Artificial Intelligence
ADC503

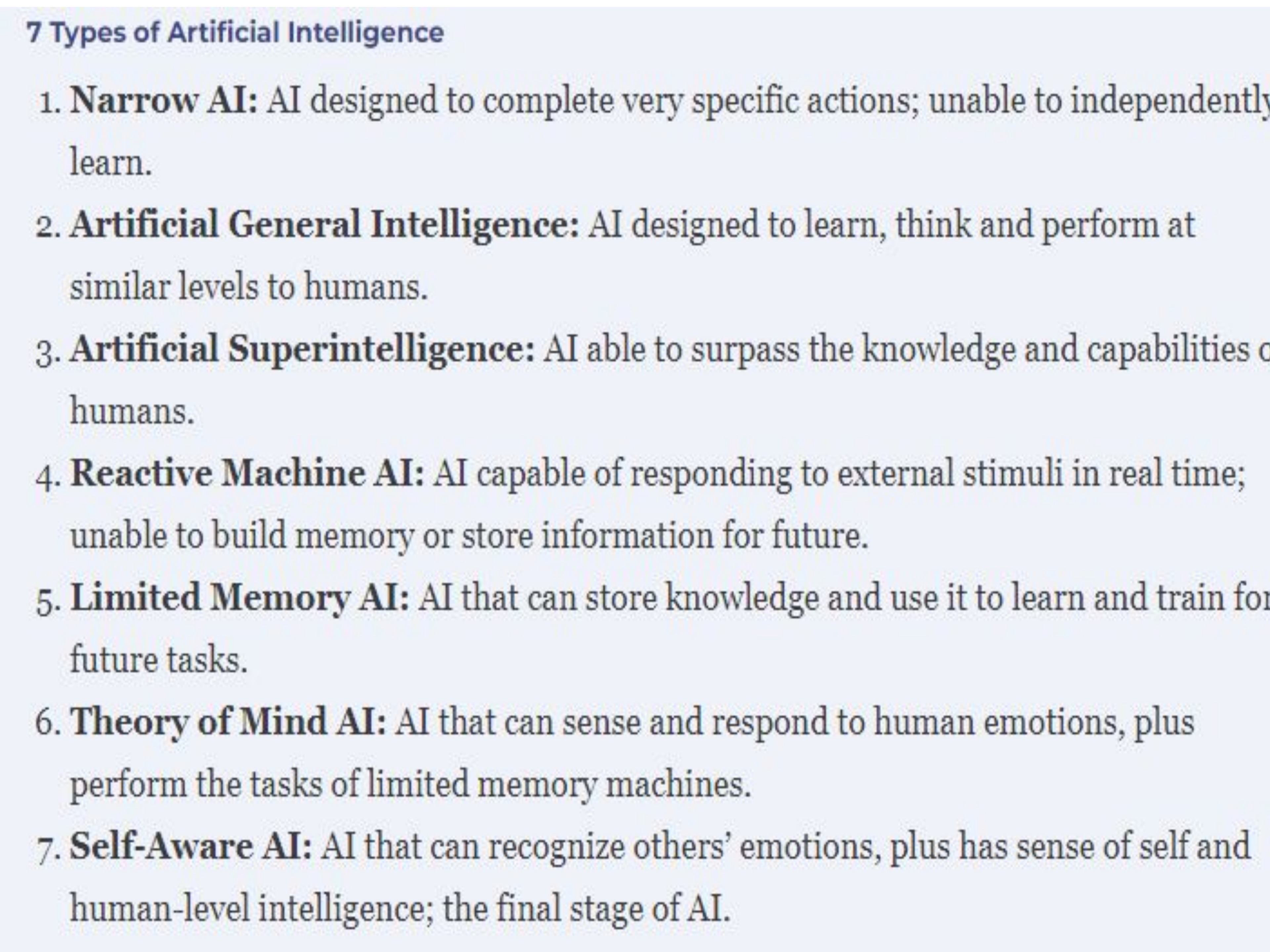
Lecture 4

Prof. Smita S. Mane

Asst. Prof.

Dept. of AI & DS,

VESIT,Chembur



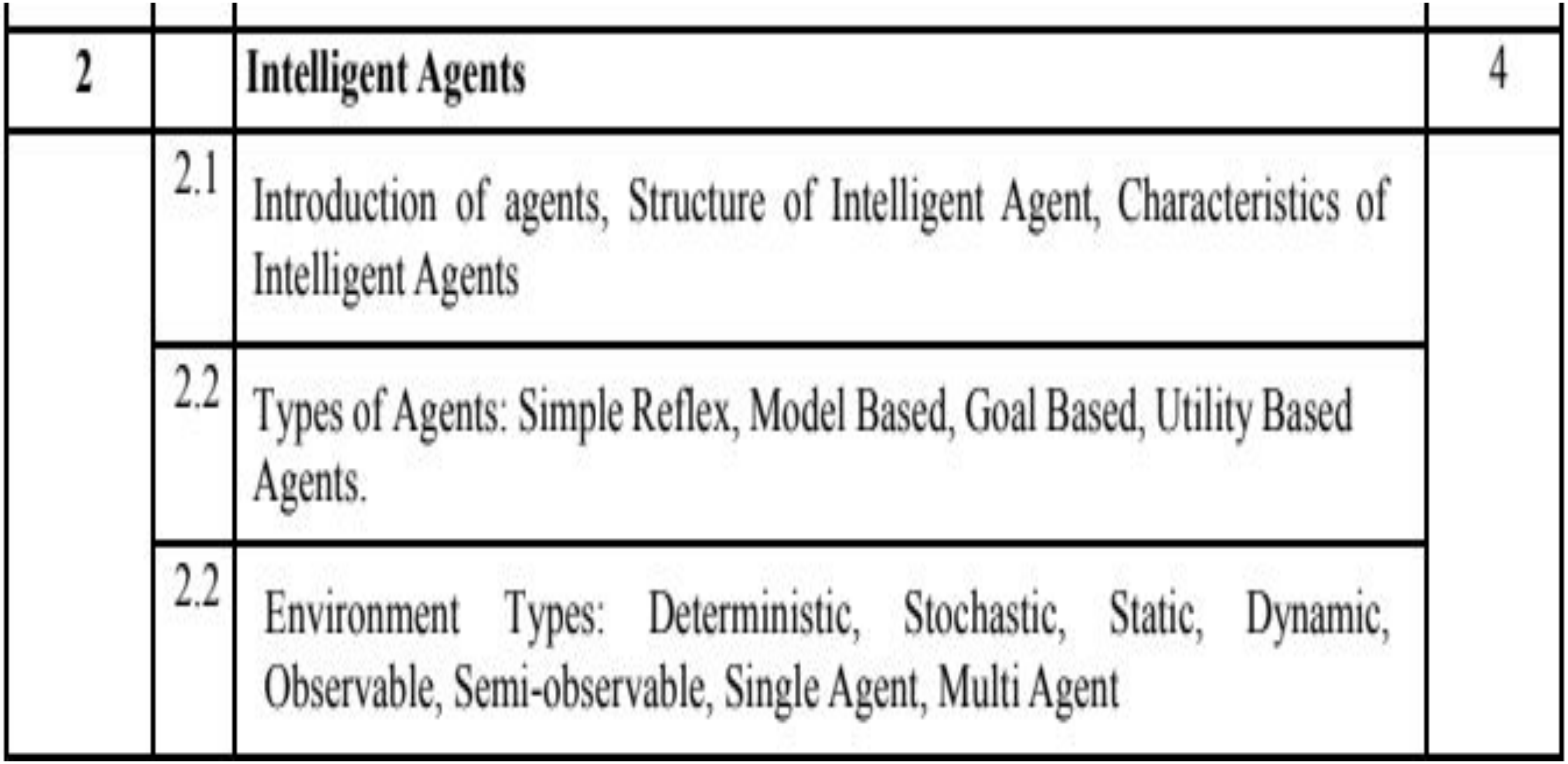
7 Types of Artificial Intelligence

1. **Narrow AI:** AI designed to complete very specific actions; unable to independentl learn.
2. **Artificial General Intelligence:** AI designed to learn, think and perform at similar levels to humans.
3. **Artificial Superintelligence:** AI able to surpass the knowledge and capabilities o humans.
4. **Reactive Machine AI:** AI capable of responding to external stimuli in real time; unable to build memory or store information for future.
5. **Limited Memory AI:** AI that can store knowledge and use it to learn and train fo future tasks.
6. **Theory of Mind AI:** AI that can sense and respond to human emotions, plus perform the tasks of limited memory machines.
7. **Self-Aware AI:** AI that can recognize others' emotions, plus has sense of self and human-level intelligence; the final stage of AI.

Curriculum

Intelligent Agents

Module 2



2		**Intelligent Agents**	4
	2.1	Introduction of agents, Structure of Intelligent Agent, Characteristics of Intelligent Agents	
	2.2	Types of Agents: Simple Reflex, Model Based, Goal Based, Utility Based Agents.	
	2.2	Environment Types: Deterministic, Stochastic, Static, Dynamic, Observable, Semi-observable, Single Agent, Multi Agent	

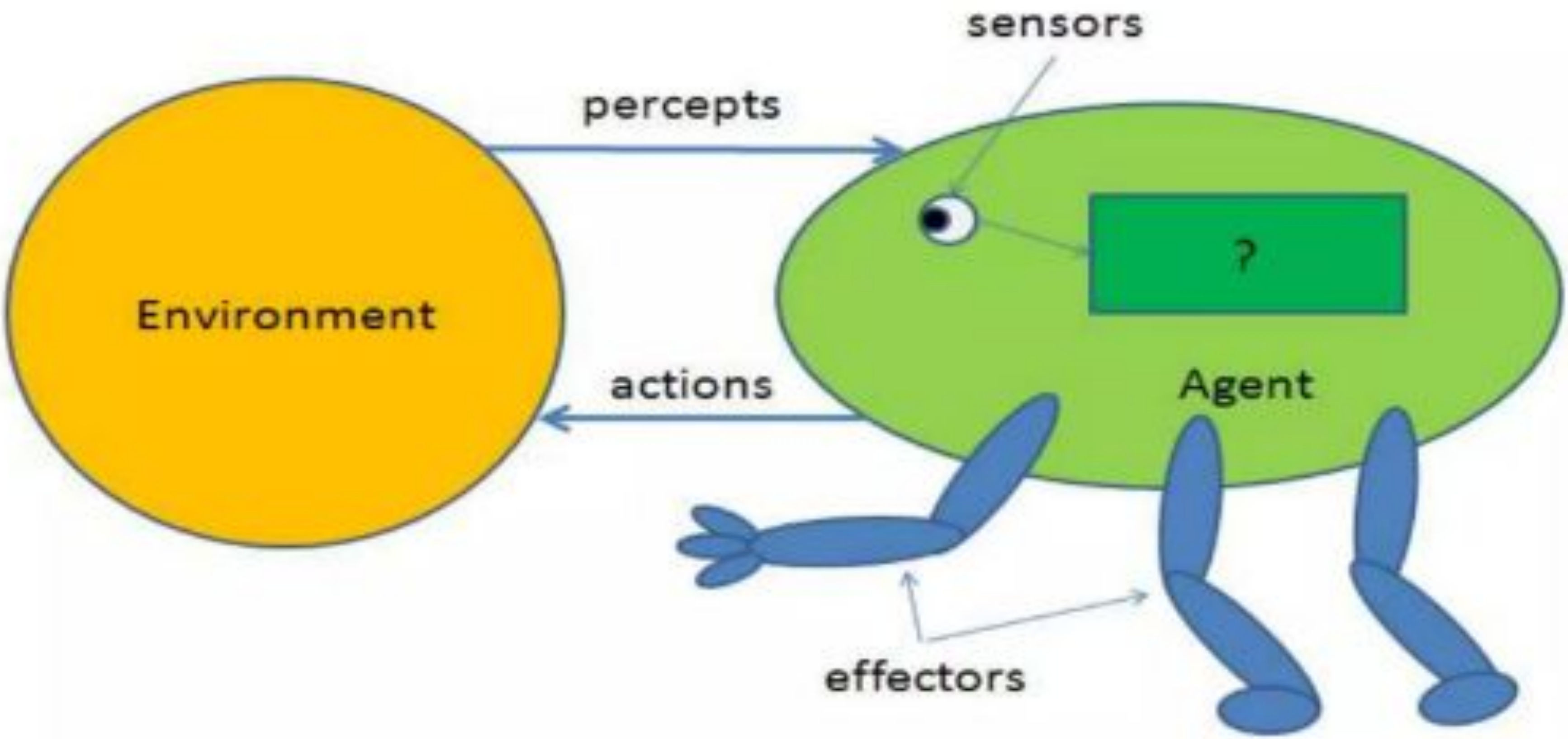
Agents

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

- A human agent has eyes, ears, and other organs for sensors, and hands, legs, mouth, and other body parts for effectors.

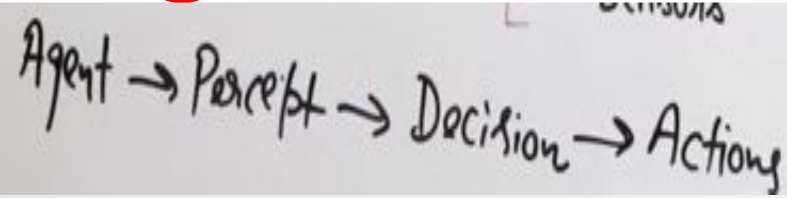
- A robotic agent substitutes cameras and infrared range finders for the sensors and various motors for the effectors.

Agent and Environment

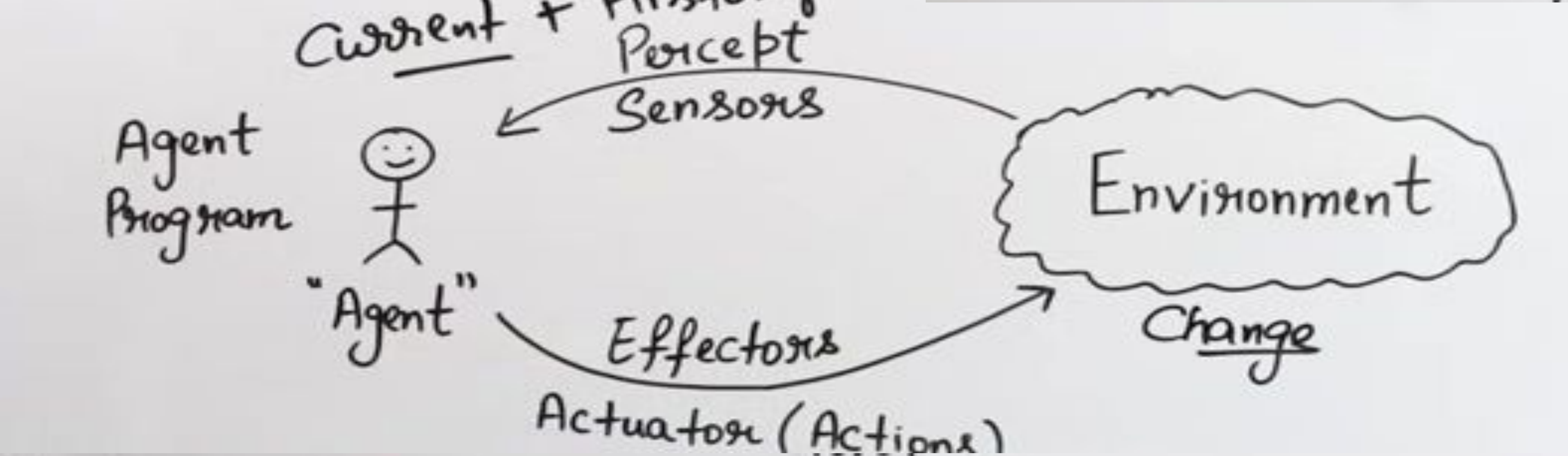


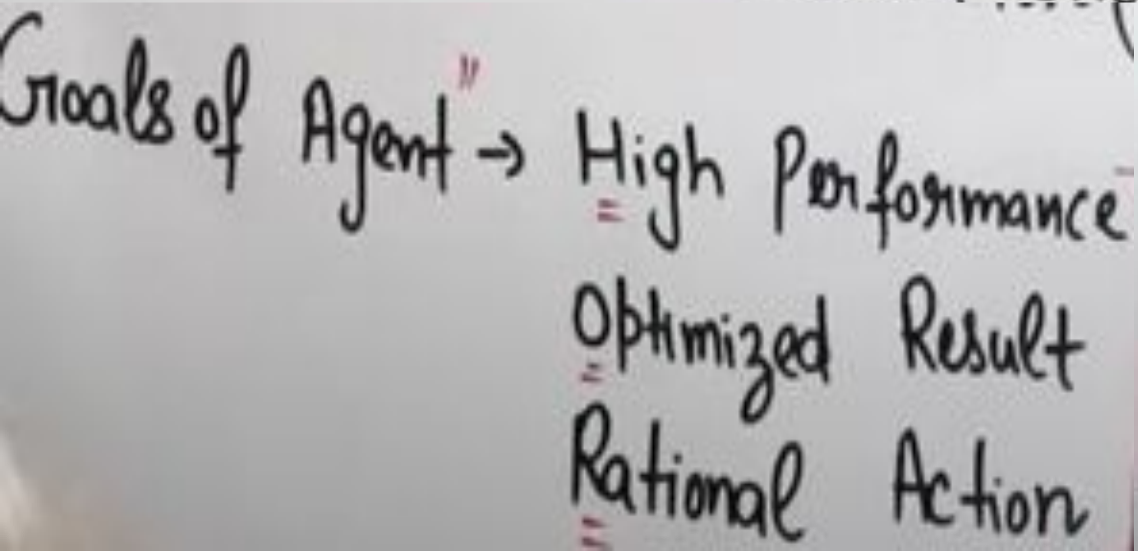
Introduction of agents

Agents / Intelligent Agents



Agent → Percept → Decision → Actions

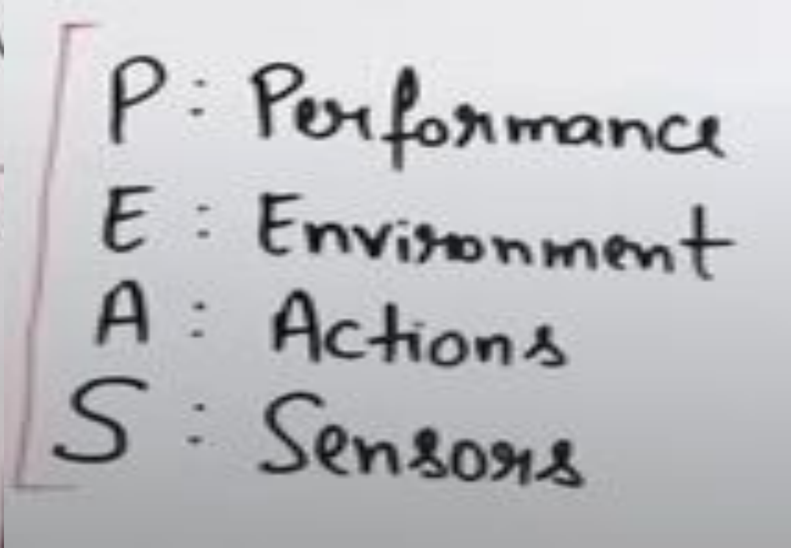




Goals of Agent" → High Performance

Optimized Result

Rational Action



P : Performance

E : Environment

A : Actions

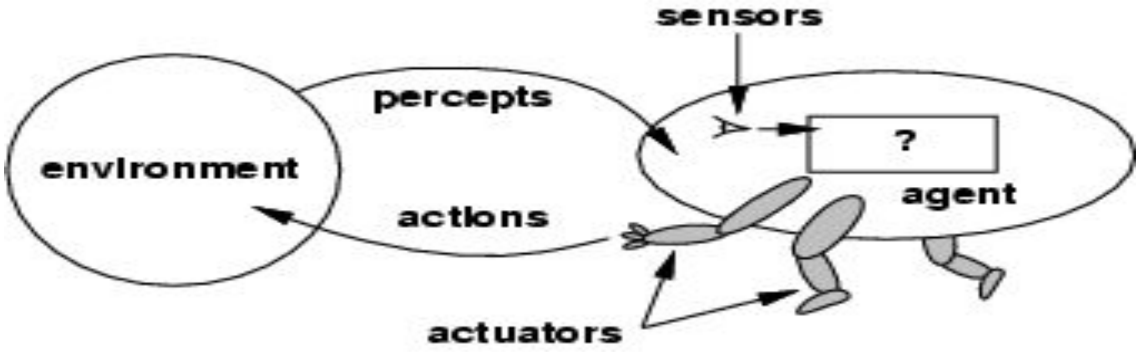
S : Sensors

Agents (rational agents)

- An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators
- 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

- **"An agent is a computer software system whose characteristics are situatedness, autonomy, adaptivity and sociability."**

Agents & Environments



- Percept: refer to the agents perceptual input at any given instant.
- The agent function maps any given percept sequence to an action [*f*: $\mathcal{P}^*$ □ $\mathcal{A}$]
  Agent function will be implemented by agent program
- The agent function is an abstract mathematical description ; agent program is a concrete implementation, running on agent architecture

Agents

- Operate in an environment.
- Perceives its environment through sensors.
- Acts upon its environment through actuators/effectors.
- Have goals.
- Intelligent Agent:

  must sense,

  must act,

  must be autonomous(to some extent)

  must be rational.

Sensors & Effectors

- An agent Perceives its environment through sensors.
- The complete set of inputs at a given time is called percept.
- The current percept, or a sequence of percepts can influence the actions of an agent.

Sensors & Effectors

- It can change the environment through effectors.
- An operation involving an actuator is called an action.
- Actions can be grouped in to action sequences.
- So an agent program implement mapping from percept sequences to actions.

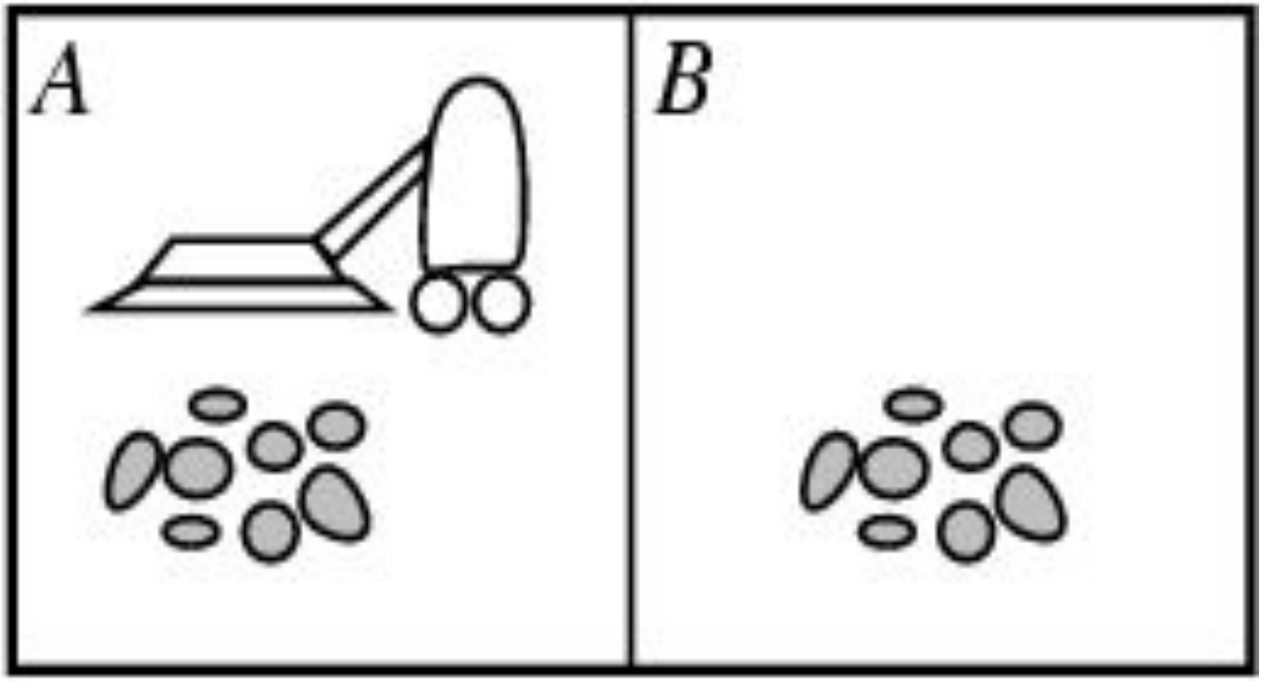
Percept : input at given instance

Percept sequence : complete history

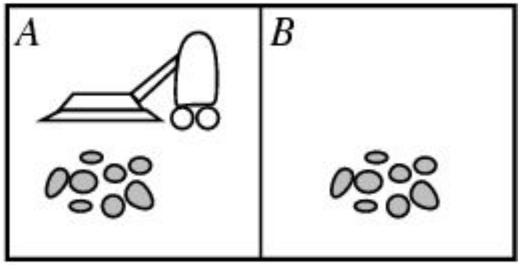
Tabulating the agent function that describes any given agent
(for most agent it will be a long table)

Given an agent to experiment with, we can in principle construct this table by trying out all possible percept sequence and recording which actions the agent does in response

Vacuum-cleaner world



Vacuum-cleaner world



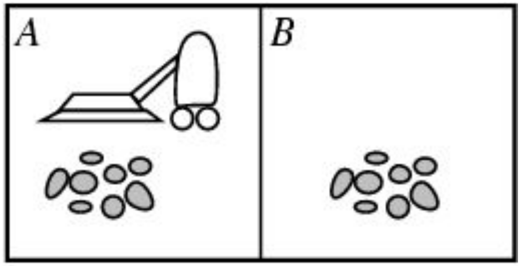
Input

- Percepts: location and contents, e.g., [A,Dirty]
- Actions: *Left*, *Right*, *Suck*, *NoOp*

Output

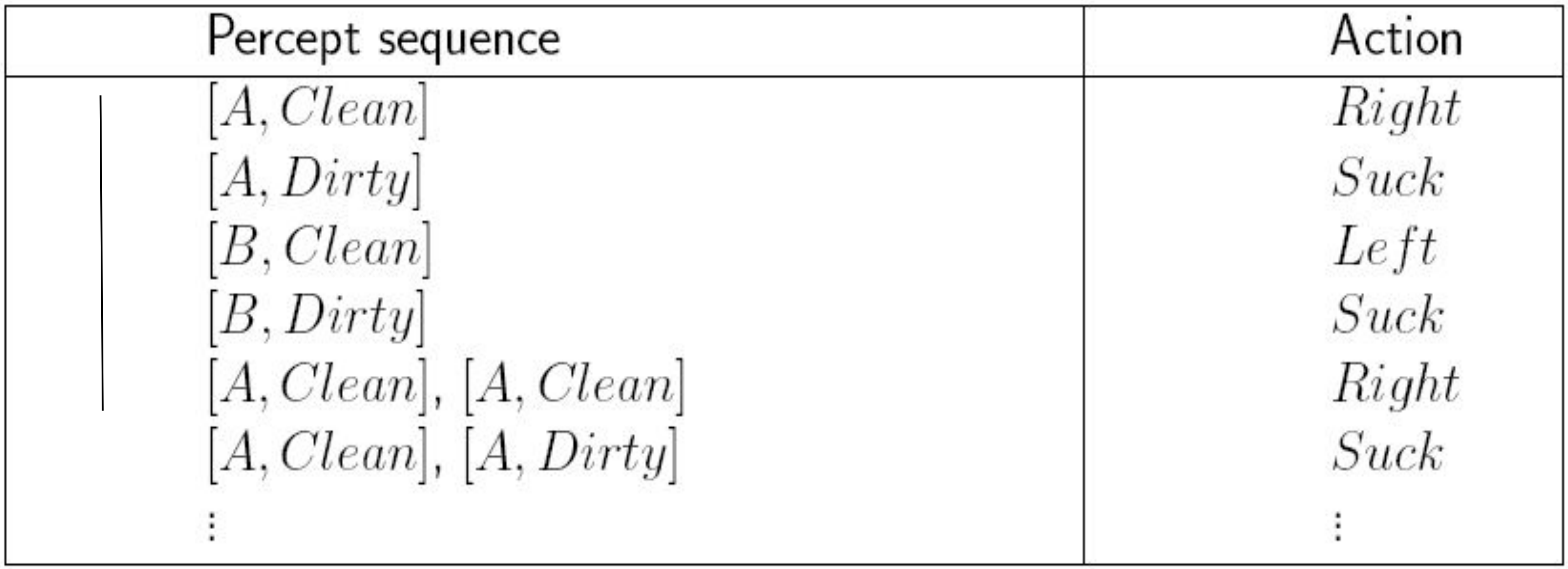
Example: Vacuum Cleaner Agent

history : complete sequence agent has perceived

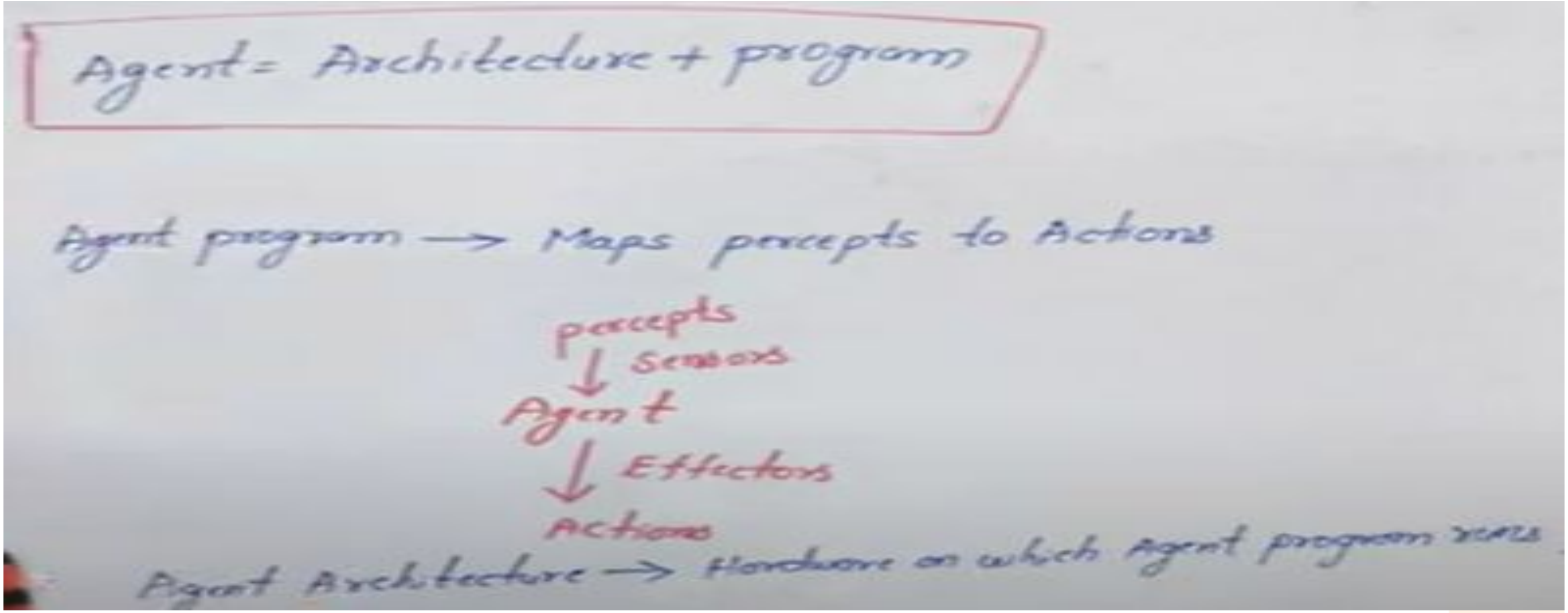


i **Percepts**: location and contents, e.g., [*A, Dirty*]

i **Actions**: *Left, Right, Suck, NoOp*



Percept sequence	Action
[*A*, *Clean*]	*Right*
[*A*, *Dirty*]	*Suck*
[*B*, *Clean*]	*Left*
[*B*, *Dirty*]	*Suck*
[*A*, *Clean*], [*A*, *Clean*]	*Right*
[*A*, *Clean*], [*A*, *Dirty*]	*Suck*
⋮	⋮



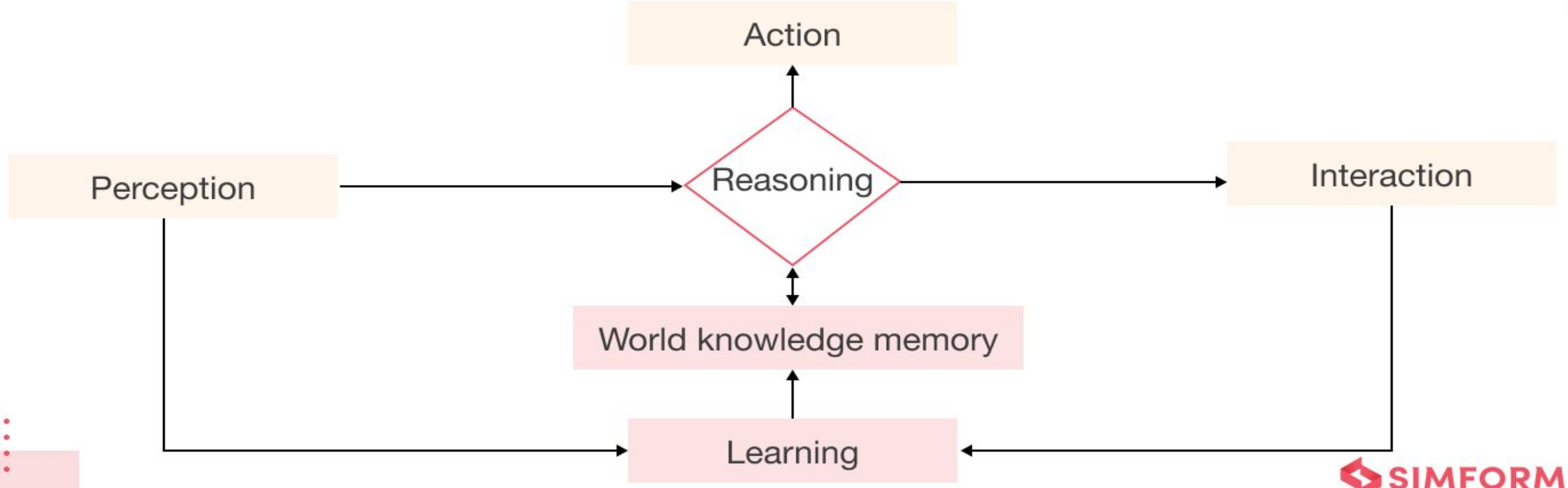
Agent = Architecture + program

Agent program → Maps percepts to Actions

percepts
↓ Sensors
Agent
↓ Effectors
Actions

Agent Architecture → Hardware on which Agent program runs

Structure of an Intelligent agent



SIMFORM

1. Environment

The environment refers to the area or domain in which an AI agent operates. It can be a physical space, like a factory floor, or a digital space, like a website.

2. Sensors

Sensors are the tools that an AI agent uses to perceive its environment. These can be cameras, microphones, or any other sensory input that the AI agent can use to understand what is happening around it.

3. Actuators

Actuators are the tools that an AI agent uses to interact with its environment. These can be things like robotic arms, computer screens, or any other device the AI agent can use to change the environment.

4. Decision-making mechanism

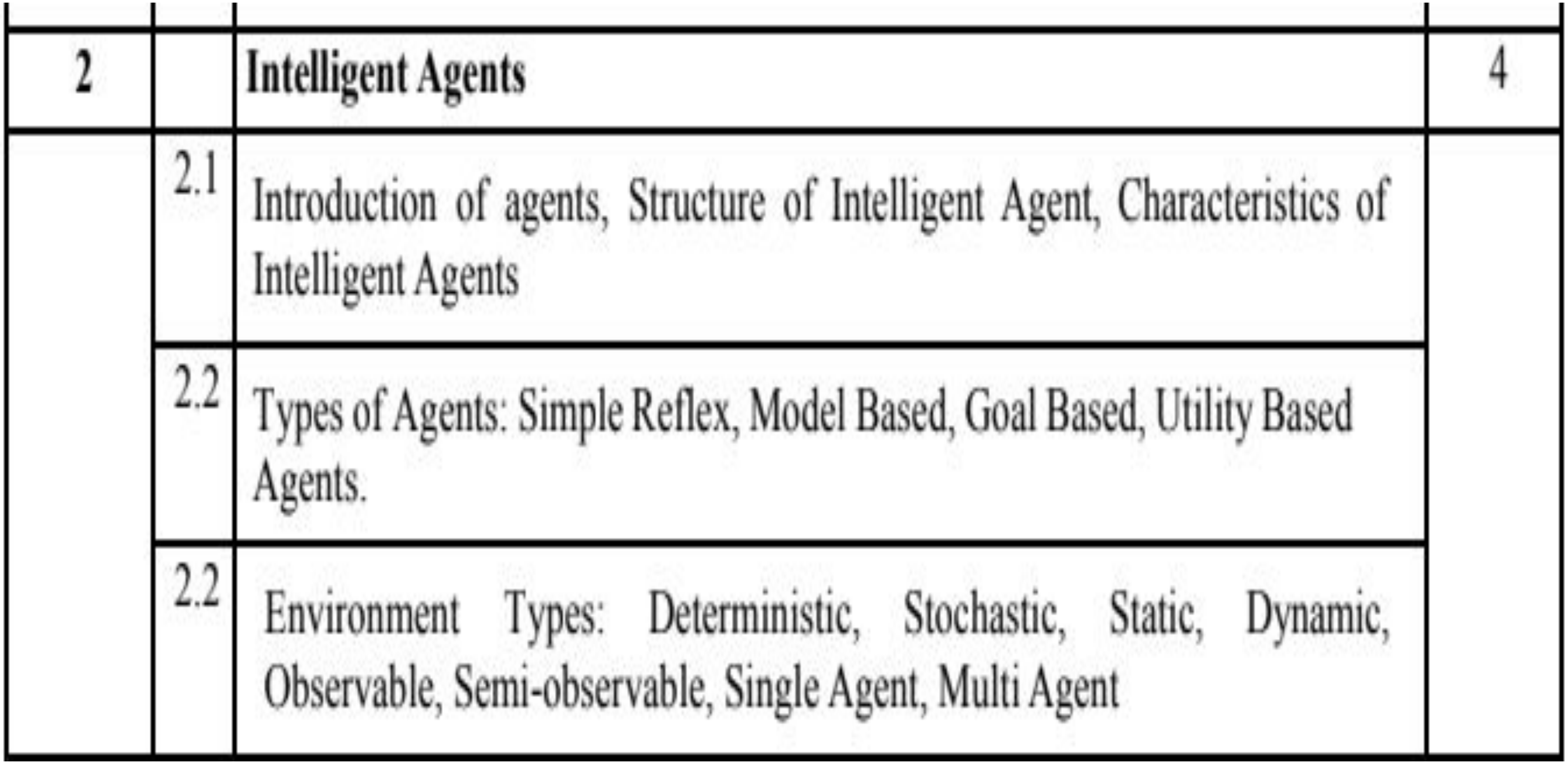
It is the brain of an AI agent. It processes the information gathered by the sensors and decides what action to take using the actuators.

AI agents use various decision-making mechanisms, such as rule-based systems, expert systems, and neural networks, to make informed choices and perform tasks effectively.

5. Learning system

It enables the AI agent to learn from its experiences and interactions with the environment.

It uses techniques like reinforcement learning, supervised learning, and unsupervised learning to improve the performance of the AI agent over time.



2		**Intelligent Agents**	4
	2.1	Introduction of agents, Structure of Intelligent Agent, Characteristics of Intelligent Agents	
	2.2	Types of Agents: Simple Reflex, Model Based, Goal Based, Utility Based Agents.	
	2.2	Environment Types: Deterministic, Stochastic, Static, Dynamic, Observable, Semi-observable, Single Agent, Multi Agent	

Characteristics of Intelligent Agents

1. **Situatedness**

When an Agent receives some form of sensory input from its environment, it then performs some actions that change its environment in some way.

1. **Autonomy**

An AI virtual agent is capable of performing tasks independently without requiring constant human intervention or input.

Notes

Characteristics of Intelligent Agents

3. Perception:

The agent function senses and interprets the environment they operate in through various sensors, such as cameras or microphones.

4. Reactivity:

An AI agent can assess the environment and respond accordingly to achieve its goals.

5.Reasoning and decision-making:

AI agents are intelligent tools that can analyze data and make decisions to achieve goals. They use reasoning techniques and algorithms to process information and take appropriate actions.

Characteristics of Intelligent Agents

6. Learning:

They can learn and enhance their performance through machine, deep, and reinforcement learning elements and techniques.

7.Communication:

AI agents can communicate with other agents or humans using different methods, like understanding and responding to natural language, recognizing speech, and exchanging messages through text.

8. Goal-oriented:

They are designed to achieve specific goals, which can be pre-defined or learned through interactions with the environment.

Advantages of using AI agents

- Increased efficiency
- Learning agents
- Increased speed
- Better decision-making
- Personalization
- 24*7 availability

SIMFORM

1. Increased efficiency

AI agents can automate repetitive tasks, allowing businesses to complete them faster and more accurately. This efficiency improvement frees employees’ time to focus on more business-critical tasks and improves productivity.

2. Better decision-making

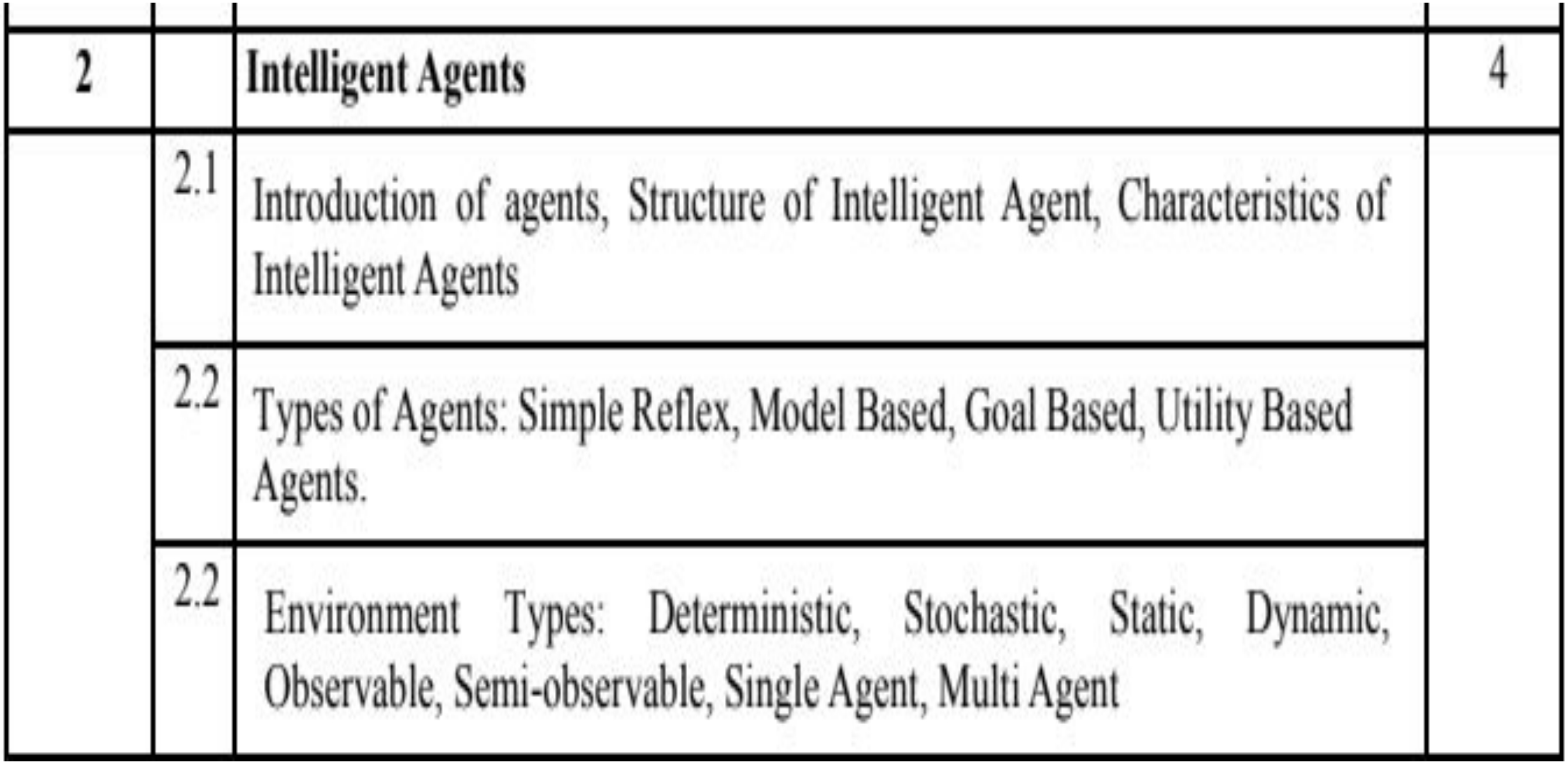
AI agents can analyze large amounts of data and provide valuable insights to support decision-making processes. By leveraging advanced algorithms and machine learning, AI agents can identify patterns, trends, and correlations that humans may overlook.

3. Improved customer experience

AI agents can provide personalized and timely interactions with customers, enhancing their experience. They can offer instant support, answer queries, and provide recommendations, leading to increased customer satisfaction and loyalty.

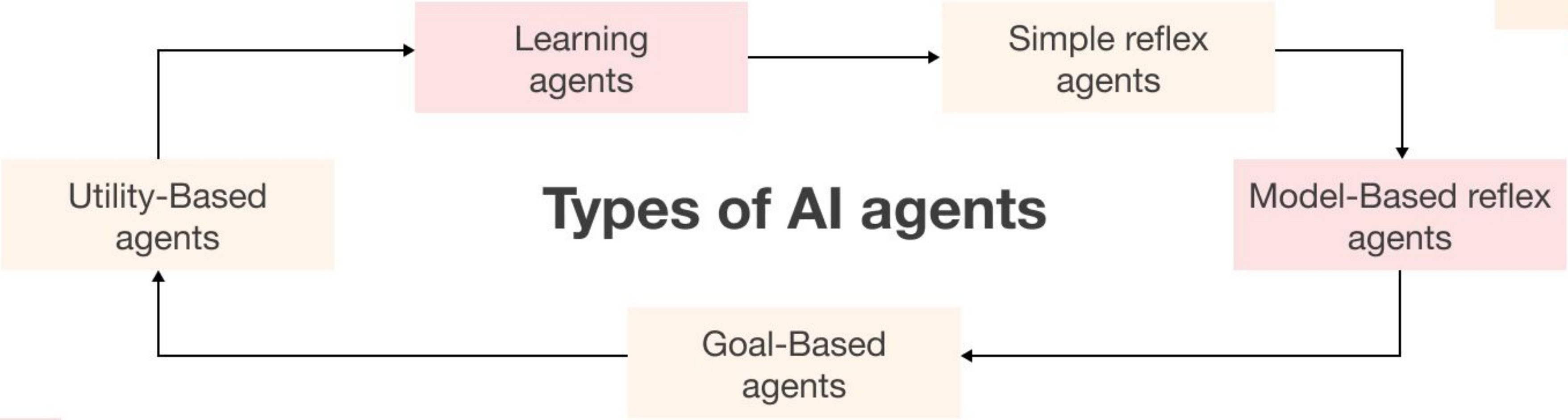
4. Cost savings

By automating tasks, AI agents can reduce the need for human resources and manual labor, resulting in cost savings for businesses. They can handle high-volume, repetitive tasks without fatigue or errors



2		**Intelligent Agents**	4
	2.1	Introduction of agents, Structure of Intelligent Agent, Characteristics of Intelligent Agents	
	2.2	Types of Agents: Simple Reflex, Model Based, Goal Based, Utility Based Agents.	
	2.2	Environment Types: Deterministic, Stochastic, Static, Dynamic, Observable, Semi-observable, Single Agent, Multi Agent	

All Agent Notes



Types of AI agents

Learning agents

Simple reflex agents

Model-Based reflex agents

Goal-Based agents

Utility-Based agents

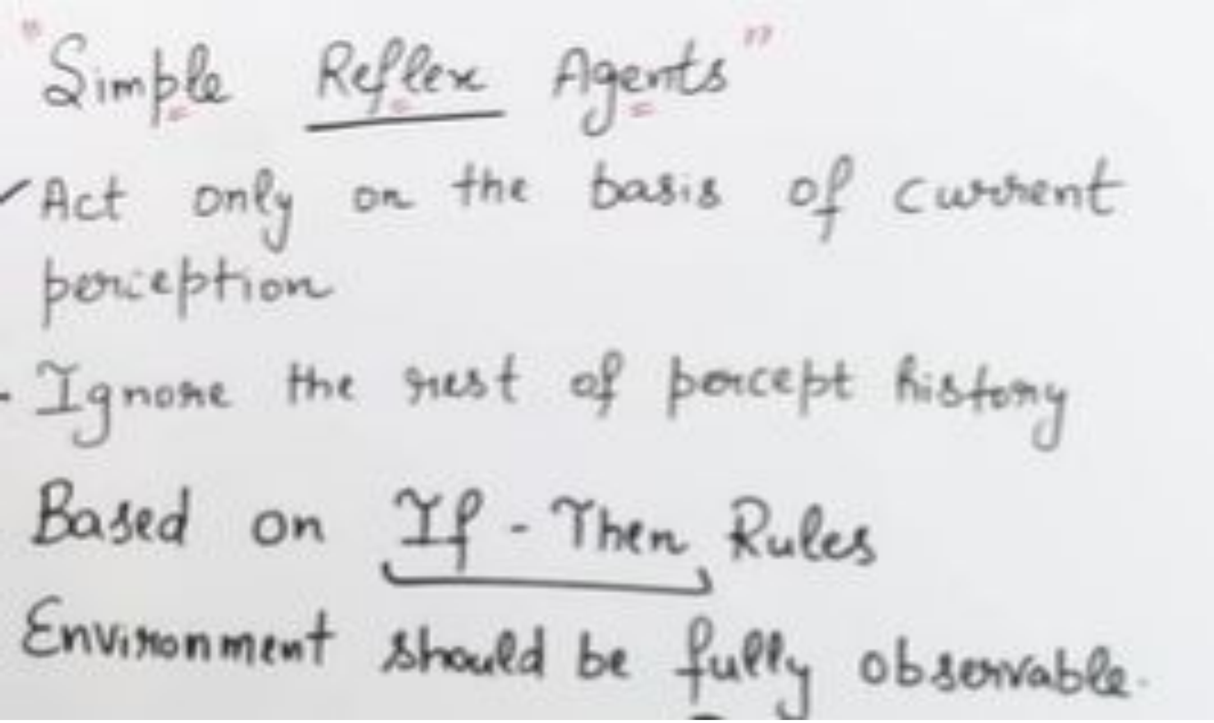
SIMFORM

Agent types/Program

Four basic types in order of increasing generality:

1) Simple reflex agents
2) Model-based reflex agents
3) Goal-based agents
4) Utility-based agents
5) Learning agent

1) **Simple reflex** agents are programmed to respond to specific environmental stimuli based on predefined rules.
2) **Model-based reflex** agents are reactive agents that maintain an internal model of the environment and use it to make decisions.
3) **Goal-based agents** execute a program to achieve specific goals and take actions based on evaluating the current state of the environment.
4) **Utility-based** agents consider the potential outcomes of their actions and choose the one that maximizes the expected utility.
5) **Learning** agents execute machine learning techniques to improve their decision-making over time.

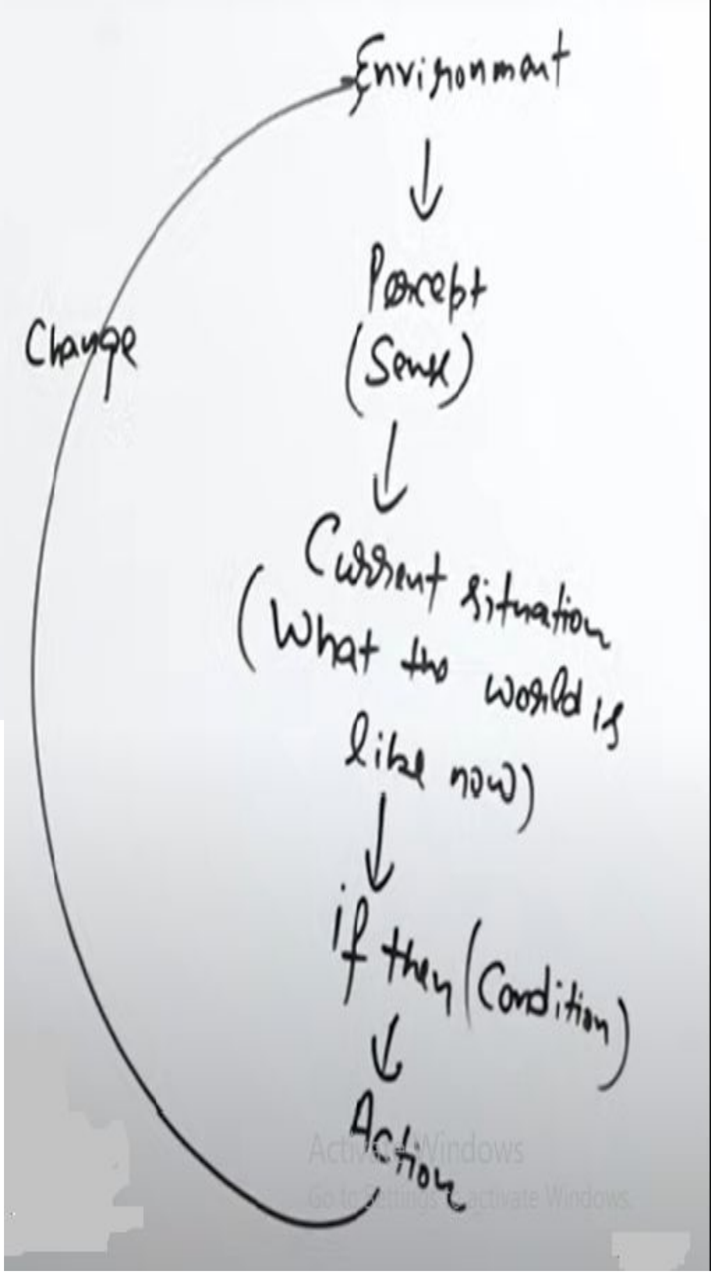


"Simple Reflex Agents"

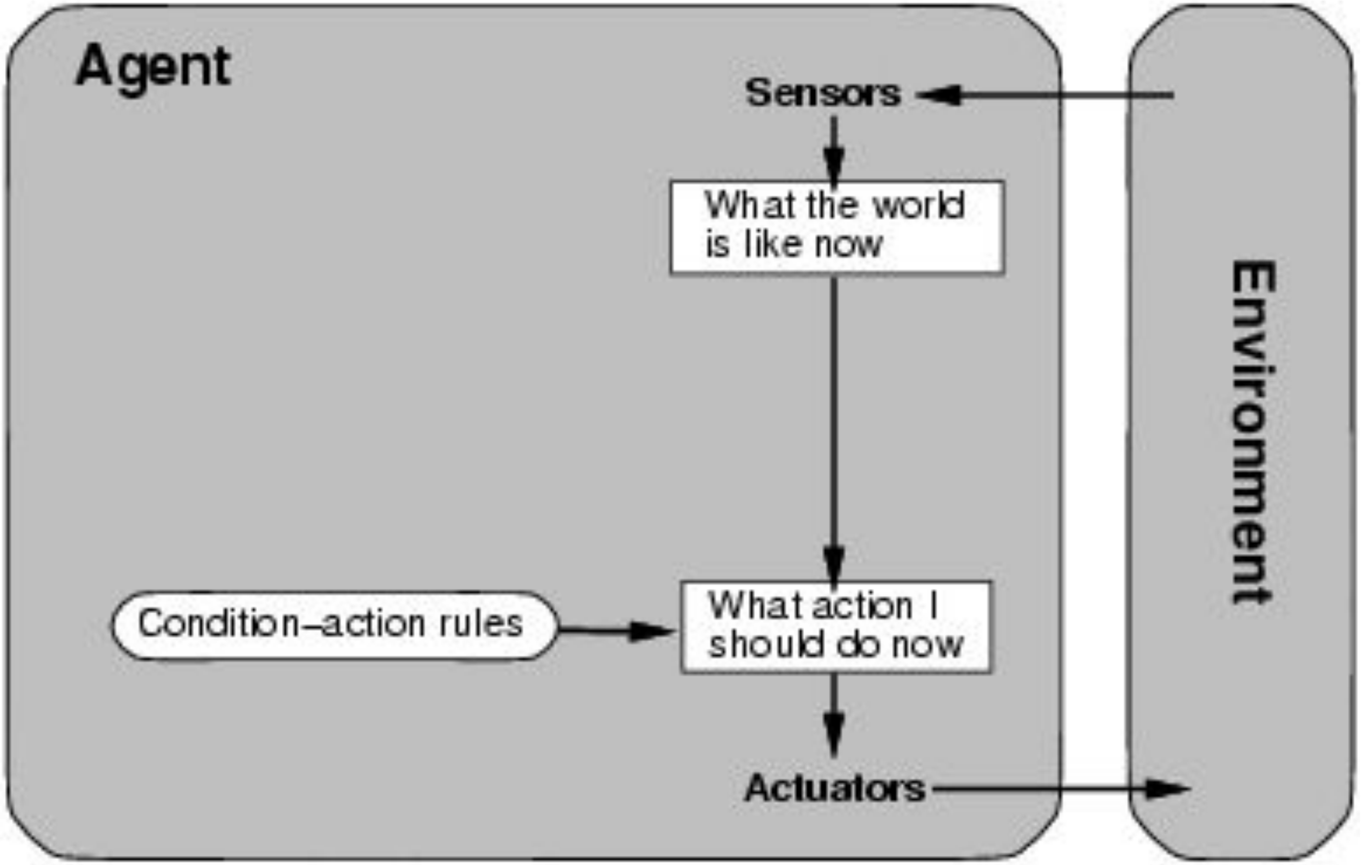
- Act only on the basis of current perception
- Ignore the rest of percept history

Based on If - Then Rules

Environment should be fully observable.



Simple reflex agents



We use rectangle to denote the current internal state of the agents decision process and ovals to represent the background information used in the process

Simple Reflex Agent

- **function** SiMPLE-REFLEX-AGENT(Percept)**returns** *action*
- **static:** *rules,* a set of condition-action rules
- *state* <— INTERPRET-INPUT(percept)
- rule <- RULE-MATCH(state, *rules)*
- *action* <- RULE-ACTiON[rule]
- **return** *action*

It acts according to a rule whose condition matches the current state as defined by percept.

Simple Reflex Agent

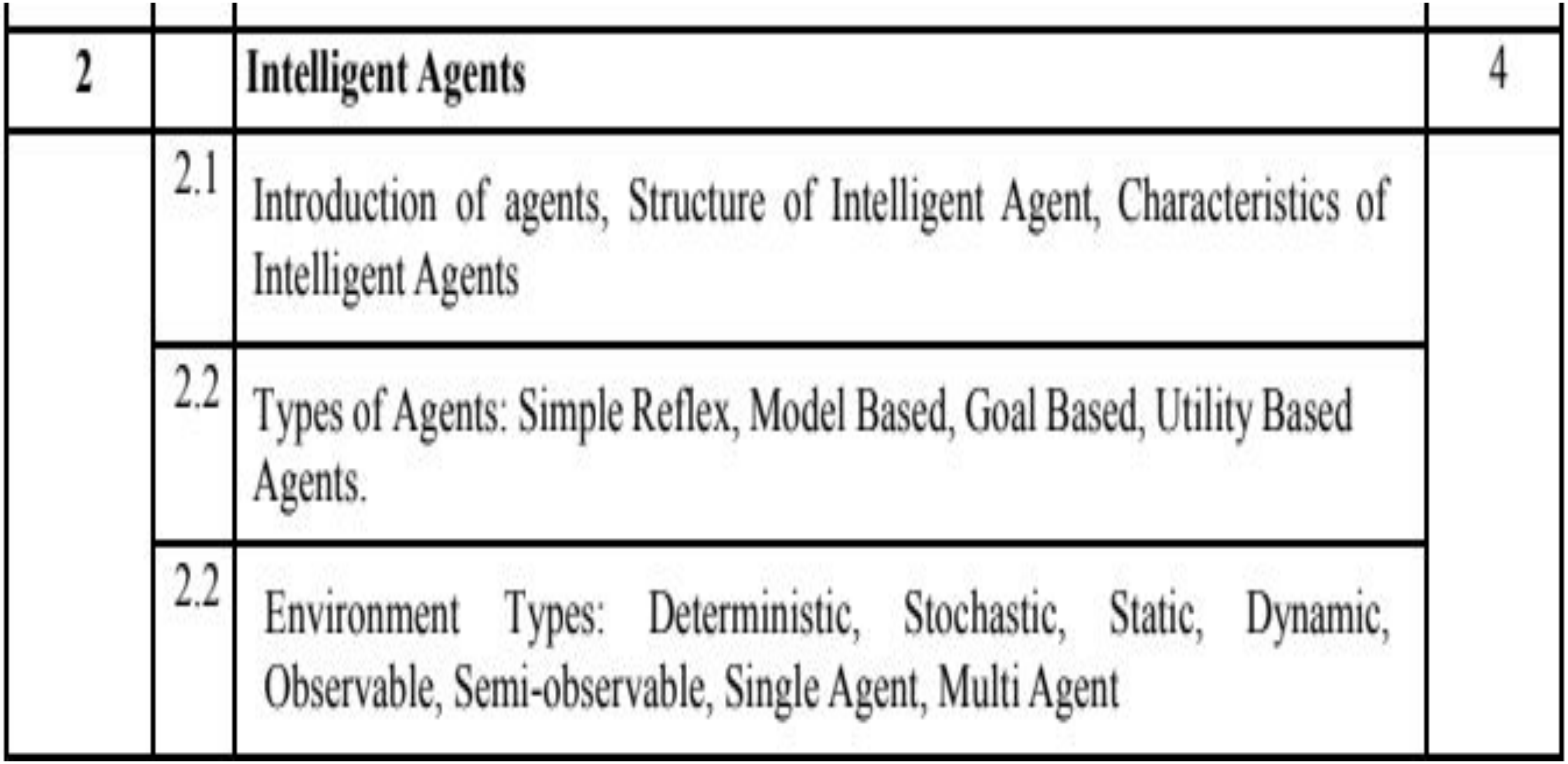
The simplest Kind of agent, agent select actions on the basis of the current percept, ignoring the rest of the percept history.

- Vacuum Cleaner – based on current location and on whether that contains dirt
- Car driving: if the car in front brakes, and its brake lights come on, then the driver should notice this and initiate braking. **condition-action *rule*** ***if*** *car-in-front-is-braking* **then** *initiate-braking.*

Issues

- *If decision can be made on the basis of current percept i.e only If the environment is fully observable.*
- sensors do not provide access to the complete state of the world. In such cases, the agent may need to maintain some internal state information
- *Infinite Loop: due to partially observable environment.*

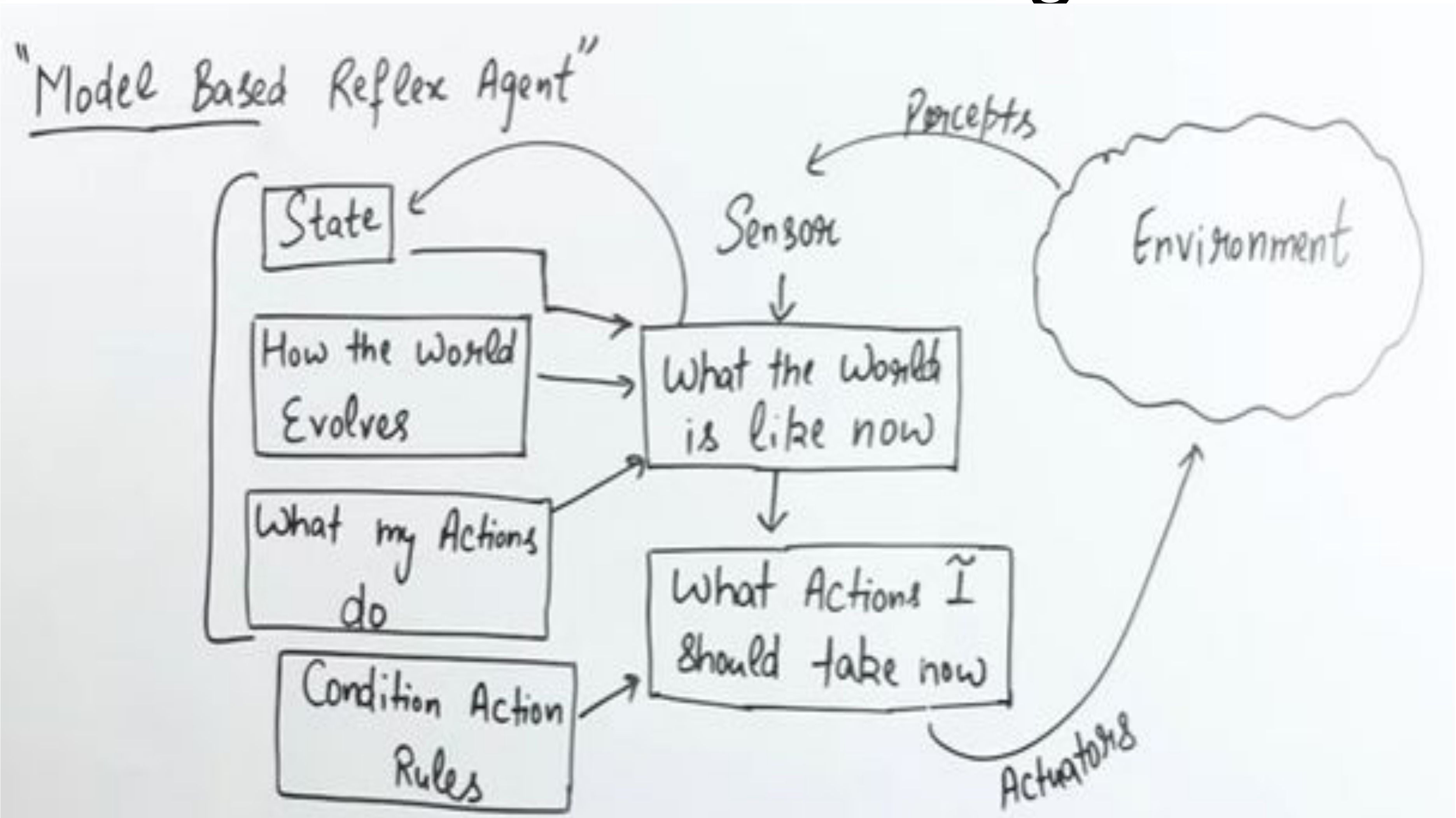
```
function
REFLEX_VACUUM_AGENT([LOCATION,STSTUS])
returns an action

if STATUS = Dirty then return Suck
else if location = A then return Right
else if location = B then return Left
```

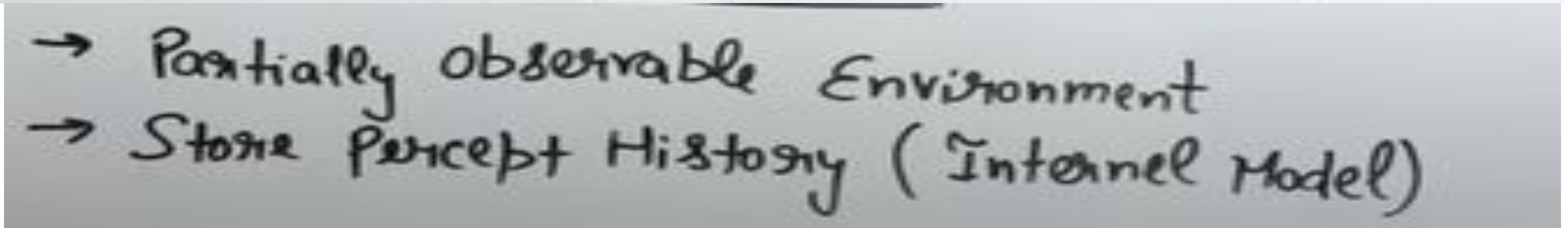


2		**Intelligent Agents**	4
	2.1	Introduction of agents, Structure of Intelligent Agent, Characteristics of Intelligent Agents	
	2.2	Types of Agents: Simple Reflex, Model Based, Goal Based, Utility Based Agents.	
	2.2	Environment Types: Deterministic, Stochastic, Static, Dynamic, Observable, Semi-observable, Single Agent, Multi Agent	

Waymo Project

Model based reflex agents





→ Partially observable Environment

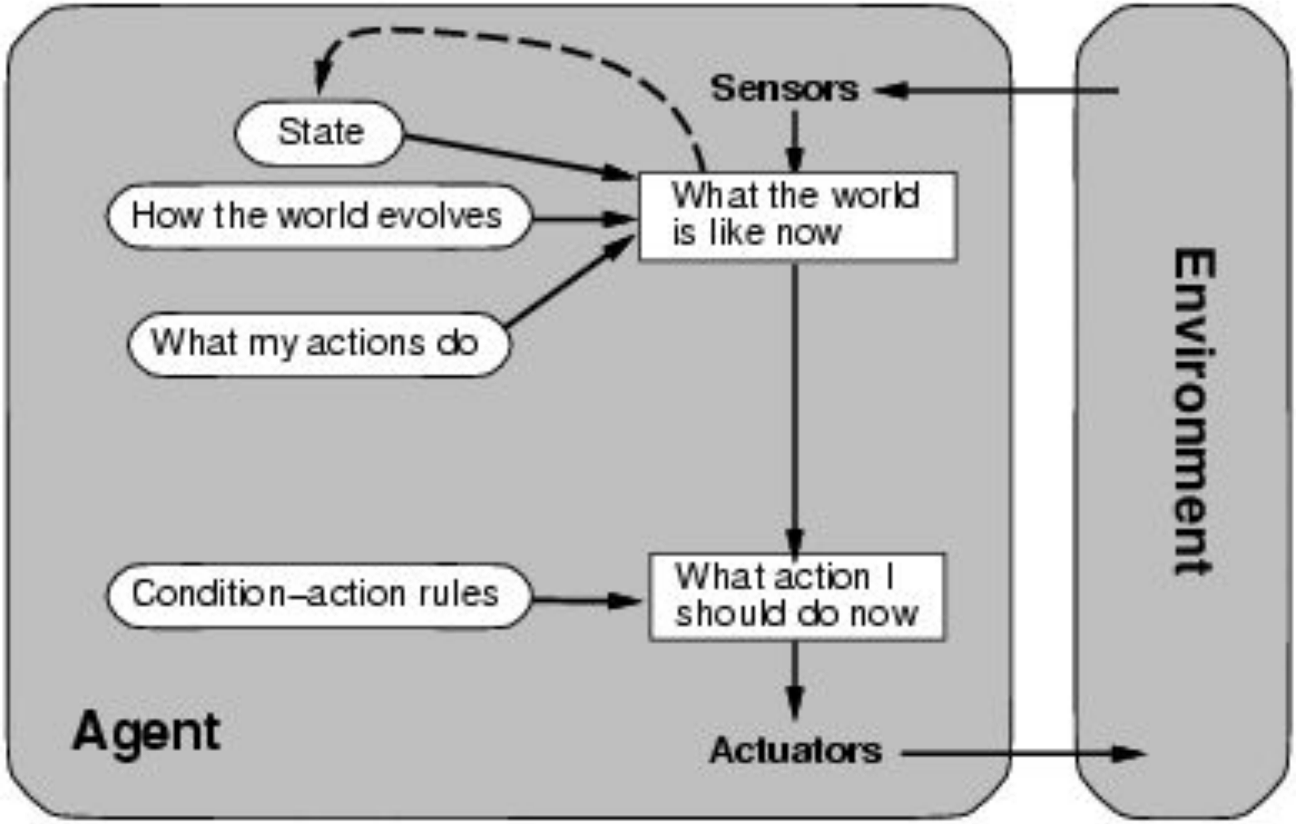
→ Store Percept History (Internel Model)

Model based reflex agents

- Agent to keep track of the part of the word it cant see now.
- Maintain some sort of internal state that depend on the percept history and thereby reflect at least some of unobservable aspect of current state.
- Updating this **internal state information** as time goes by requires two kinds of knowledge to be encoded in the agent program.

1. we need some information about how the world evolves independently of the agent—for example, that an **overtaking car** generally will be closer behind than it was a moment ago.
2. we need some information about' how the agent's own actions affect the world—for example, that when the agent changes lanes to the right, **there is a gap (at least temporarily)** in the lane it was in before, or that after driving for five minutes northbound ; on the freeway one is usually about five miles north of where one was five minutes ago.

Model-based reflex agents



Current percept is combined with old internal state to generate the updated description of the current state.

Model-based reflex agents

- **Function Reflex agent with state returns an** *action*
- **static:**
- *state* <— Update-state(state,action,percept)
- rule <- RULE-MATCH(state, *rules)*
- *action* <- RULE-ACTiON[rule]
- **return** *action*
- Update state is responsible for creating the new internal state description as well as interpreting the new percept in the light of existing knowledge about the state.

Action may depend on history or unperceived aspects of the world.

- Need to maintain internal world model.

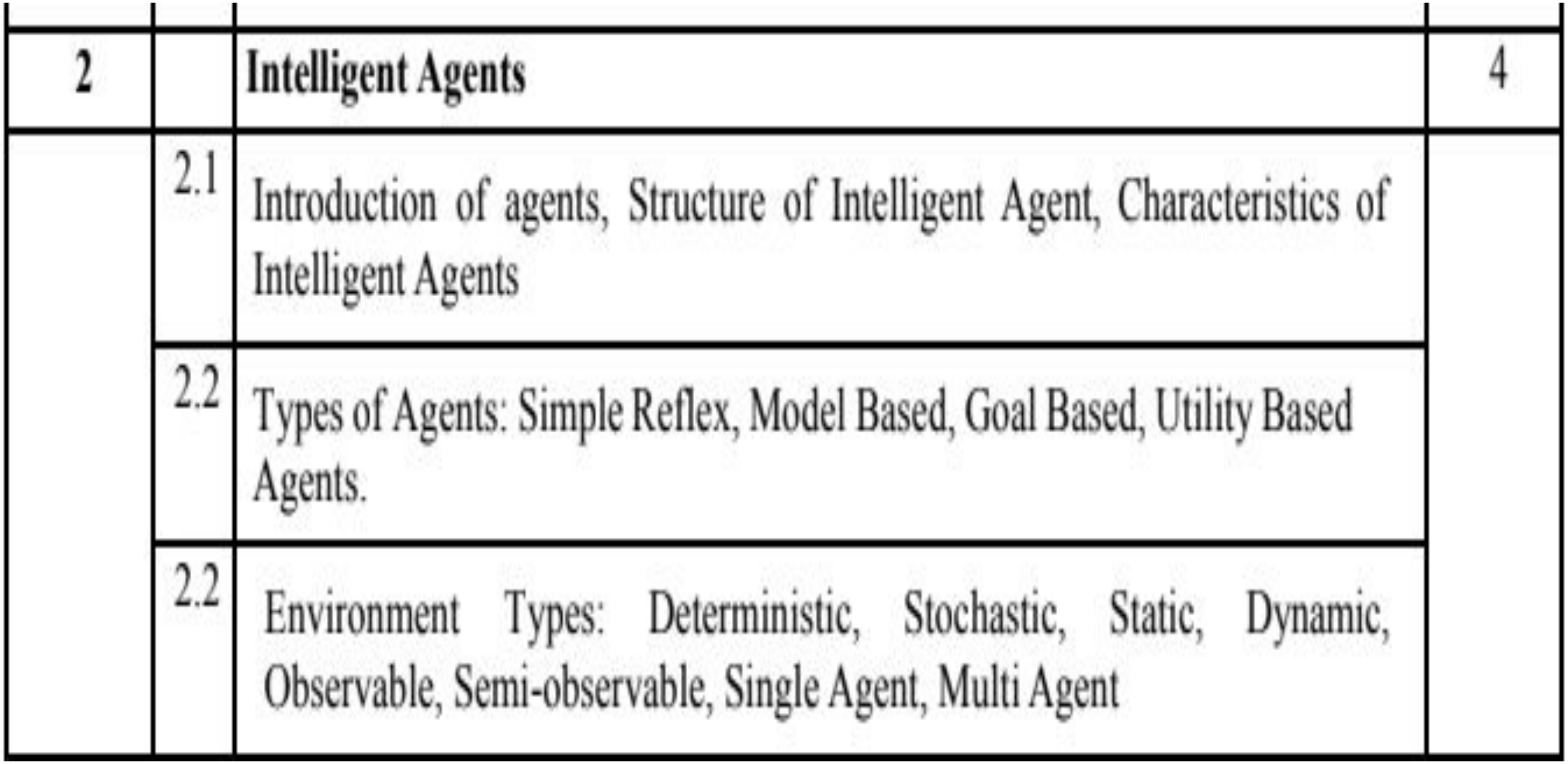
Example:

Agent: robot vacuum cleaner

Environment: dirty room, furniture.

Model: map of room, which areas already cleaned.

Sensor/model trade-off.

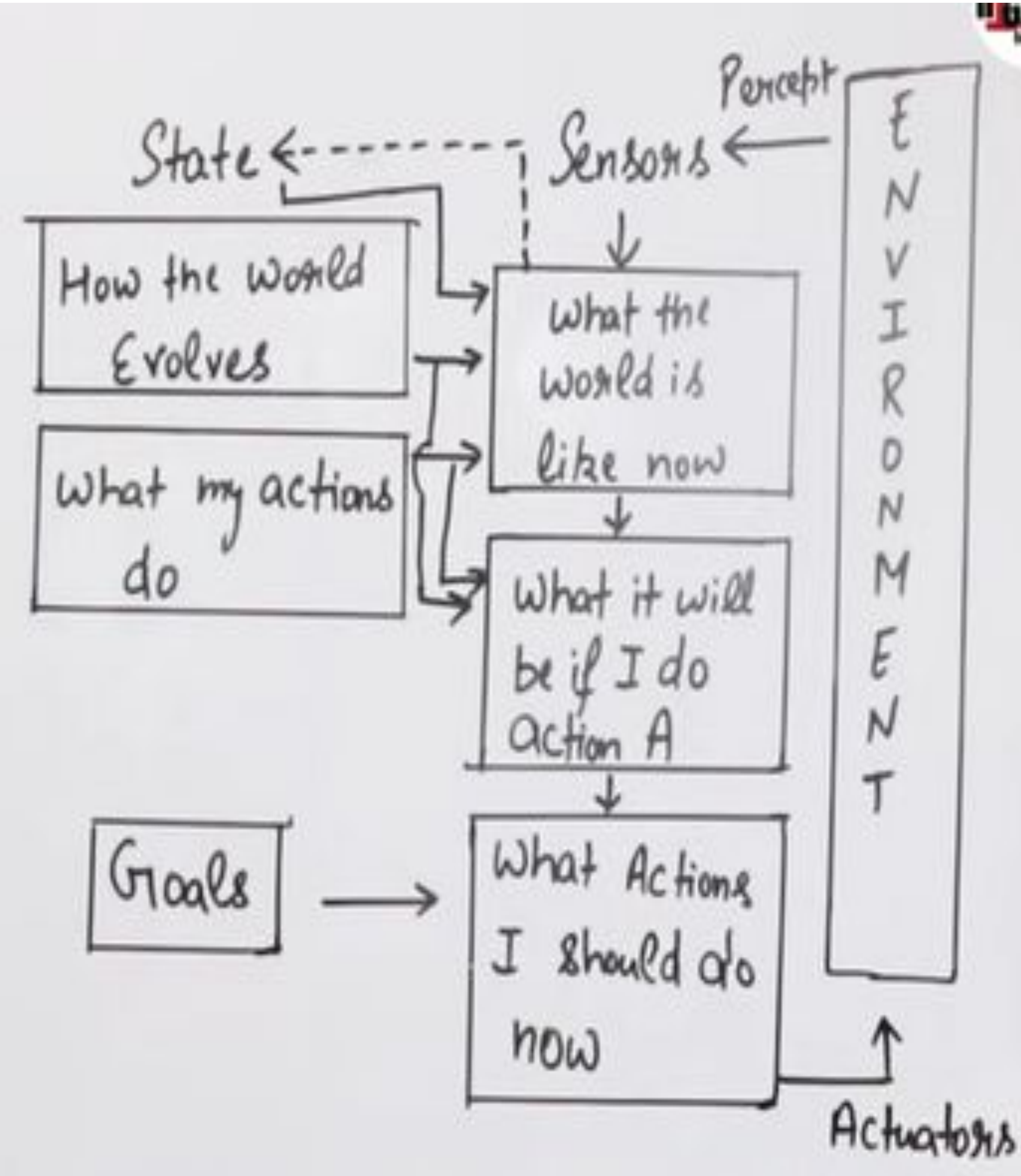


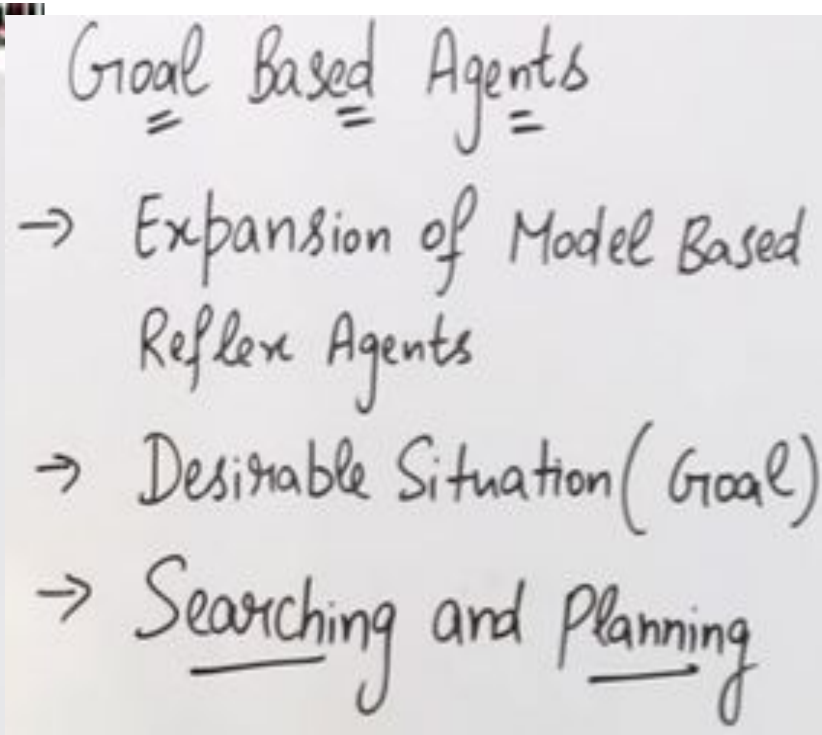
2		**Intelligent Agents**	4
	2.1	Introduction of agents, Structure of Intelligent Agent, Characteristics of Intelligent Agents	
	2.2	Types of Agents: Simple Reflex, Model Based, Goal Based, Utility Based Agents.	
	2.2	Environment Types: Deterministic, Stochastic, Static, Dynamic, Observable, Semi-observable, Single Agent, Multi Agent	

Alibaba (G Plus)

https://www.youtube.com/watch?v=9VFcJ8Qjhmk

Goal based reflex agents


Percept
State
Sensors
How the world Evolves
What my actions do
What the world is like now
What it will be if I do action A
Goals
What Actions I should do now
ENVIRONMENT
Actuators


Goal Based Agents

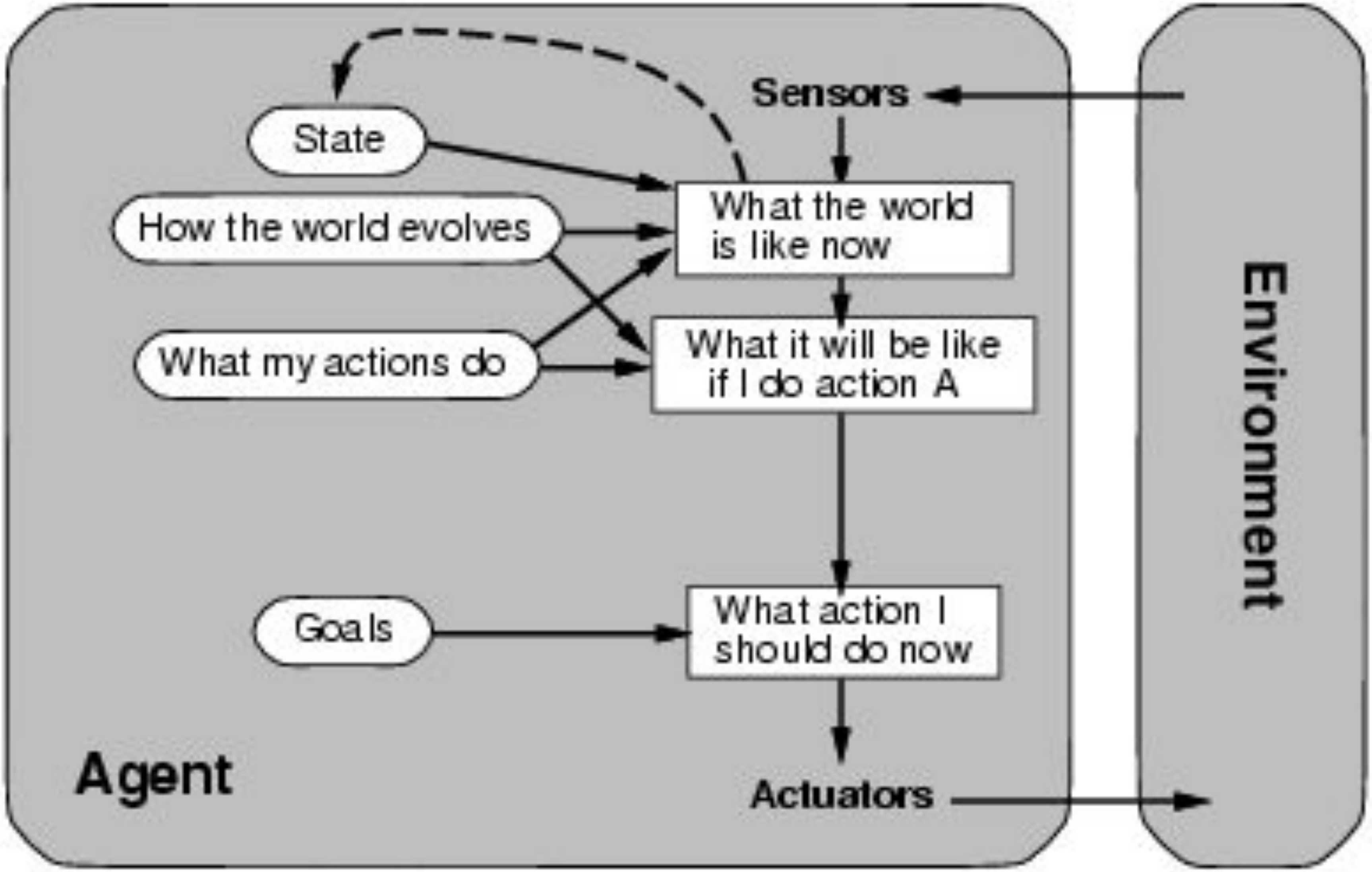
- → Expansion of Model Based Reflex Agents
- → Desirable Situation (Goal)
- → Searching and Planning

Alibaba (G Plus)
https://www.youtube.com/watch?v=9VFcJ8Qjhmk

Goal-based agents

- **Knowing about the current state** of the environment is not always enough to decide what to do. For example, at a road junction, the taxi can turn left, right, or go straight on. The right decision depends on where the taxi is trying to get to.
- Agent needs some sort of **goal** information, which **describes situations** that are desirable for example, being at the passenger's destination.
- Agent Program choose the action that achieve the goal
- Searching and Planning are the subfields of AI devoted to finding action sequence that **achieve the agents' goal.**
- Decision making is different from condition-action rule
- Reflex agent use built in rule to map percept to action , Goal based could reason ,although the goal based agent appeared less efficient it is more flexible because knowledge that support its decision is represented explicitly and can be modified.

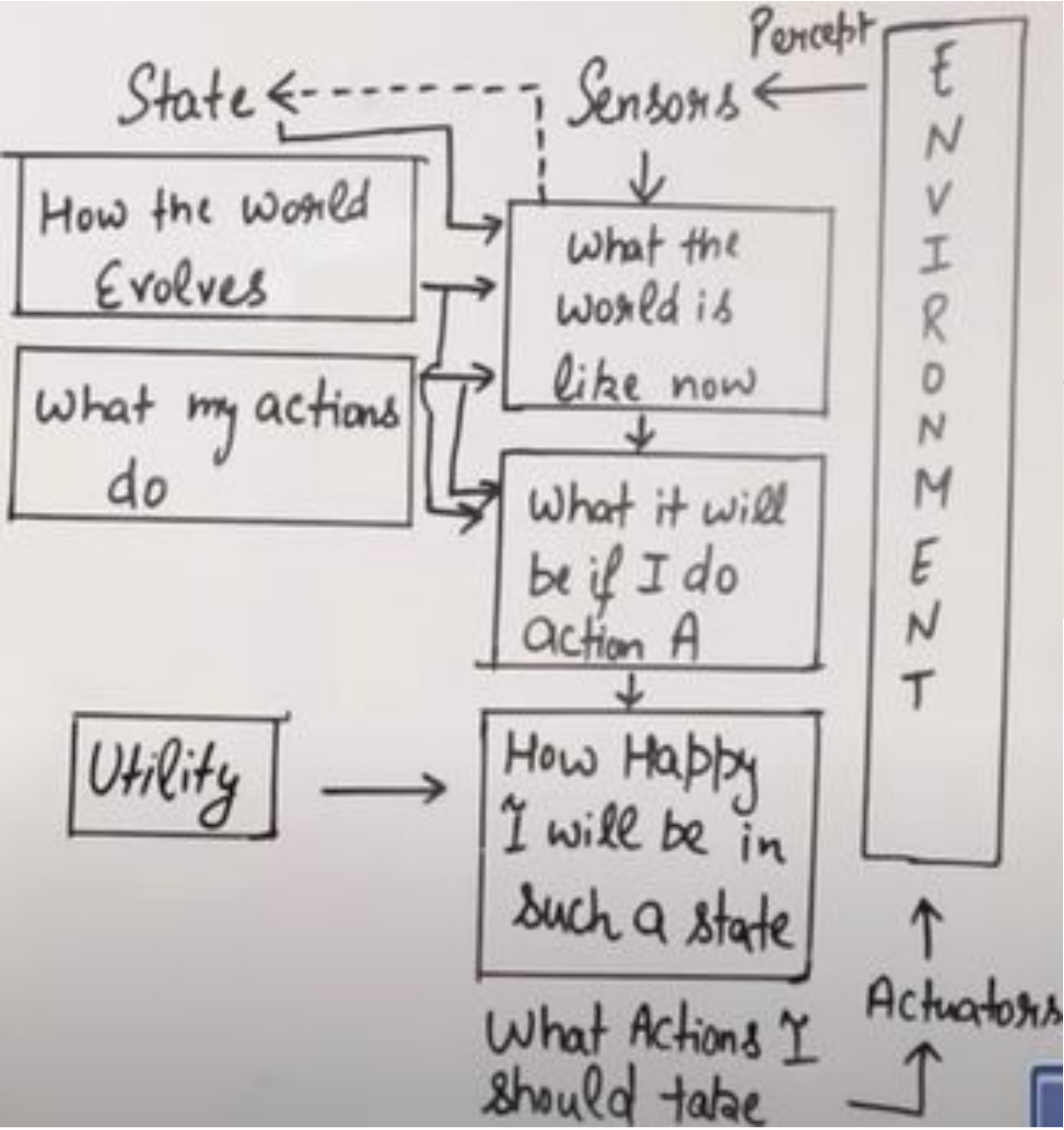
Goal-based agents

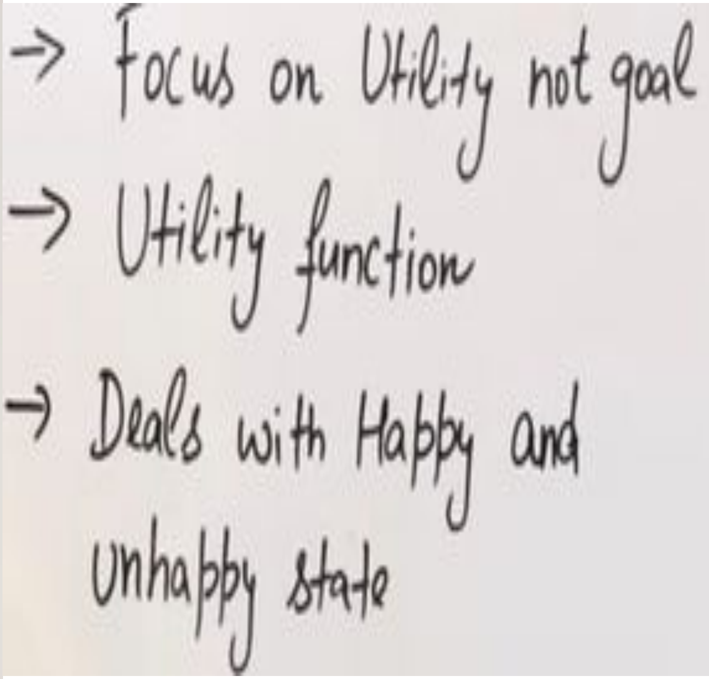

Sensors
State
How the world evolves
What my actions do
What the world is like now
What it will be like if I do action A
Goals
What action I should do now
Agent
Actuators
Environment

goal-based agent

```
def action (agent, environment):
    let state = agent. sense (environment)
    for action in agent.available Actions :
        let resultingState = agent.do (action, state)
        if resultingState == agent. goal:
            return action
```

Utility-based agents


Percept
State
Sensors
ENVIRONMENT
How the world Evolves
What the world is like now
What my actions do
What it will be if I do action A
Utility
How Happy I will be in such a state
What Actions I should take
Actuators



- → Focus on Utility not goal
- → Utility function
- → Deals with Happy and unhappy state

Utility-based agents

- These agents are similar to the goal-based agent but provide an **extra component of utility measurement** which makes them different by providing a measure of success at a given state.
- Utility-based agent act based **not only goals but also the best way to achieve the goal.**
- The Utility-based agent is useful when there are **multiple possible alternatives,** and an agent has to choose in order to perform the **best action**.
- The utility function maps each state to a real number to check how efficiently each action achieves the goals.

Utility-based agents

- A utility-based agent is an agent that acts based not only on what the goal is, but the **best way to reach that goal.**
- When there are multiple possible alternatives, then to decide **which one is best.**
- They choose actions based on a preference (utility) for each state.
- Agent happiness should be taken into consideration. Utility describes how **"happy"** the agent is. Because of the uncertainty in the world, a utility agent chooses the action that maximizes the expected utility.

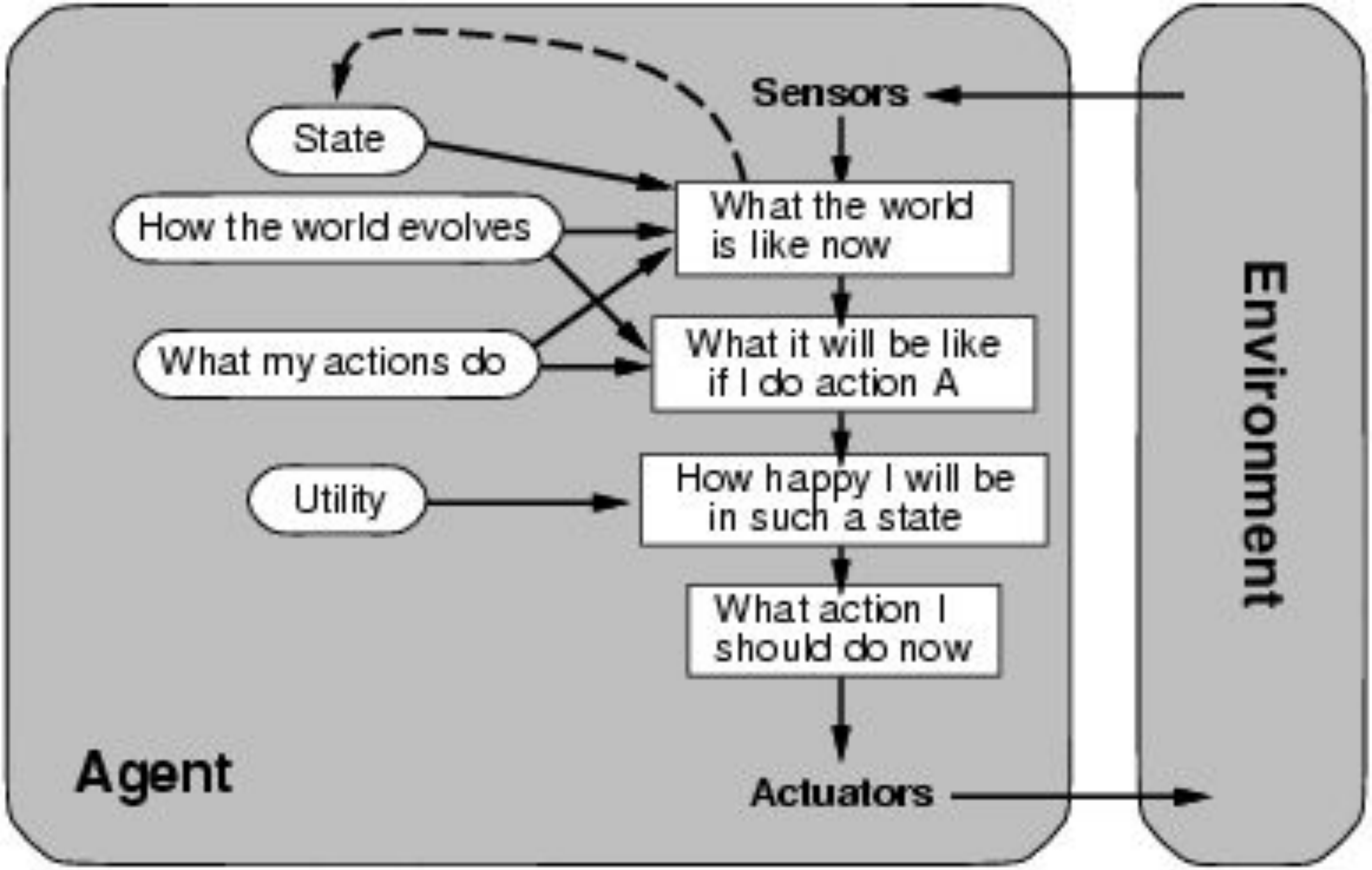
- *action sequences that will get the taxi to its destination, thereby achieving the goal, but some are **quicker, safer, more reliable, or cheaper than others.***

Utility-based agents

- Goals alone are not really enough to generate high-quality behavior.
- Utility is therefore a function that maps a state onto a real number.
- A complete specification of the utility function allows rational decisions in two kinds of cases where goals have trouble:

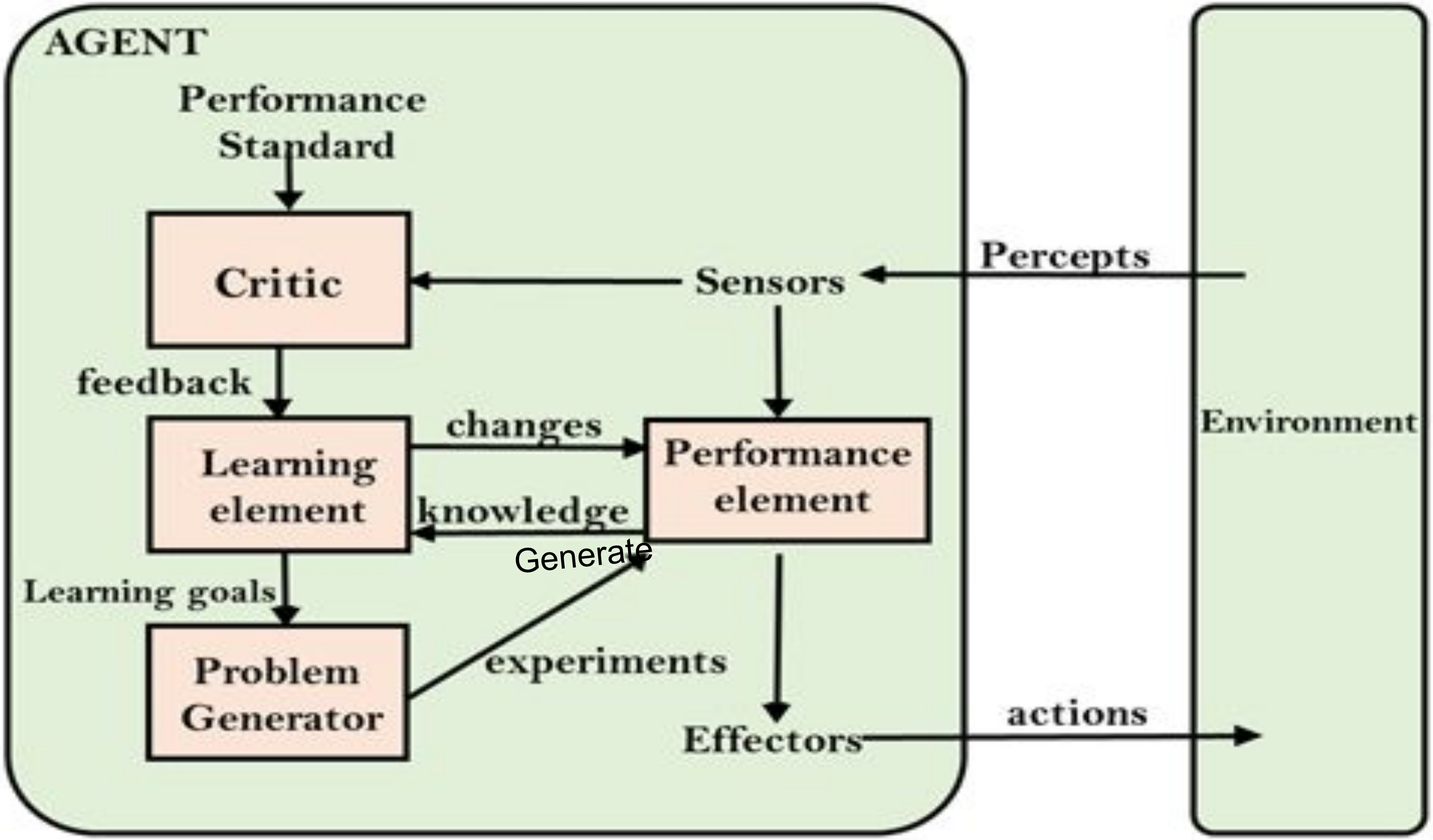
1. when there are conflicting goals only some of which can be achieved (for example, **speed and safety**), the utility function specifies the appropriate trade-off.
2. when there are several goals that the agent can aim for, none of which can *be* achieved with certainty, utility provides a way in which the likelihood of success can be weighed up against the importance of the goals.

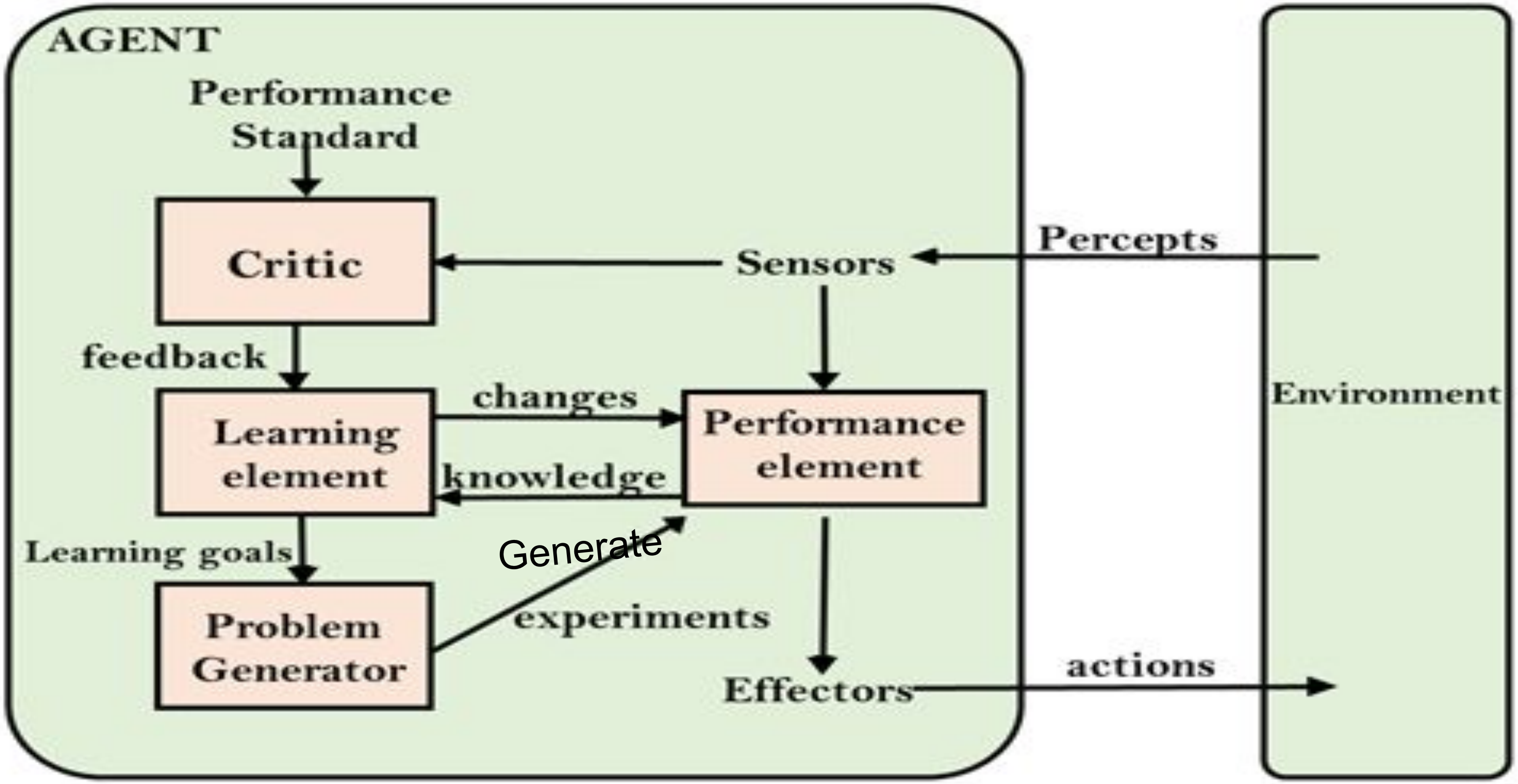
Utility-based agents


Sensors
State
How the world evolves
What my actions do
Utility
What the world is like now
What it will be like if I do action A
How happy I will be in such a state
What action I should do now
Agent
Actuators
Environment

```
def action ( agent , environment ) :
    let state = agent . sense ( environment )
    let maxUtility = −Infinity
    let maxUtility State = None
    for action in agent.available Actions :
        let resulting State = agent. do (action, state)
        let resulting StateUtility = agent. getUtility ( resulting State )
        if resulting StateUtility > maxUtility :
            maxUtility = resulting StateUtility
            maxUtility State = resulting State
    return maxUtility State
```

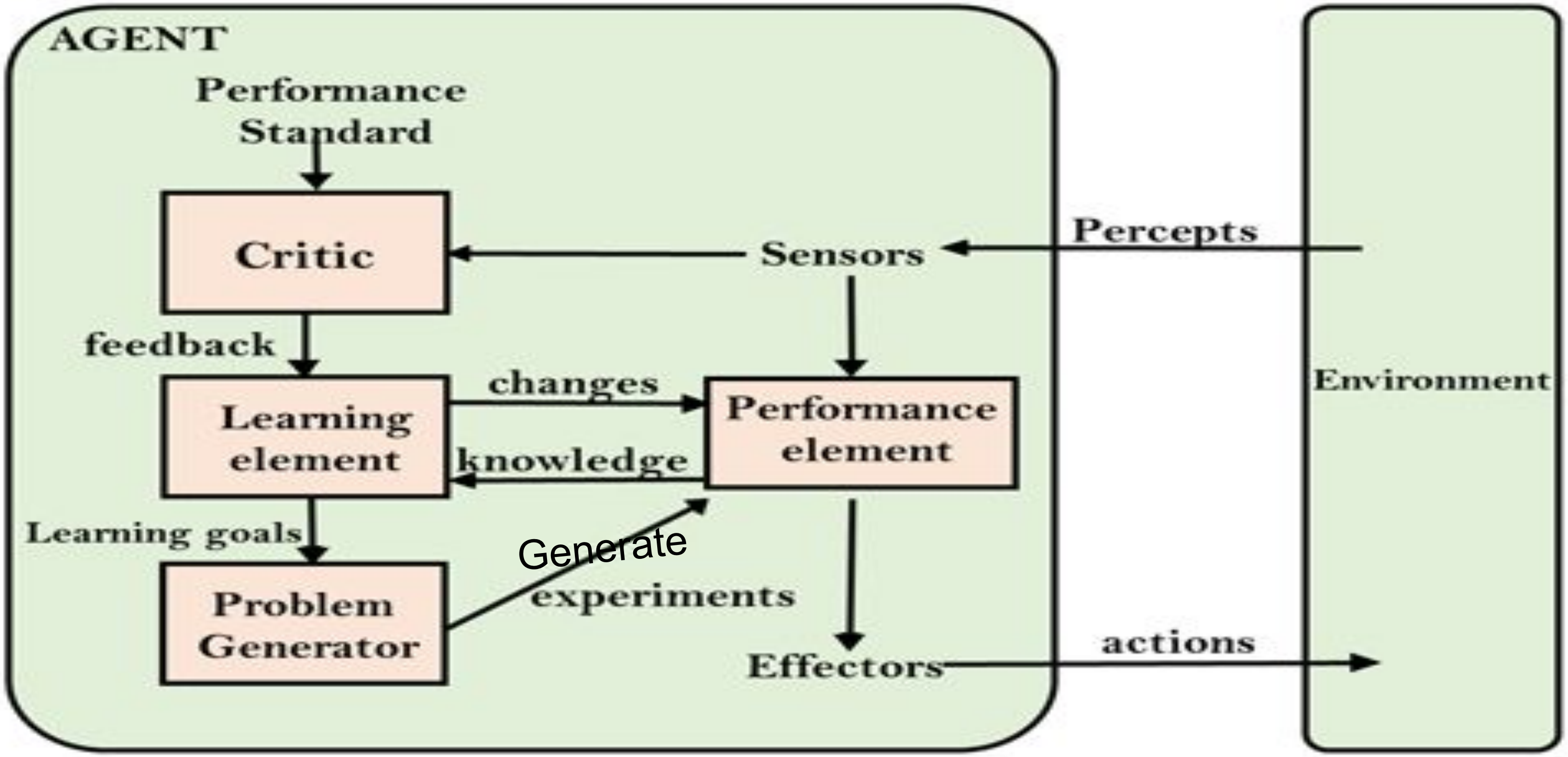
Learning agents





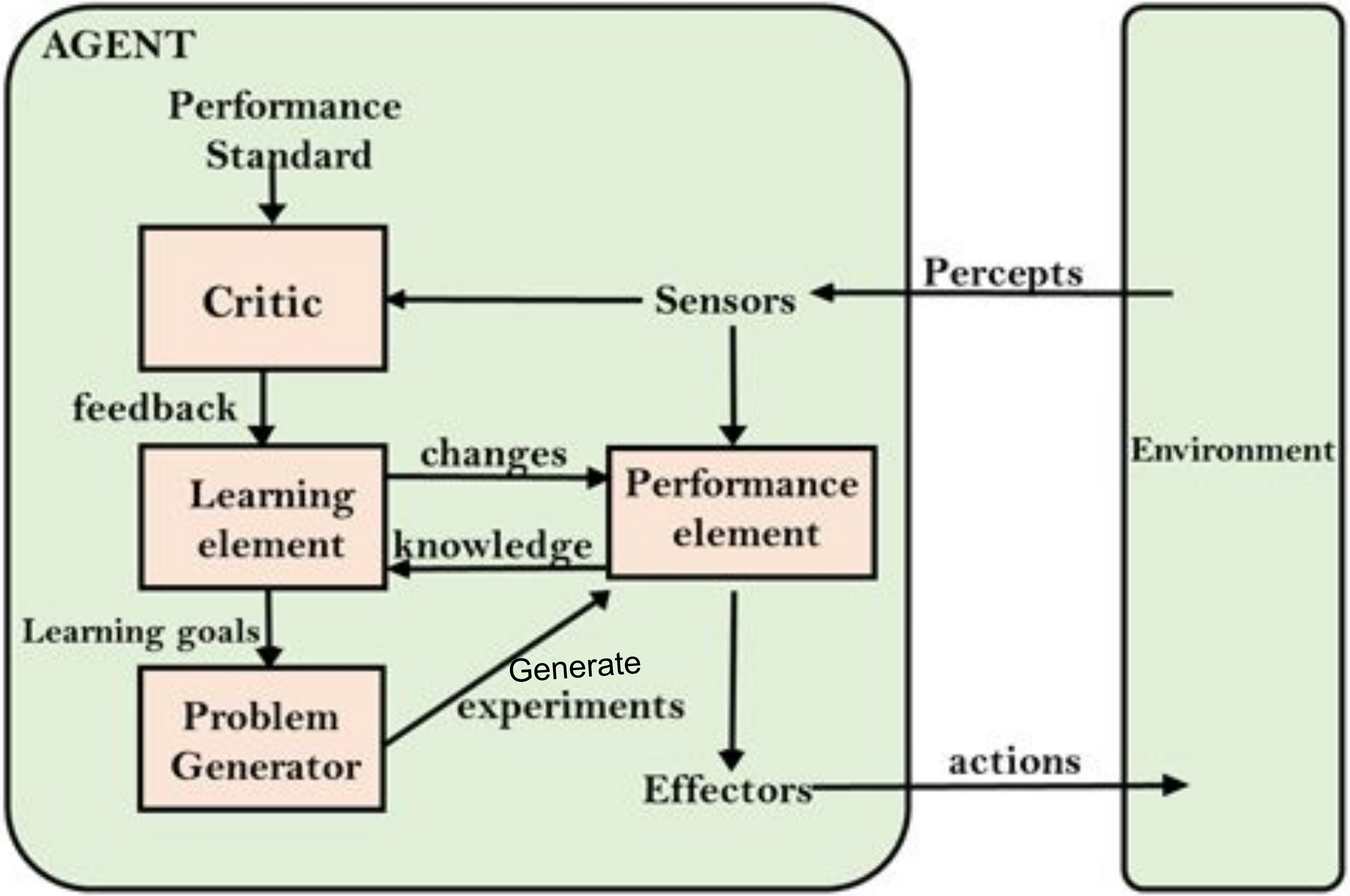
- All agents can improve their performance through learning

1. Learning element - which is responsible for making improvement **(When to do what)**
2. Performance element - selecting external actions: it takes percepts and decide on actions **(How to do everything)**



1. Critic how agent is doing and how the performance should be modified to do better.
2. Problem generator suggest exploratory actions that lead to new and informative experiences.

- Critic tells the learning element how well the agent is doing w.r.t performance standard
- **Hence, learning agents are able to learn, analyze performance, and look for new ways to improve the performance.**


AGENT
Performance Standard
Critic
Sensors
Percepts
feedback
Learning element
changes
Performance element
knowledge
Environment
Learning goals
Generate experiments
Problem Generator
Effectors
actions

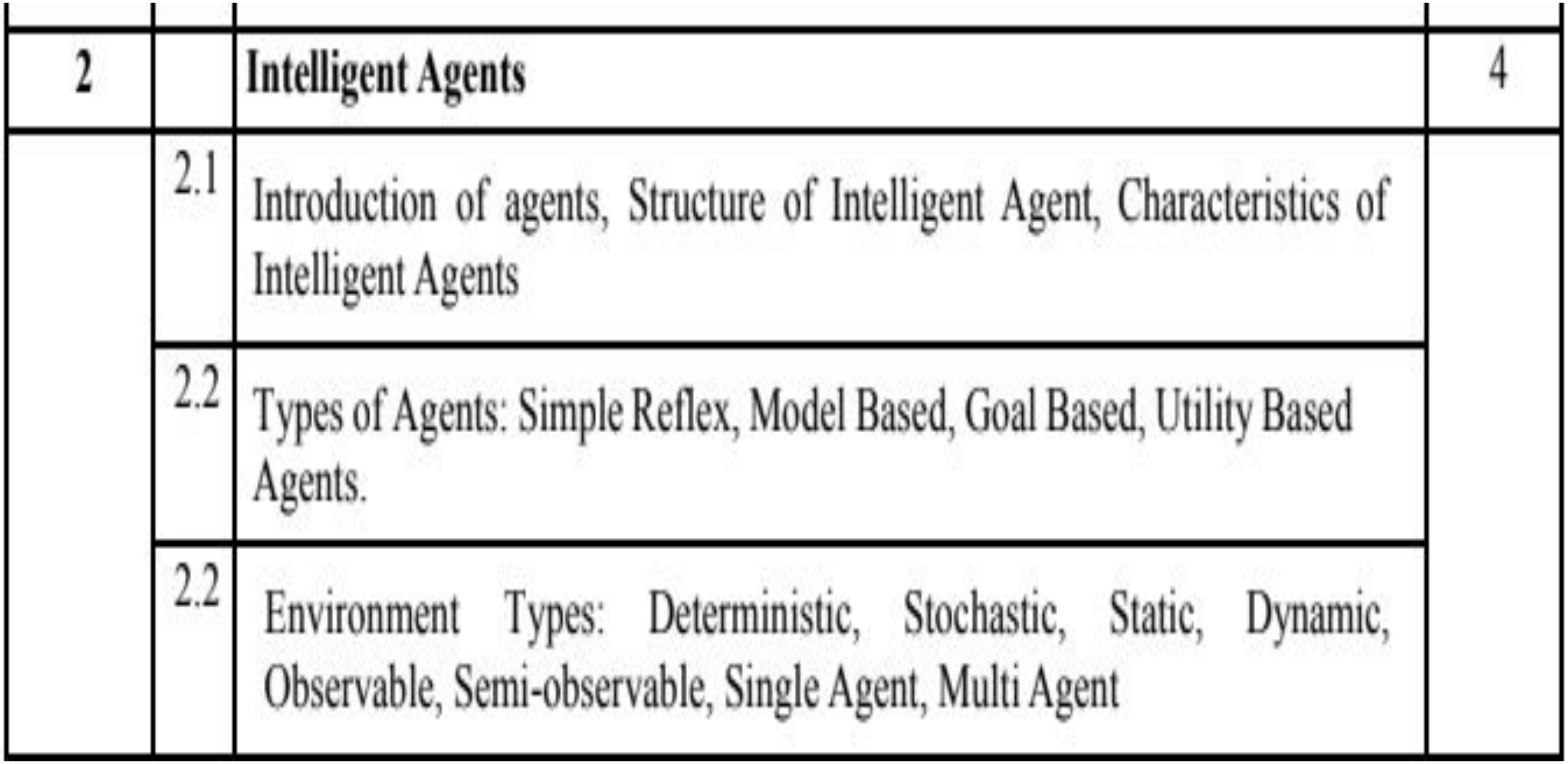
Compare

- Simple reflex agent. It works by finding a **rule** whose condition matches the current situation (as defined by the percept) and then doing the action associated with that rule.
- Model reflex agent with **internal state**. It works by finding a rule whose condition matches the current situation (as defined by the **percept and the stored internal state**) and then doing the action associated with that rule.
- Goal based keep track of world state as well as **set goal** it is trying to achieve and choose an action that will lead to achievement of its goal
- Utility based uses utility function that measures its preferences among states of the world and chooses the action that leads to the **best expected utility**. Expected utility is calculated by averaging all possible outcome state, weighted by probability of outcome .

For each of the following agents develop a

1. PEAS description of task environment ,
2. Environment type and
3. Write agent

- Robot Soccer Player
- Internet book shopping agents
- Autonomous Mars rover
- Mathematician’s theorem proving assistant
- Automated taxi driver
- Medical diagnosis system
- Part-picking robot
- Interactive English tutor



2		**Intelligent Agents**	4
	2.1	Introduction of agents, Structure of Intelligent Agent, Characteristics of Intelligent Agents	
	2.2	Types of Agents: Simple Reflex, Model Based, Goal Based, Utility Based Agents.	
	2.2	Environment Types: Deterministic, Stochastic, Static, Dynamic, Observable, Semi-observable, Single Agent, Multi Agent	

Rational agents

- An agent should strive to "do the right thing", based on what it can perceive and the actions it can perform. The right action is the one that will cause the agent to be most successful

- Performance measure: An objective criterion for success of an agent's behavior

- E.g., performance measure of a vacuum-cleaner agent could be amount of dirt cleaned up, amount of time taken, amount of electricity consumed, amount of noise generated, etc.

Rational agents

- Rational depend on :
  - ✔ The performance measure that defines degree of success.
  - ✔ What the agent knows about the environment.
  - ✔ The actions that the agent can perform.(Actuator)
  - ✔ The agent's percept sequence to date (Sensors)

- Def : Rational Agent: For each possible percept sequence, a rational agent should select an action that is expected to maximize its performance measure, given the evidence provided by the percept sequence and whatever built-in knowledge the agent has.

Rational agents

- Rationality is distinct from omniscience (all-knowing with infinite knowledge)
- Omniscience agent knows the actual outcome of its action and act accordingly
- Agents can perform actions in order to modify future percepts so as to obtain useful information (information gathering, exploration) learning
- An agent is autonomous if its behavior is determined by its own experience (with ability to learn and adapt)

Specifying the task environment

- Task Environment : essentially the “problems” to which rational agent are the “solution”

PEAS

- Must first specify the setting for intelligent agent design

- Consider, e.g., the task of designing an 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

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

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

Agent: Interactive English tutor

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

2		**Intelligent Agents**	4
	2.1	Introduction of agents, Structure of Intelligent Agent, Characteristics of Intelligent Agents	
	2.2	Types of Agents: Simple Reflex, Model Based, Goal Based, Utility Based Agents.	
	2.2	Environment Types: Deterministic, Stochastic, Static, Dynamic, Observable, Semi-observable, Single Agent, Multi Agent	

Environment types

- Fully observable (vs. partially observable): An agent's sensors give it access to the complete state of the environment at each point in time.

- Deterministic (vs. stochastic): The next state of the environment is completely determined by the current state and the action executed by the agent. (If the environment is deterministic except for the actions of other agents, then the environment is strategic)

- Episodic (vs. sequential): The agent's experience is divided into atomic "episodes" (each episode consists of the agent perceiving and then performing a single action), and the choice of action in each episode depends only on the episode itself.

Environment types

- Static (vs. dynamic): The environment is unchanged while an agent is deliberating. (The environment is semidynamic if the environment itself does not change with the passage of time but the agent's performance score does)

- Discrete (vs. continuous): A limited number of distinct, clearly defined percepts and actions.

- Single agent (vs. multiagent): An agent operating by itself in an environment.

Fully observable / Partially Observable

Fully observable

- An agent can always see the entire state of an environment.
- **Example**: Chess

Partially Observable

- An agent can never see the entire state of an environment.
- **Example**: Card game

Deterministic / Stochastic

Deterministic

- An agent's current state and selected action can completely determine the next state of the environment.
- **Example:** Tic tac toe

Stochastic

- A stochastic environment is random in nature and cannot be determined completely by an agent.
- **Example**: Ludo (Any games that involve dice)

Episodic / Sequential

Episodic

- Only the current percept is required for the action
- Every episode is independent of each other
- **Example**: part picking robot

Sequential

- An agent requires memory of past actions to determine the next best actions.
- The current decision could affect all future decisions.
- **Example**: Chess

Single-agent / Multi-agent

Single-agent

- If only one agent is involved in an environment, and operating by itself then such an environment is called single agent environment.
- **Example**: maze

Multi-agent

- if multiple agents are operating in an environment, then such an environment is called a multi-agent environment.
- **Example**: football

Discrete / Continuous

Discrete

- The environment consists of a finite number of actions that can be deliberated in the environment to obtain the output.
- **Example**: chess

Continuous

- The environment in which the actions performed cannot be numbered ie.,is not discrete, is said to be continuous.
- **Example**: Self-driving cars

Known / Unknown

Known

- In a known environment, the results for all actions are known to the agent.

- **Example**: Card game

Unknown

- In unknown environment, agent needs to learn how it works in order to perform an action.
- **Example**: A new video game

Environment types

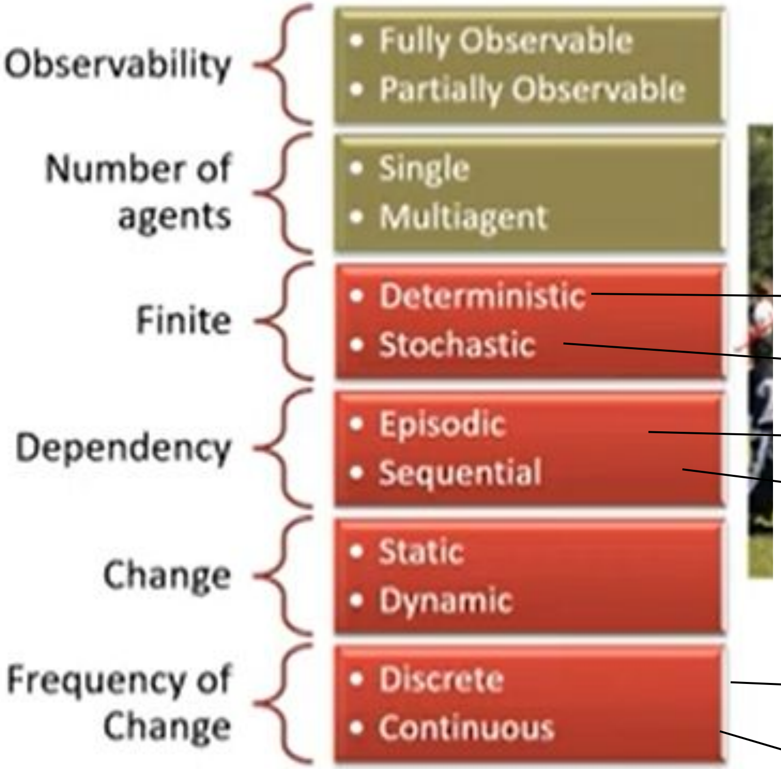
	Chess with a clock	Chess without a clock	Taxi driving
Fully observable			
Deterministic			
Episodic			
Static			
Discrete			
Single agent			

Environment types

	Chess with a clock	Chess without a clock	Taxi driving
Fully observable	Yes	Yes	No
Deterministic	Strategic	Strategic	No
Episodic	No	No	No
Static	Semi	Yes	No
Discrete	Yes	Yes	No
Single agent	No	No	No

- The environment type largely determines the agent design
- The real world is (of course) partially observable, stochastic, sequential, dynamic, continuous, multi-agent

Types of environment



Observability
- Fully Observable
- Partially Observable

Number of agents
- Single
- Multiagent

Finite
- Deterministic
- Stochastic

Dependency
- Episodic
- Sequential

Change
- Static
- Dynamic

Frequency of Change
- Discrete
- Continuous

chess

Random - football, cricket (real time)

non sequential : chatbot

current action will affect future : Badminton

environment will not change : chess board

environment will change continuously : car driving

Environment

Environment types

	Chess with a clock	Chess without a clock	Taxi driving
Fully observable	Yes	Yes	No
Deterministic	Strategic	Strategic	No
Episodic	No	No	No
Static	Semi	Yes	No
Discrete	Yes	Yes	No
Single agent	No	No	No

- The environment type largely determines the agent design
- The real world is (of course) partially observable, stochastic, sequential, dynamic, continuous, multi-agent

THANK
YOU.

The structure of Agent

- The job of AI is to design the **agent program:** that implement the agent function mapping percepts to actions. We assume this program will run on some sort of computing device, which we will call the **architecture.**
- In general, the architecture makes the percepts from the sensors available to the program, runs the program, and feeds the program's action choices to the effectors as they are generated.

agent = architecture + program

- **function** TABLE-DRIVEN-AGENT(percept) **returns** *action*

  static: *percepts,* a sequence, initially empty

  table, a table of action, indexed by percept sequences, initially fully specified

  append *percept* to the end of *percepts*

  action <— LOOKUP(percepts, table) **return** *action*

Agent Program :Keeps track of percept sequence and use it to index into a table of action.

The table represent explicitly the agent function that agent program embodies

Table-lookup agent

- Drawbacks:
  - Huge table
  - Take a long time to build the table
  - No autonomy
  - Even with learning, need a long time to learn the table entries.

The key challenge for AI is to find out how to write programs that to the extent possible produce rational behavior from small amount of code rather than large number of table entries.