represents the agent's preferences can be complex.
- Computational cost increases as the agent evaluates many possible actions.

Learning Agents



Type of Agent	Key Feature	Strengths	Weaknesses	Example
Simple Reflex Agents	Condition-action rules	Easy to design; works in simple environments	Fails in partially observable or complex environments	Thermostat
Agents with World Tracking	Maintains an internal state	Handles partial observability; adapts to changes	Higher computational cost	Robot vacuum cleaner
Goal-Based Agents	Goal-driven behavior	Flexible; supports reasoning and planning	Computationally expensive	GPS navigation system
Utility-Based Agents	Uses utility to maximize desirability	Balances trade-offs; highly adaptive	Complex to design utility functions	Autonomous cars

Environments

- environments refer to the external world in which an agent operates and interacts

- Types of environments
  - Fully Observable vs. Partially Observable
  - Deterministic vs. Stochastic
  - Episodic vs. Sequential
  - Static vs. Dynamic
  - Discrete vs. Continuous
  - Single Agent vs. Multi-Agent

Human Agent

- Sensors : Eyes, ears, and Other Organs
- Effectors : Hands ,Legs, Mouth, and any other body part

Robotic Agent

- Sensors : Cameras, Infrared range finders
- Effectors : Various Motors