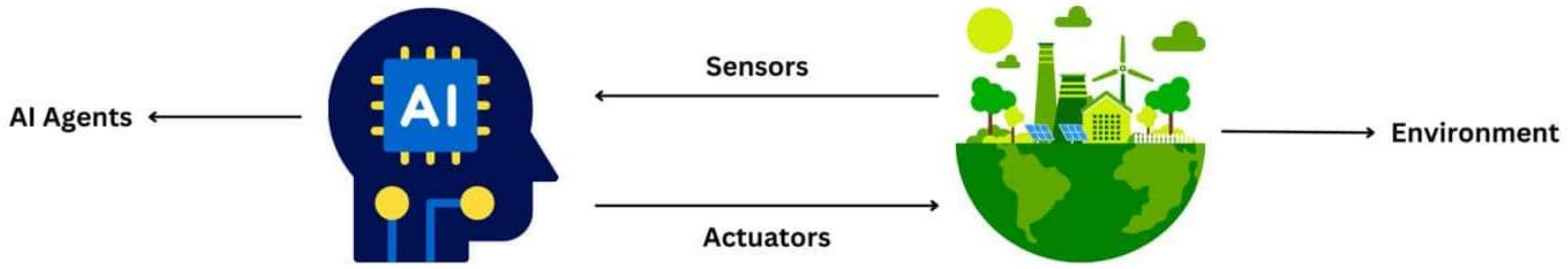
# ARTIFICIAL INTELLIGENCE & MACHINE LEARNING

**SESSION NO :3**

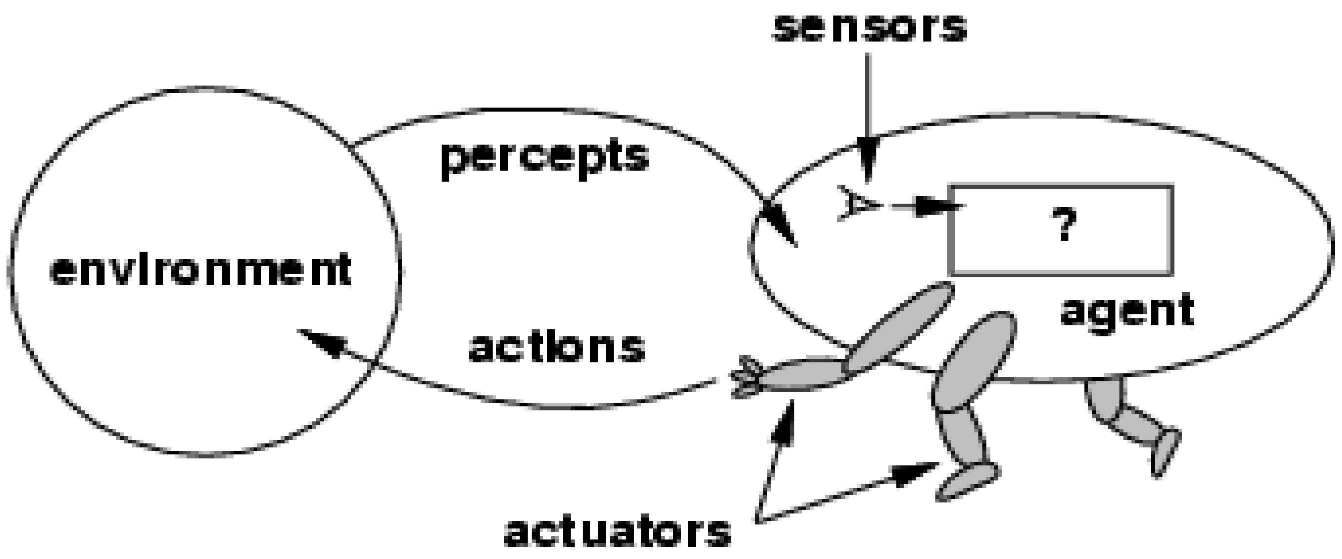

AI Agents
AI
Sensors
Actuators
Environment

## Intelligent Agent

- *Agent*: element in a system or environment that can trigger action.
- When deciding what to do, an agent makes decisions based on how they perceive their surroundings.
- The perception capability is usually called a *sensor*.
- The most recent perception or the full history (percept sequence) may influence the actions.

# Agents

- An agent is anything that can be viewed as:
  - perceiving its environment through sensors and
  - acting upon that environment through actuators
  - Assumption: Every agent can perceive its own actions (but not always the effects)

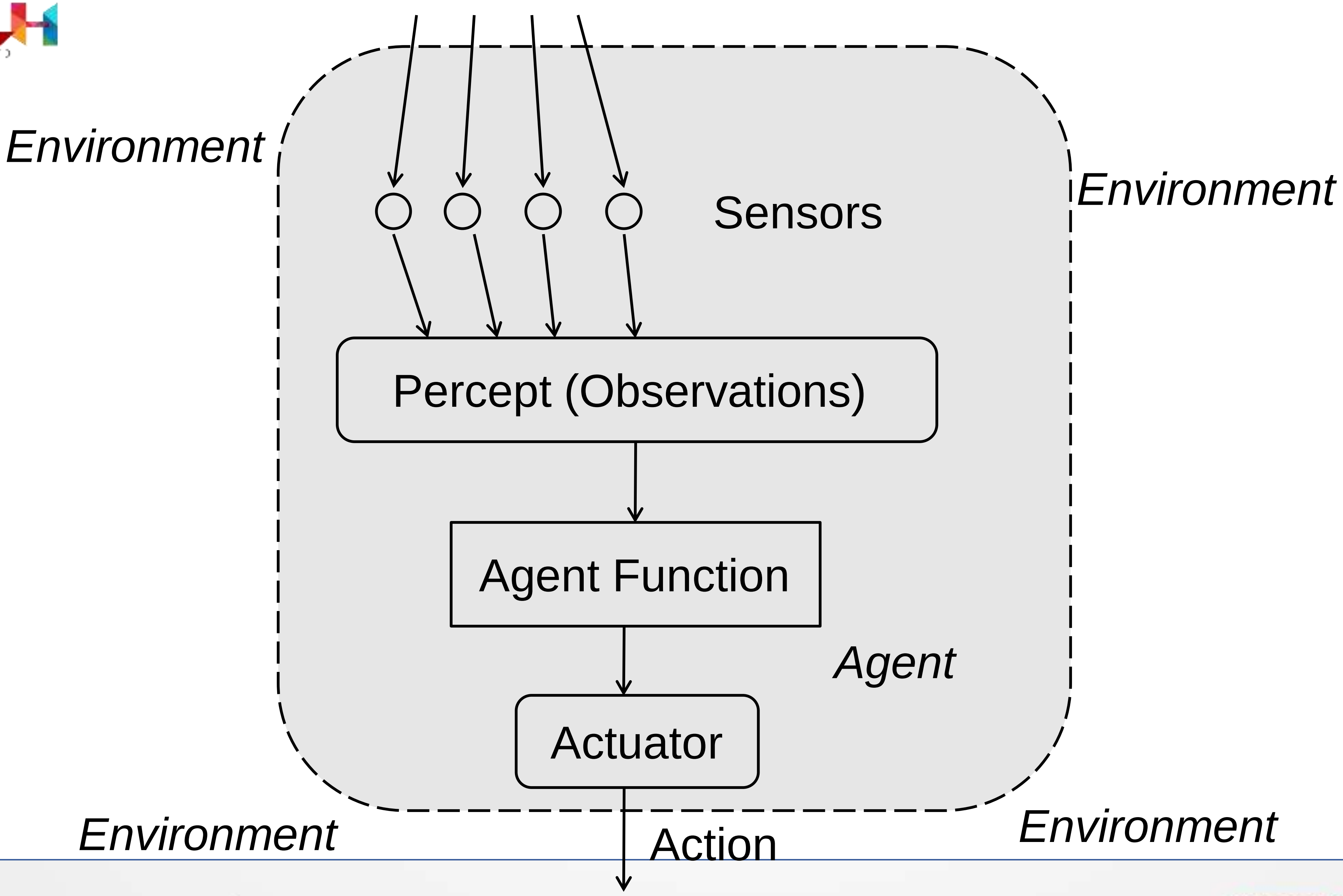


# Agents…

- Human agent:
  - Eyes, ears, and other organs for sensors;
  - Hands, legs, mouth, and other body parts for actuators
- Robotic agent:
  - cameras and infrared range finders for sensors;
  - various motors for actuators
- A software agent:
  - Keystrokes, file contents, received network packages as sensors
  - Displays on the screen, files, sent network packets as actuators

## Agent function

- A mathematical formula called the ***agent function*** converts a series of observations into actions.
- The function is implemented as the *agent program*.
- Actuator refers to the component of an agent that performs an action.
- environment → sensors → ***agent function*** → actuators → environment


Environment
Environment
Sensors
Percept (Observations)
Agent Function
Agent
Actuator
Environment
Action
Environment

## Rational agent

- A *rational agent* is one *who can make the best choice in any circumstance*.
- *A set of standards or testing ground for an agent's behaviour that serves as a performance measure.*
- Based on the agent's intended impact on the environment, performance metrics should be developed.

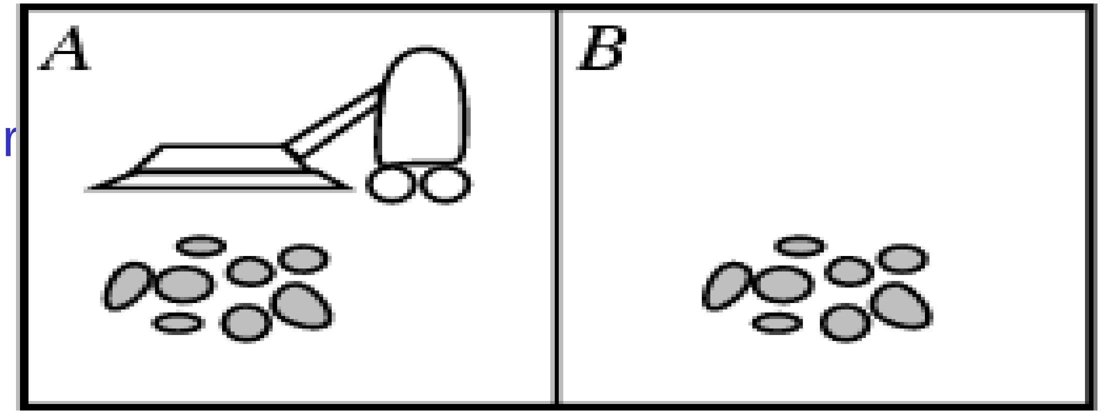
## Agent types

- Four basic types in order of increasing generality:
- Rather than a table how we can produce rational behavior from a small amount of code

  - Simple reflex agents
  - Model-based reflex agents
  - Goal-based agents
  - Utility-based agents

# Example: vaccum-agent

- **Percepts:**
  - Location and status, e.g., [A,Dir
- **Actions:**
  - Left, Right, Suck, NoOp



- **function Vacuum-Agent**([location,status]) returns an action
  - *if* status = Dirty *then* return Suck
  - *else if* location = A *then* return Right
  - *else if* location = B *then* return Left

# Vaccum cleaner agent

- Let’s assume the following
  - The performance measure awards one point for each clean square
  - at each time step, over a lifetime of 1000 time steps
  - The geography of the environment is known a priori but the dirt distribution and the initial location of the agent are not. Clean squares stay clean and the sucking cleans the current square. The Left and Right actions move the agent left and right except when this would take the agent outside the environment, in which case the agent remains where it is
  - The only available actions are Left, Right, Suck and NoOp
  - The agent correctly perceives its location and whether that location contains dirt
- Under these circumstances the agent is **rational**: its expected performance is at least as high as any other agent’s

# Vaccum cleaner agent

- Same agent would be **irrational** under different circumstances:
  - Once all dirt is cleaned up it will oscillate needlessly back and forth.
  - If the performance measure includes a penalty of one point for each movement left or right, the agent will fare poorly.
  - A better agent for this case would do nothing once it is sure that all the squares are clean.
  - If the clean squares can become dirty again, the agent should occasionally check and clean them if needed.
  - If the geography of the environment is unknown the agent will need to explore it rather than stick to squares A and B

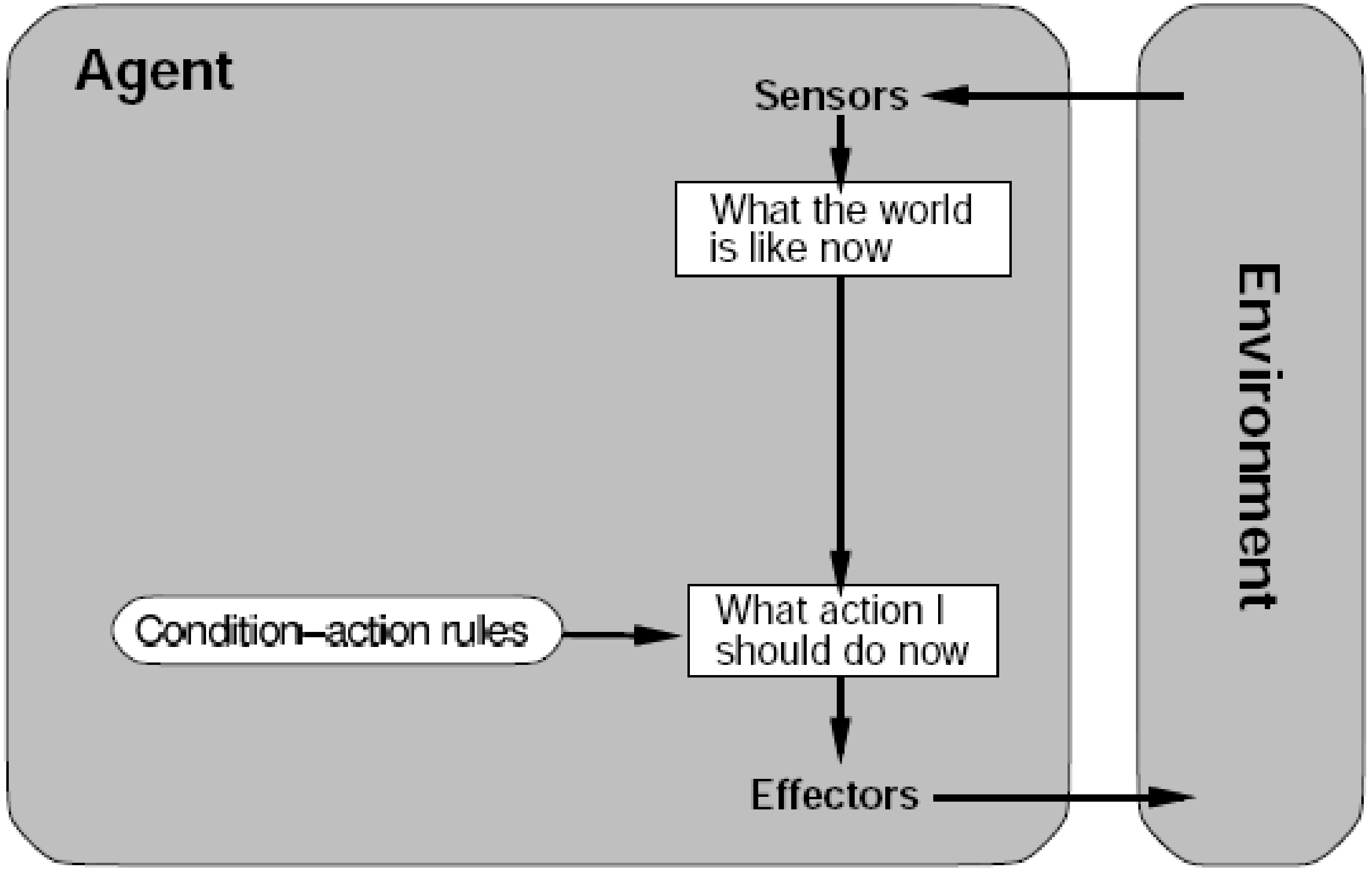
# Simple reflex agents

- Select actions on the basis of the current percept ignoring the rest of the percept history
- Example: simple reflex vacuum cleaner agent

```
function REFLEX-VACUUM-AGENT([location,status])
returns an action
if status = Dirty then return Suck
else if location = A then return Right
else if location = B then return Left
```

- **Condition-action-rule**
- Example: **if** *car-in-front-is-breaking* **then** *initiate-breaking*

## Simple reflex agents



## Simple reflex agents

```
function SIMPLE-REFLEX-AGENT(percept) returns an action
 static: rules, a set if condition-action rules
   state <-- INTERPRET_INPUT(percept)
     rule <-- RULE_MATCH(state, rules)
          action <-- RULE_ACTION[rule]
return action
```

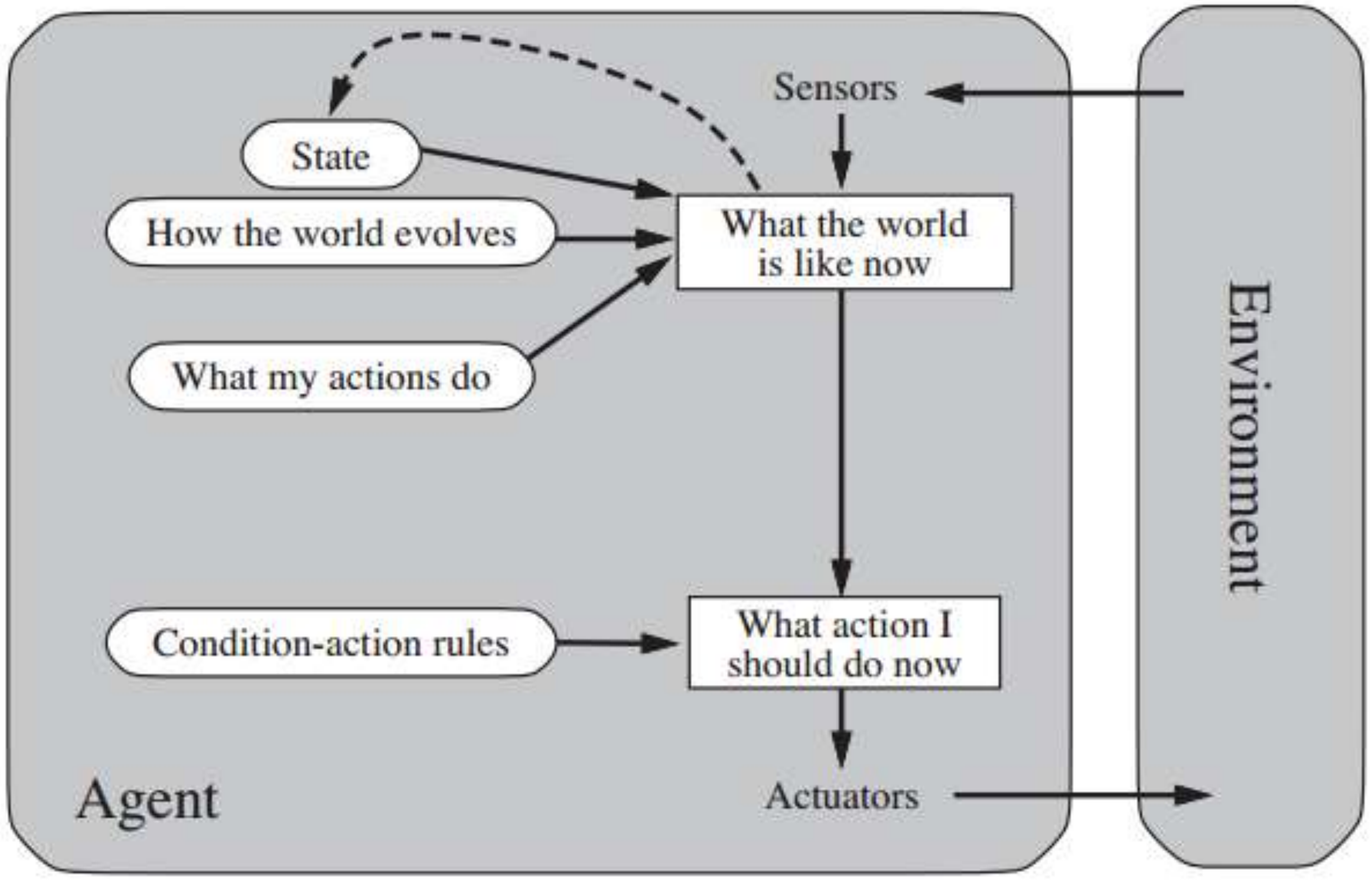
- Simple-reflex agents are simple, but they turn out to be of very limited intelligence
- The agent will work only if the correct decision can be made on the basis of the current percept – that is only if the environment is fully observable
- Infinite loops are often unavoidable – escape could be possible by randomizing

# Model-based reflex agents

- The agent should keep track of the part of the world it can't see now
- The agent should maintain some sort of internal state that depends on the percept history and reflects at least some of the unobserved aspects of the current state
- Updating the internal state information as time goes by requires two kinds of knowledge to be encoded in the agent program
  - Information about how the world evolves independently of the agent
  - Information about how the agent's own actions affects the world

- Model of the world – model based agents

# Model-based reflex agents



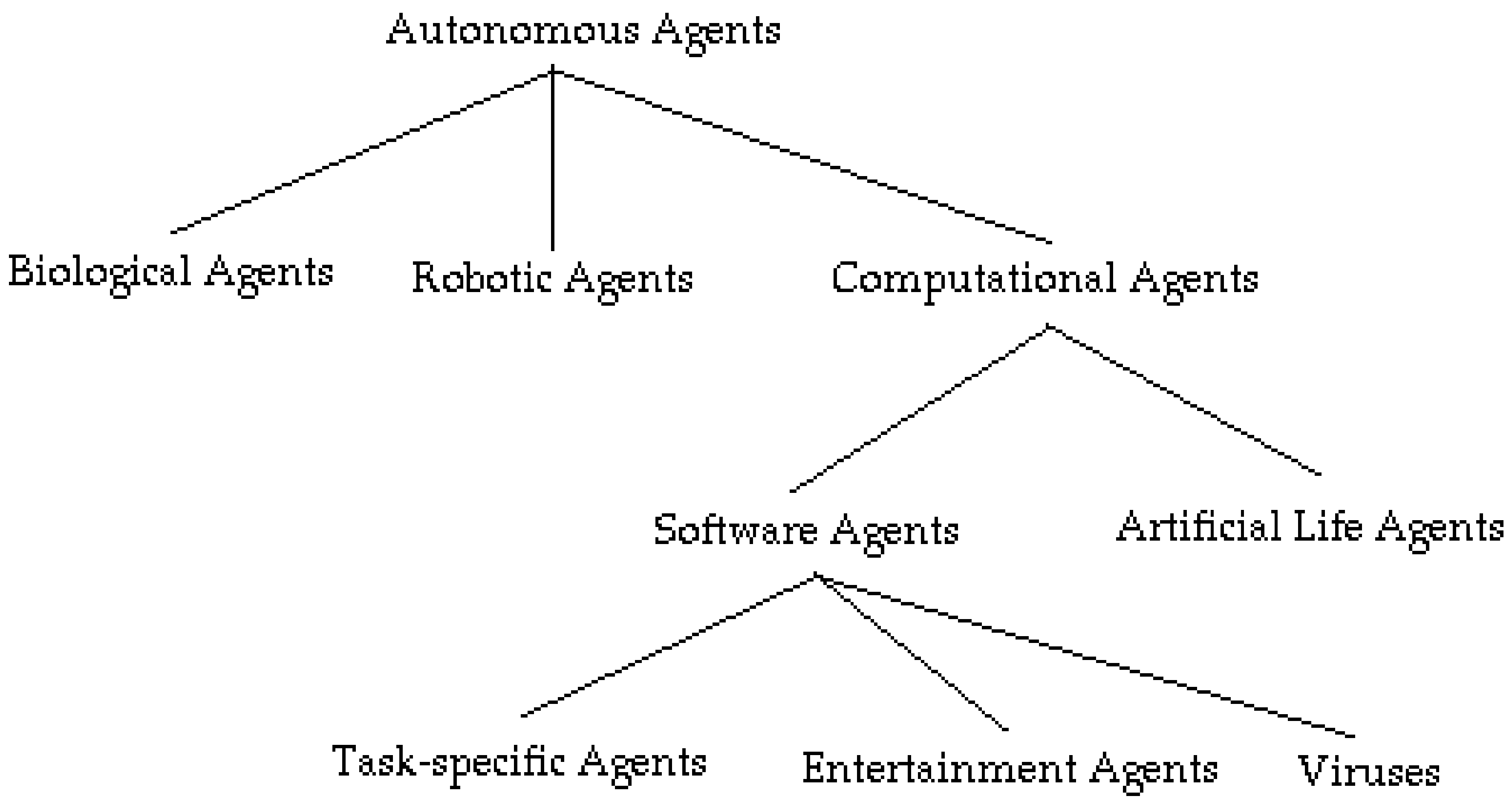
# Goal-driven agents

- The agent has a goal, and the action that will be taken will rely on both the current situation and the goal that the agent is attempting to achieve.
- The objective is sometimes simple to reach. In other cases, it entails planning, searching through a search space for potential answers, and coming up with a strategy.
- *Utility-based agents*: The utility function that calculates how near the current state is to the agent's goal is known to the agent.

# Learning agents

- Agents with the capacity to learn new skills through observations and acts.
- Components:
  - Learning component (modifies the performance component)
  - aspect of performance (selects actions)
  - exploration element (problem generator) and
  - feedback element (critic).

# Agent classification



## PEAS in Artificial Intelligence

- *PEAS stands for Performance measure, Environment, Actuator, Sensor.*
- The **PEAS system** is a critical framework used to categorize these agents based on their **performance, environment, actuators, and sensors**.
- Understanding the PEAS system is essential for grasping how different AI agents function effectively in diverse environments. Among these agents, Rational Agents are considered the most efficient, consistently choosing the optimal path for maximum efficiency.

## PEAS Representation in AI

- PEAS is a framework used to specify the structure of an intelligent agent in AI.
- It breaks down the agent's interaction with the environment into four key components:
  1. **Performance Measure**: The criteria that define the success of the agent's actions.
  2. **Environment**: The surroundings or the context in which the agent operates.
  3. **Actuators**: The mechanisms through which the agent interacts with the environment.
  4. **Sensors**: The tools the agent uses to perceive its environment.
- By defining these elements, PEAS provides a clear outline for designing and evaluating intelligent systems, ensuring they are equipped to perform their tasks effectively.

## P: Performance Measure

- *Performance measure is the unit to define the success of an agent. Performance varies with agents based on their different precepts.*
- Performance measure is a quantitative measure that evaluates the outcomes of an agent's actions against a predefined goal. The performance measure is crucial because it guides the agent's decision-making process, ensuring that it acts in a way that maximizes its success.
- For example, in a self-driving car, the performance measure could include criteria such as safety (avoiding accidents), efficiency (minimizing travel time), and comfort (ensuring a smooth ride). The car's AI will aim to optimize these factors through its actions.

## E: Environment

- Environment is the surrounding of an agent at every instant. It keeps changing with time if the agent is set in motion.
- There are 5 major types of environments:
  - Fully Observable & Partially Observable
  - Episodic & Sequential
  - Static & Dynamic
  - Discrete & Continuous
  - Deterministic & Stochastic
- The environment includes all external factors and conditions that the agent must consider when making decisions. The environment can vary significantly depending on the type of agent and its task.
- For instance in the case of a smart thermostat, the environment for a smart thermostat includes the rooms in the house, the outside weather conditions, the heating or cooling system, and the presence of people, all of which the thermostat interacts with to maintain the desired temperature efficiently..

## A: Actuators

- An actuator is a part of the agent that delivers the output of action to the environment.
- They are responsible for executing the actions decided by the agent based on its perceptions and decisions. In essence, actuators are the "hands and feet" of the agent, enabling it to carry out tasks.
- The actuators for a smart thermostat include the heating system, cooling system, and fans, which it controls to adjust the room temperature and maintain the desired comfort level.
- The design and choice of actuators are crucial because they directly affect the agent's ability to perform its functions in the environment.

# S: Sensors

- Sensors are the receptive parts of an agent that takes in the input for the agent.
- Sensors collect data from the environment, which is then processed by the agent to make informed decisions. Sensors are the "eyes and ears" of the agent, providing it with the necessary information to act intelligently.
- The sensors for a smart thermostat include temperature sensors to measure the current room temperature, humidity sensors to detect moisture levels, and motion sensors to determine if people are present in the house.
- The quality and variety of sensors used in an AI system greatly influence its ability to perceive and understand its environment.

# Examples of PEAS in AI

AUTONOMOUS VEHICLES

ROBOTIC VACUUM CLEANERSS

GAME-PLAYING AI

# Advantages of PEAS in AI

**Structured Design**: Provides a clear framework for designing intelligent agents by breaking down their components.

**Versatility**: Applicable to various AI systems, from simple bots to complex autonomous agents.

**Goal-Oriented**: Ensures that agents are designed with specific, measurable objectives, leading to better performance.

**Systematic Development**: Facilitates organized planning and development, making the process more efficient.

# Disadvantages of PEAS in AI

**Complexity**: Can be complex to implement in dynamic environments with many variables.

**Over-Simplification**: Might oversimplify real-world scenarios, leading to gaps in agent behavior.

**Resource-Intensive**: Requires significant resources to accurately define and implement each PEAS component.

**Limited Adaptability**: May struggle to adapt to unexpected changes if not designed with enough flexibility.