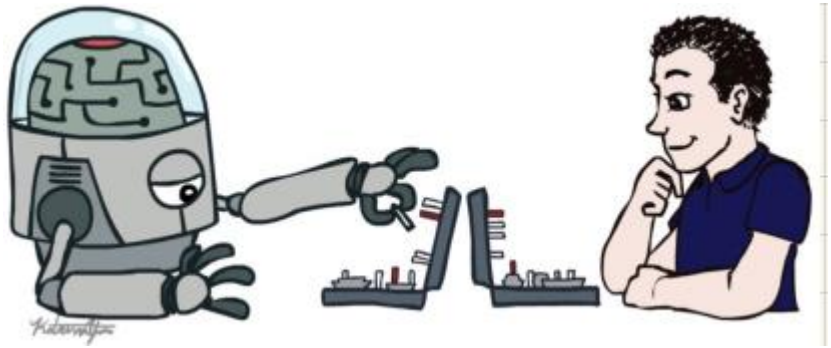
Artificial Intelligence (BCSE306L)

Module 01 - Introduction

- Structure of the Agent



Dr. Durgesh Kumar

Assistant Professor (Senior),
SCOPE VIT Vellore

Lecture Outline

- What makes an Agent Intelligent ?
  - Agent = Architecture + Program
  - The Agent Program
  - Example of Agent – Cleaning Robot
- Types of Agent – architecture, algorithm, properties, usecase
  - Simple Reflex Agents
  - Model based Reflex Agents
  - Goal based Reflex Agents
  - Utility based Agents
  - Learning based Agents

Is Your Coffee machine SMART ?

🕐 **7:00 AM. Monday morning. Your coffee machine starts brewing…**

- It didn't talk. It didn't ask. =>Yet, it acted *just right.*
- **Is it intelligent?**
- **Is it making decisions?**
- **Or is it just following instructions?**

Is Your Coffee machine SMART ?

👉 Today's Question: What really makes an AI agent *intelligent*?

🕖 **7:00 AM. Monday morning. Your coffee machine starts brewing…**

- It didn't talk. It didn't ask. =>Yet, it acted *just right.*
- **Is it intelligent?**
- **Is it making decisions?**
- **Is it *thinking*?**
- **Or is it just following instructions?**

What really makes an AI agent *intelligent ?*

Class discussion activity

- Does it need to learn ?
- Is it reacting fast enough ?
- What about understanding goals ?

Discuss in group of 3-4 and answer!

- Learning from mistakes ?
- Making decision on its own ?
- Achieving Goals ?
- Understanding Context?

Structure of *intelligent* AI agent*!*

Class discussion activity

- Simple Reflex Agent => fast reactions.
- Goal-based → purpose-driven
- Utility-based → smart trade-offs
- Learning → adaptation over time

Discuss in group of *3-4* and answer!

- Learning from mistakes ?
- Making decision on its own ?
- Achieving Goals ?
- Understanding Context?

What is an Agent ?

- Excellent! You've just described different dimensions of intelligence.

- Interestingly, in AI, all of these behaviors arise from how we **structure the agent** — its internal architecture and programming.

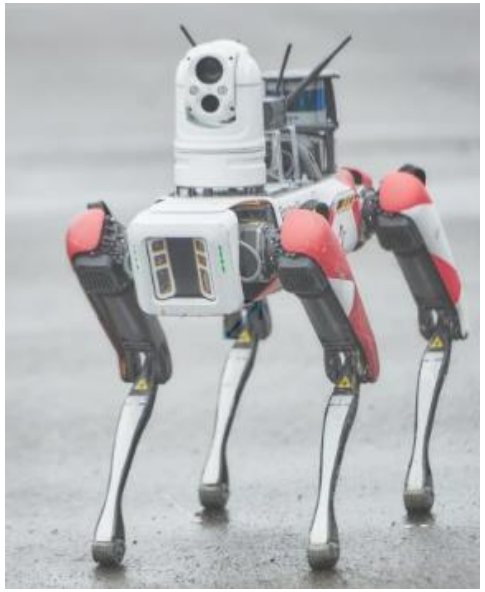
Agent = Architecture + Program

- **Architecture**: The machinery it runs on.
- **Program**: The software component that implements the agent's behavior.

What is an Agent ?

Agent = Architecture + Program

- **Architecture**: The machinery it runs on.
- **Program**: The software component that implements the agent's behavior.



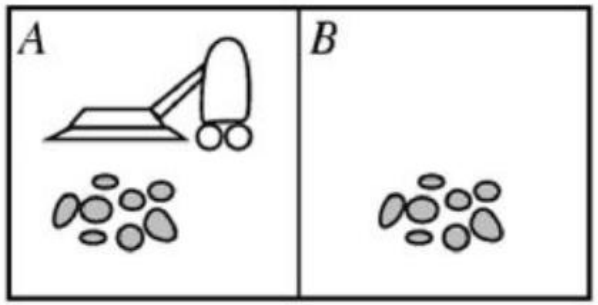
```
function TABLE-DRIVEN-AGENT(percept) returns an action
    persistent: percepts, a sequence, initially empty
                table, a table of actions, indexed by percept sequences,
initially fully specified

    append percept to the end of percepts
    action ← LOOKUP(percepts, table)
    return action
```

- Is this Agent ?
- **Hardware (Architecture)**
- ❑ **Behaviour (Program)**

The Agent Program

- Explain the logic-based structure:
  - **Perception → Action**



- **Percept to Action Mapping**

[A, clean] => Right
[A, dirty] => Clean
[B, clean] => Left
[B, dirty] => Clean

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

Type of Agents

- Simple reflex agent
- Model-based reflex agent
- Goal based Agents
- Utility based Agents
- Learning based Agents

- Let's now look at the various types of agents — from the **simplest reflex machines** to advanced learning agents — and understand what gives them their '**intelligence**."
- Let us understand each agent with **architecture**, **algorithmic logic**, **properties**, and **real-world use case**.

Simple Reflex Agent

- **Example**: Fire/Smoke Detector
  - Acts immediately when it sense smoke
    - No memory, just condition, action, rules.

- React to **current percept** using condition-action rules.
- No internal memory.

- 🧩 **Algorithm:** IF condition THEN action

- 📦 **Use Case:** Smoke detector, dishwasher.

https://github.com/aimacode/aima-pseudocode/tree/master

Simple Reflex Agent

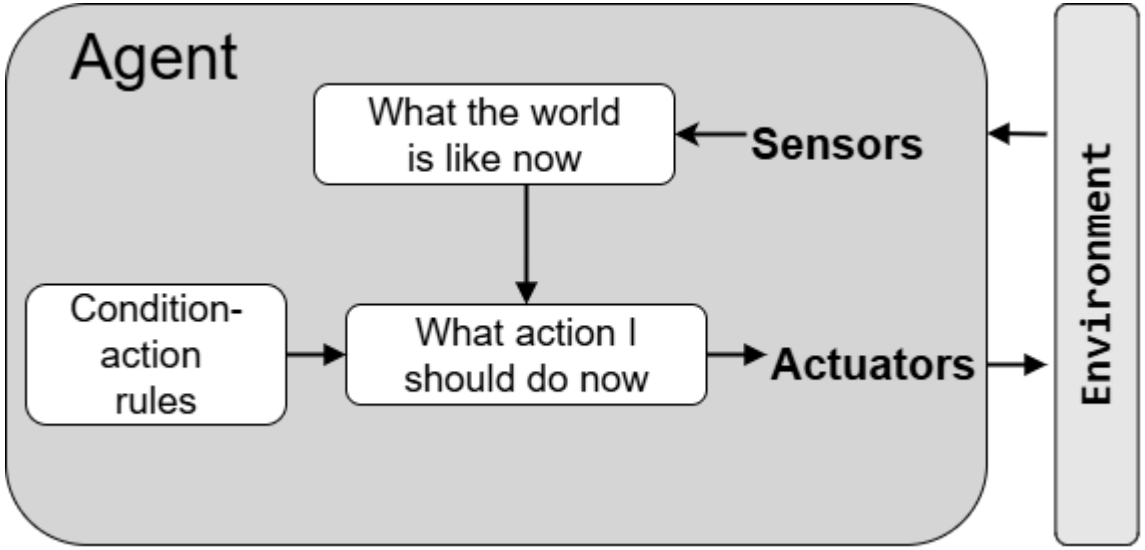


Fig: Architecture Diagram of Simple Reflex Agent

```
function SIMPLE-REFLEX-AGENT(percept ) returns an action
  persistent rules, a set of condition–action rules
   state ← INTERPRET-INPUT(percept )
   rule ← RULE-MATCH( state, rules)
   action ← rule.ACTION
return action
```

Fig: Algorithm of Simple Reflex Agent

Model based reflex Agent

- Example: Robot Vacuum Cleaner (e.g. Roomba)
  - Uses a **map of the environment** to **avoid obstacles** and **clean efficiently**.
- Maintain internal state of the world.
- Handles partially observable environments.
- 🧠 **Architecture:** Current Percept + Internal Model → Action
- 📦 **Use Case:** Indoor robot vacuum (e.g., Roomba with obstacle memory)

Model based reflex Agent

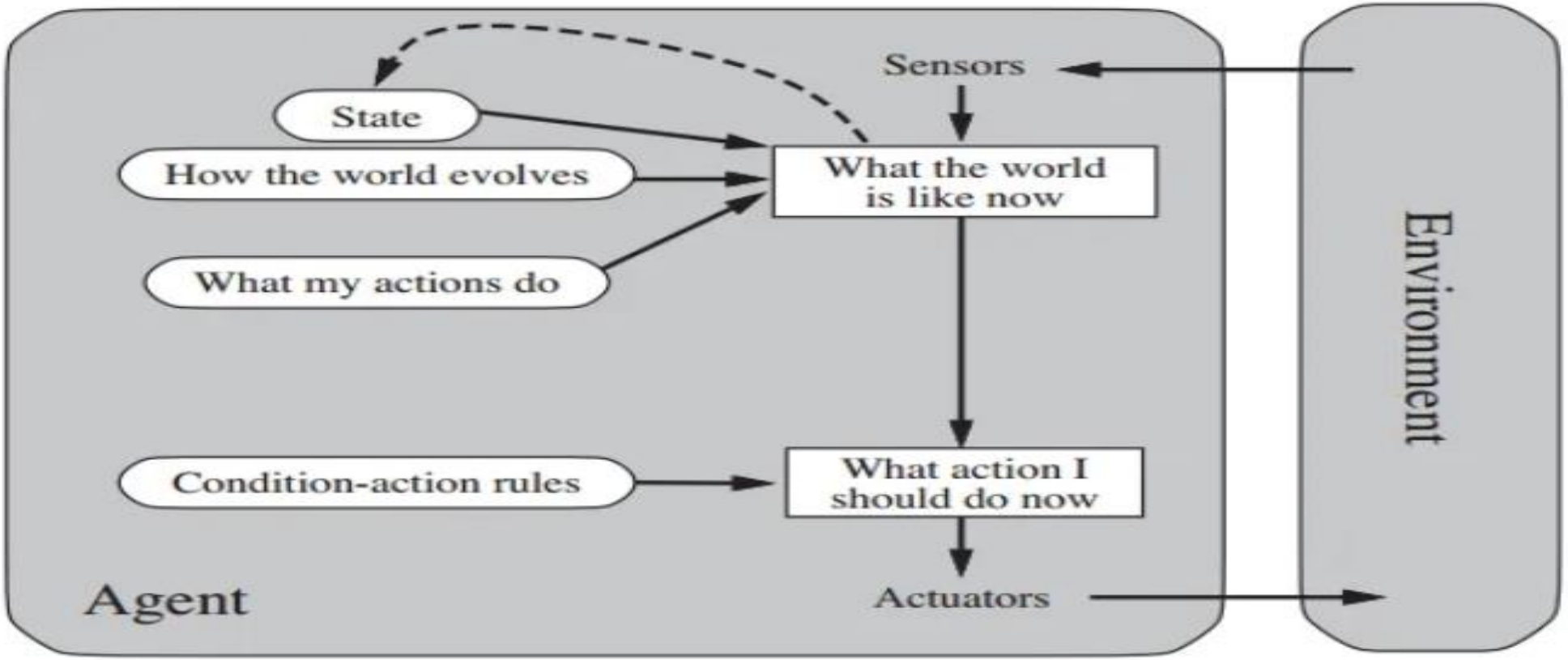


Fig: Architecture Diagram of Model based Reflex Agent

Model based reflex Agent

function MODEL-BASED-REFLEX-AGENT(percept) **returns** an action

persistent ***state***, the agent's current conception of world state, *transition_model*, *a* description of how **next state** depends on **current state** and **action**

sensor_model ← **a description of how the current world state is reflected in the agent's percepts**

rules ← **a set of condition–action rules**

action ← **the most recent action, initially none**

state ← **UPDATE-STATE**(**state, action, percept, transition_model, sensor_model**)

rule ← **RULE-MATCH**(**state, rules**)

action ← **rule.ACTION**

return action

Fig: Algorithm of Model based Reflex Agent

Goal-Based Agents

- Example: Autonomous Delivery Drone
  - **Plans its route** dynamically to reach a destination while **avoiding hazards**.
  - Acts based on desired outcomes (goals).
  - Needs search and planning.
- 🧩 **Algorithm:** A* Search / Planning Graph
- 📦 **Use Case:** Pathfinding in Google Maps, automated delivery drones.
- 💡 **Think-Pair-Share:** "Why might a goal-based agent be inefficient in real-time games?"

Goal based reflex Agent

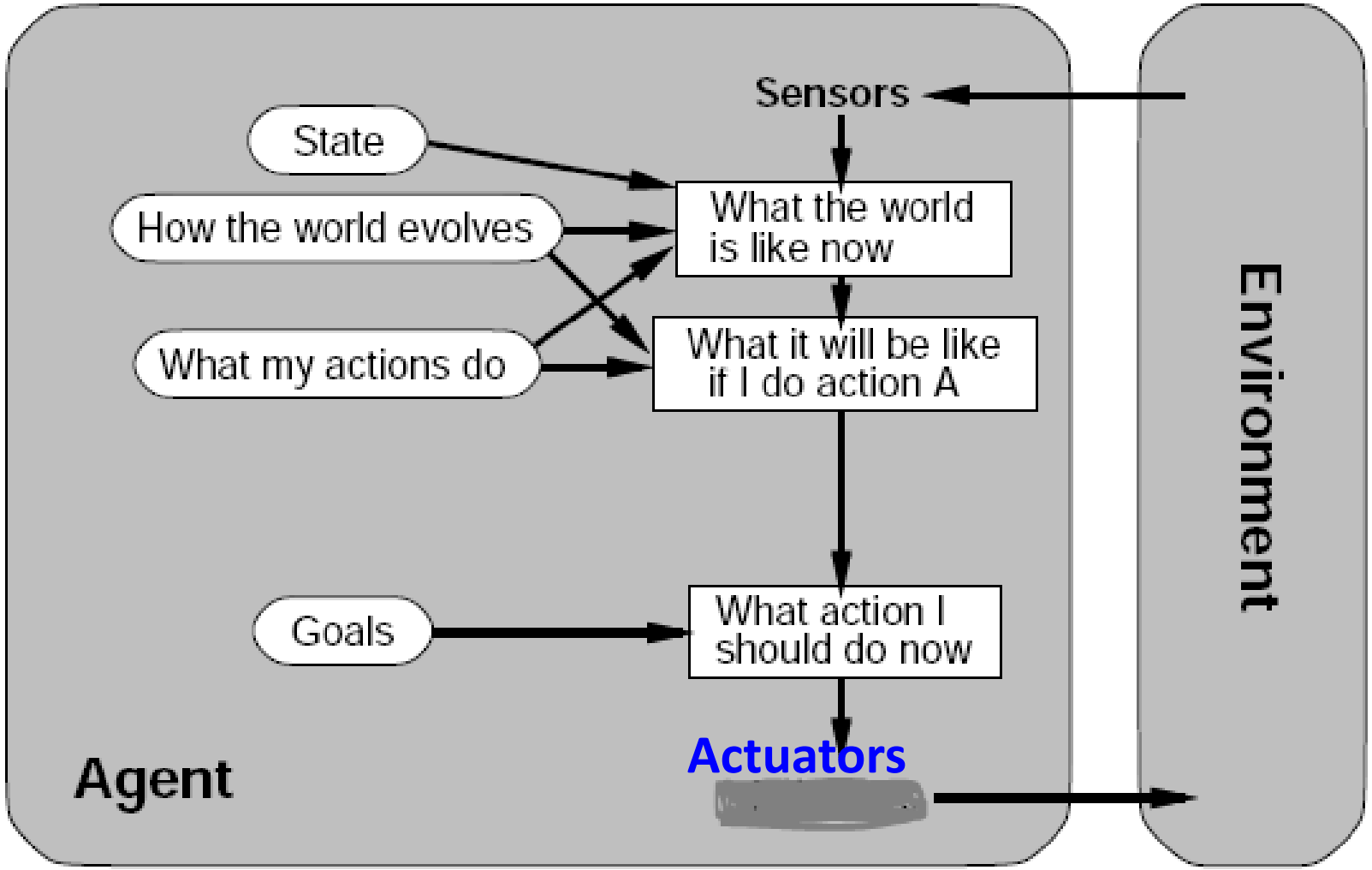
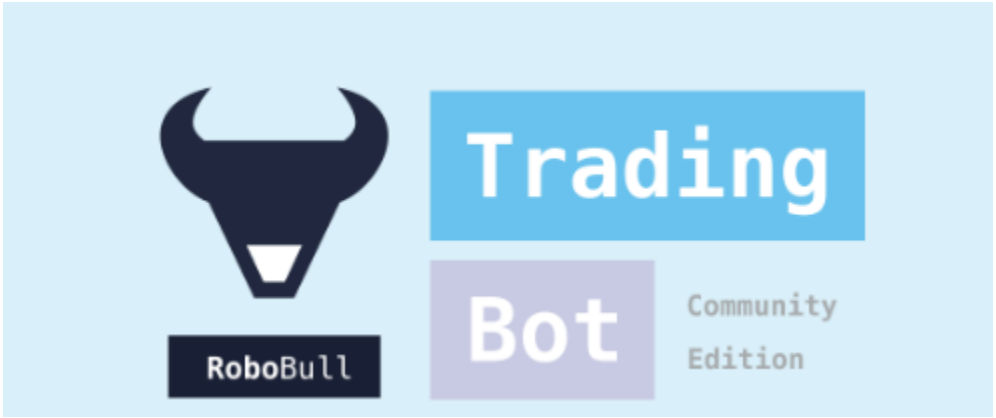


Fig: Architecture Diagram of Goal based Agent

Utility-Based Agents



https://github.com/michaeljwright/robobull-trading-bot

- Example: Stock Trading Bot
  - **Chooses trade** that **maximizes** expected financial return **not just reaching a goal**.
  - Measures *how good each state is*.
  - Maximizes expected utility.

- **Concepts:** Utility Function, Optimization
- **Use Case:** AI in trading bots, recommendation systems.
- **Exercise:** Rate 3 outcomes for a recommender system. Which has higher utility?

Utility based reflex Agent

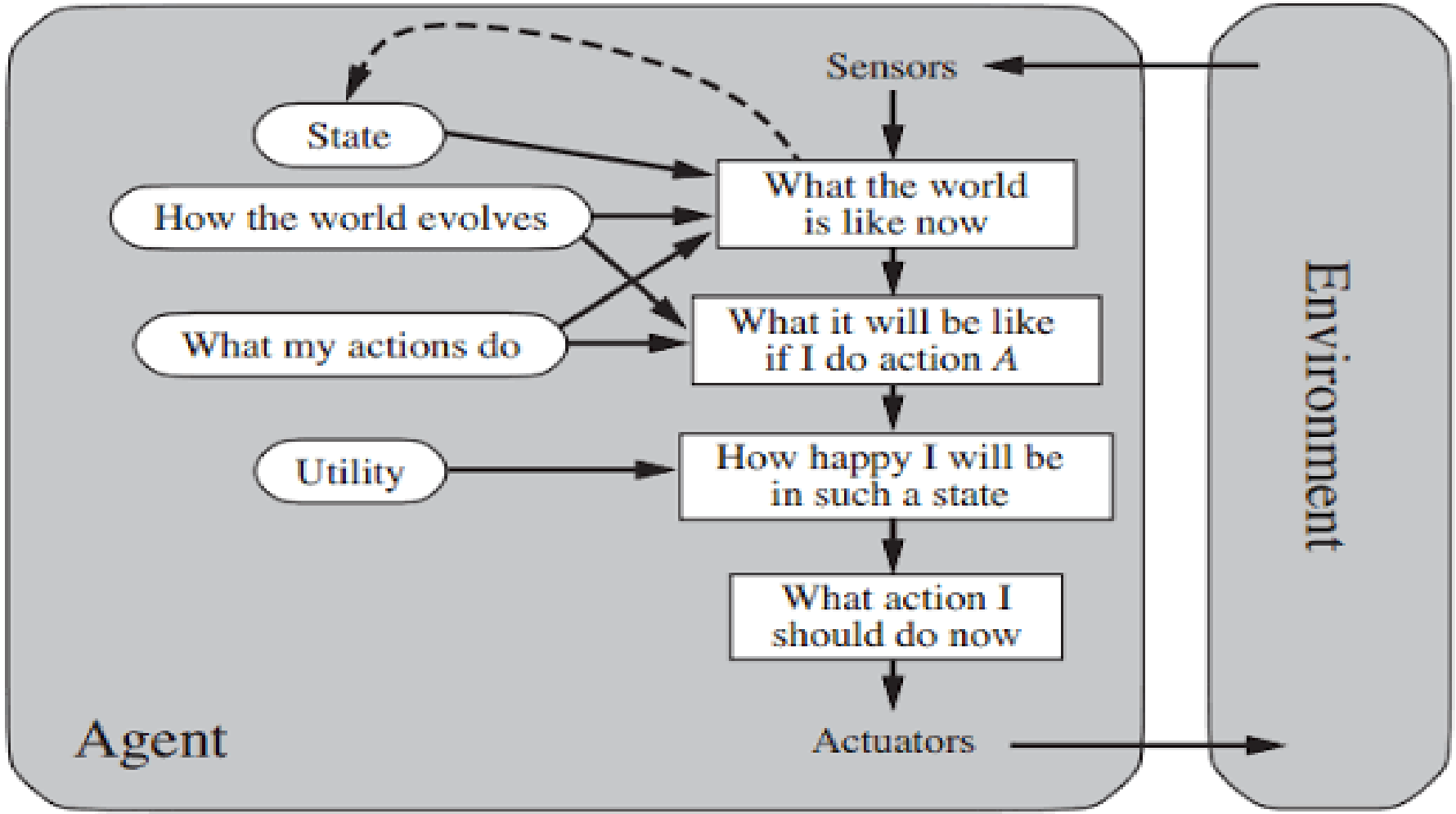


Fig: Architecture Diagram of Utility based Reflex Agent

Learning-Based Agents

- Example: ChatGPT
  - **Continuously improve** its **responses** by learning from **user interactions** **and** **feedback**.
  - Improves performance over time.
- 📈 **Components:** Learning element, Performance metric, critic, problem generator
- 📦 **Use Case:** ChatGPT, AlphaZero (chess/Go), adaptive traffic signals.
- 🧪 **Mini Case Study:** Draw diagrams to show evolution from rule-based to learning agent.

Learning based reflex Agent

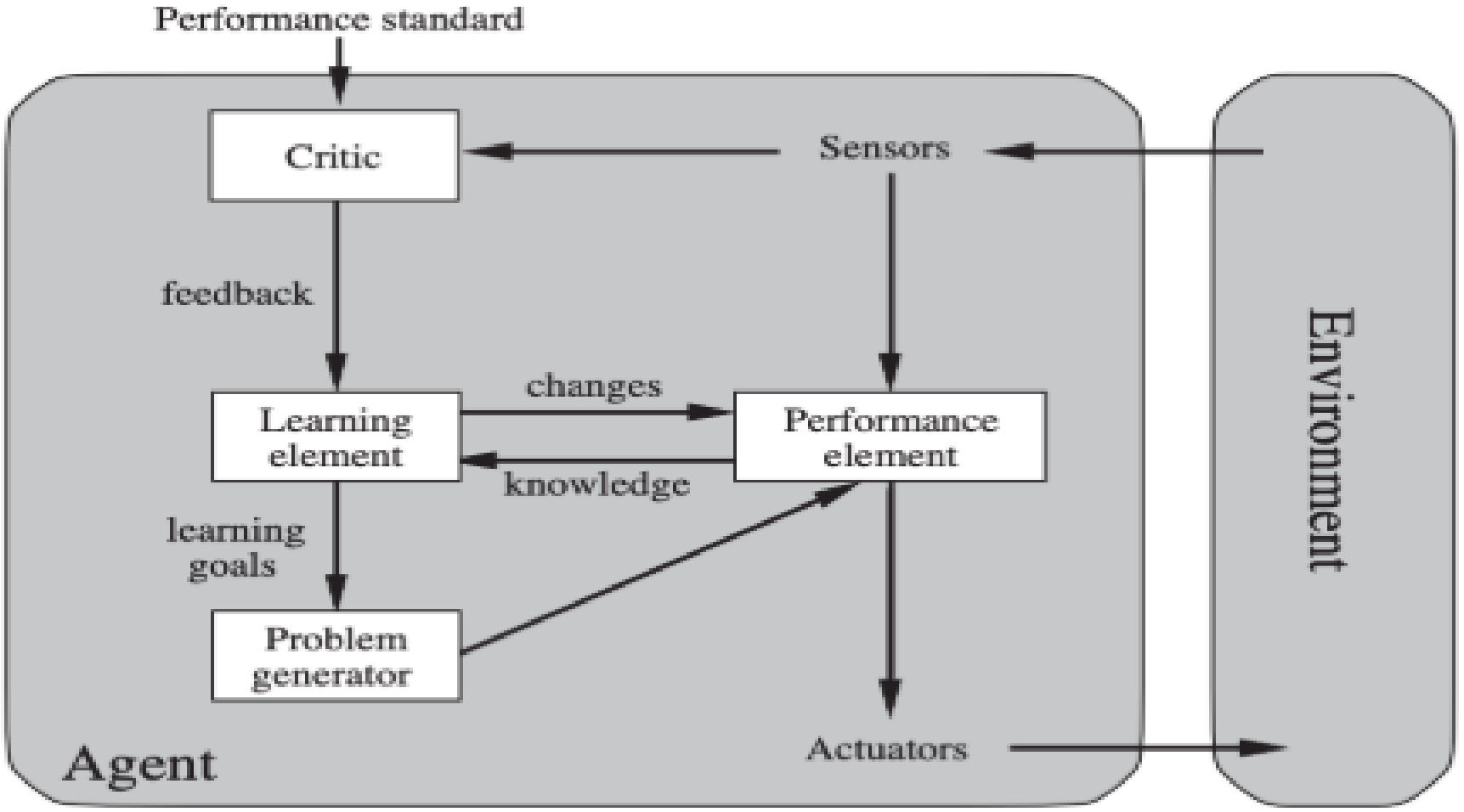


Fig: Architecture Diagram of Learning based Reflex Agent

Comparison Table of different Agents

Agent Type	Memory	Goal-Oriented	Learning	Use Case Example
Simple Reflex	❌	❌	❌	Thermostat
Model-Based	✅	❌	❌	Robot Vacuum
Goal-Based	✅	✅	❌	Navigation Assistant
Utility-Based	✅	✅	Optional	Stock Trading Bot
Learning Agent	✅	✅	✅	Autonomous Driving

Sample Questions – Q1

- Consider an artificial agent learning to play chess, where the agent learns the game's rulesand optimal moves through multiple plays and feedback from critics. Which type of agent would be most suitable for a chess-playing agent? Justify your answer. Also, briefly describe the agent architecture with a suitable diagram.**[5 Marks]**

Sample Questions – Q1

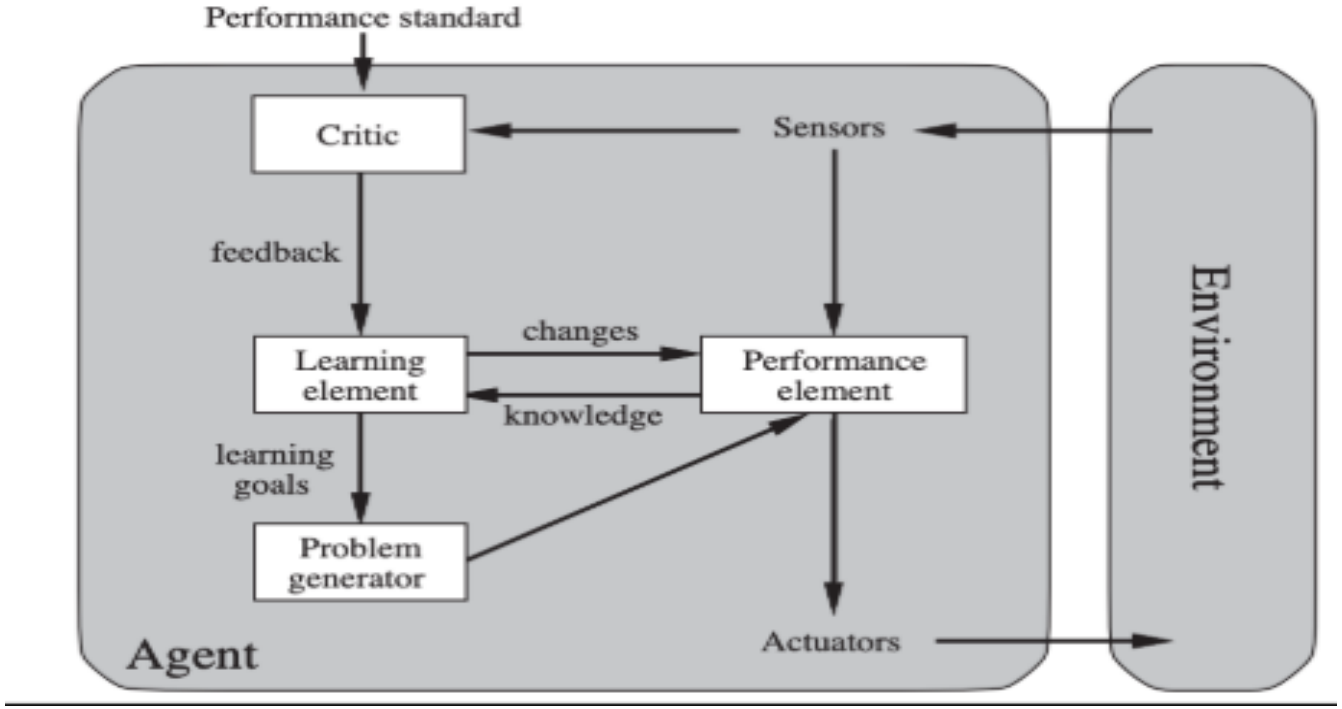
- Consider an artificial agent learning to play chess, where the agent learns the game's rulesand optimal moves through multiple plays and feedback from critics. Which type of agent would be most suitable for a chess-playing agent? Justify your answer. Also, briefly describe the agent architecture with a suitable diagram. **[5 Marks]**

- Ans: Learning based Agent is most suitable as the agent **learn game's rules** and **optimal moves** through multiple plays and feedback from critics. [2 Marks]

Sample Questions – Q1

Ans: a) Learning based Agent is most suitable as the agent **learn** **game's rules** and **optimal moves** through multiple plays and feedback from critics. [2 Marks]

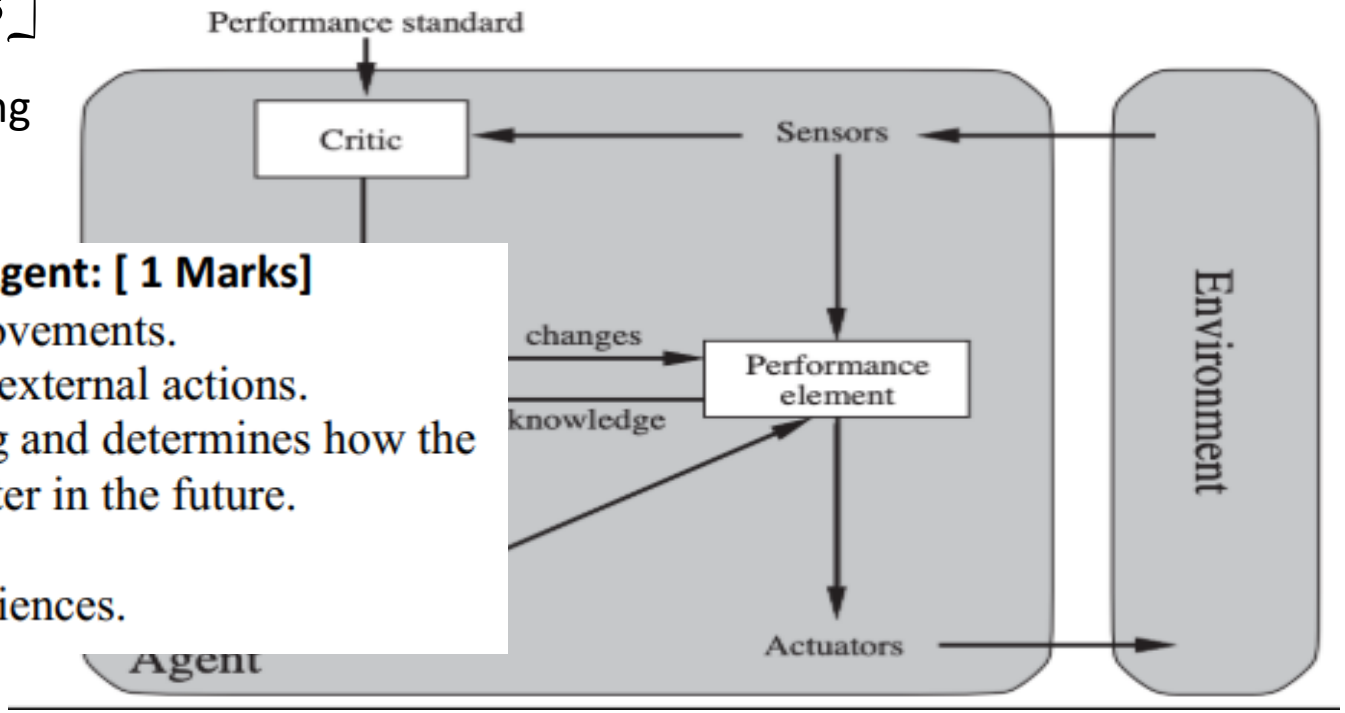
b) Architecture Diagram of Learning based Reflex Agent **[2 Marks]**



Sample Questions – Q1

Ans: a) Learning based Agent is most suitable as the agent **learn** **game's rules** and **optimal moves** through multiple plays and feedback from critics. [2 Marks]

b) Architecture Diagram of Learning based Reflex Agent **[2 Marks]**



Following are the major components of Leanring based Agent: [1 Marks]

i. **Learning element** is responsible for making improvements.
ii. **Performance element** is responsible for selecting external actions.
iii. **Critic** provides feedback on how the agent is doing and determines how the performance element should be modified to do better in the future.
iv. **Problem generator** is responsible for suggesting actions that will lead to new and informative experiences.

Thank you

Questions ?