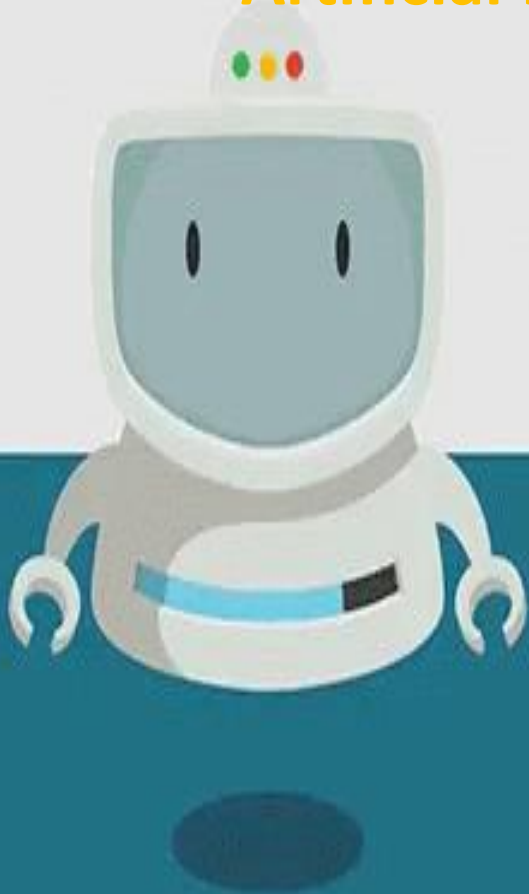




Artificial Intelligence (CS-401)



Chapter 2: Intelligent Agents

Instructor: Saeeda Kanwal

Lesson Objectives



Agents and environments

- ☐ Rationality
- ☐ PEAS (Performance measure, Environment, Actuators, Sensors)
- ☐ Environment types
- ☐ Agent types

Agents

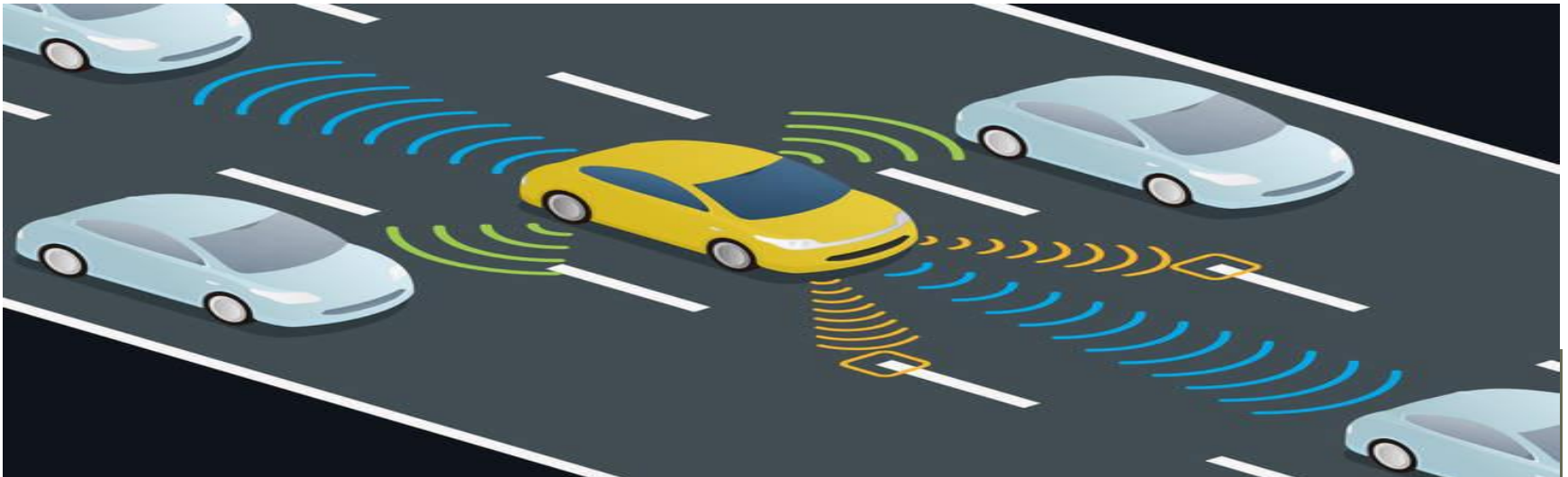
An **agent** is anything that can be viewed as **perceiving** its **environment** through **sensors** and **acting** upon that environment through **actuators**.

Human agent: Eyes, ears, and other organs for **sensors**; hands, legs, mouth, and other body parts for **actuators**

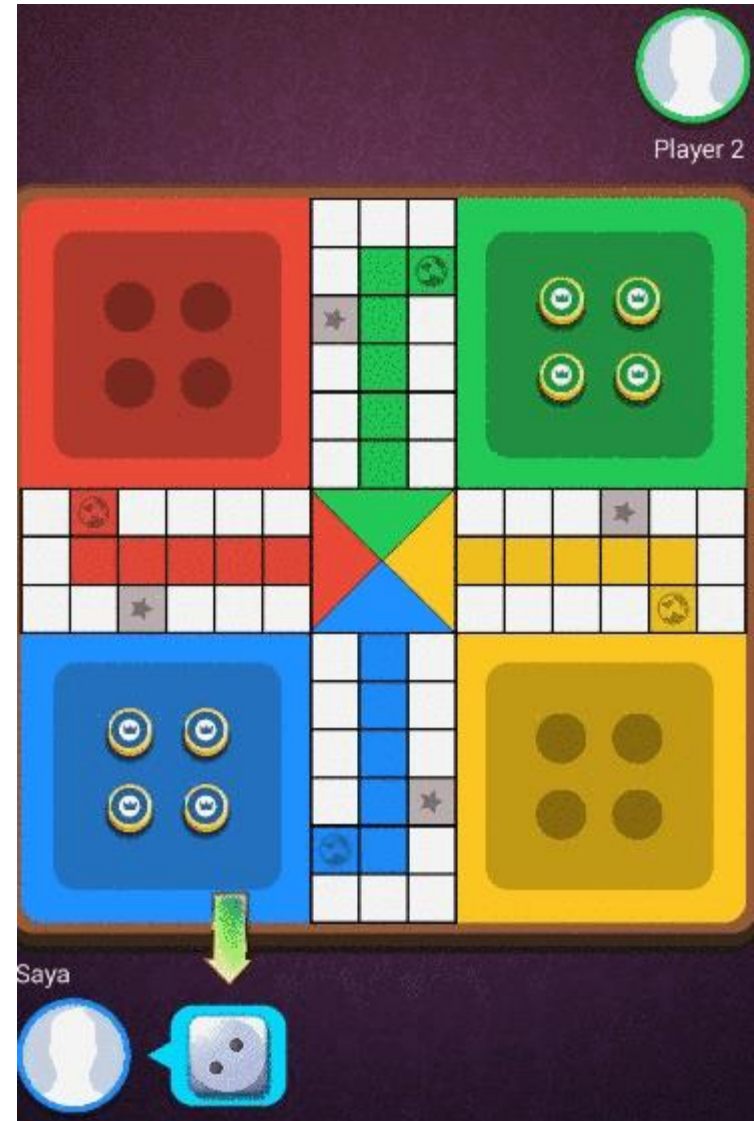
Robotic agent: Cameras and infrared range finders for **sensors**; various motors for **actuators**.

Software agent: Software agent is a **computer program** that acts **for a user or other program**: an agreement to act on one's behalf.

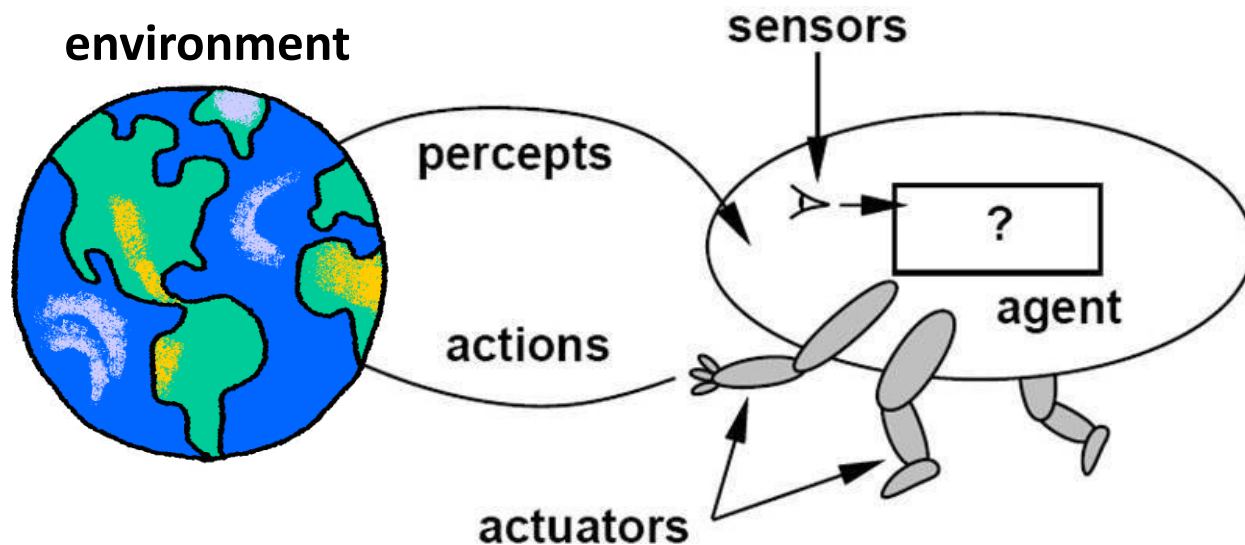
Robotic Agent



Software Agent



Agents and Environments



The **agent function** maps from percept histories to actions:
 $[f: P^* \rightarrow A]$

The **agent program** runs on the physical **architecture** to produce f

Agent = Architecture + Program

Vacuum-cleaner world



Percepts: location and contents, e.g., [A, Dirt]

Actions: *Left, Right, Suck, Do Nothing*

A vacuum-cleaner Agent



Tabulation of an agent function of the vacuum-cleaner

Percept sequence	Action
$[A, \textit{Clean}]$	<i>Right</i>
$[A, \textit{Dirty}]$	<i>Suck</i>
$[B, \textit{Clean}]$	<i>Left</i>
$[B, \textit{Dirty}]$	<i>Suck</i>
$[A, \textit{Clean}], [A, \textit{Clean}]$	<i>Right</i>
$[A, \textit{Clean}], [A, \textit{Dirty}]$	<i>Suck</i>
\vdots	\vdots

function REFLEX-VACUUM-AGENT($[location, status]$) **returns** an action

if $status = \textit{Dirty}$ **then return** *Suck*
else if $location = A$ **then return** *Right*
else if $location = B$ **then return** *Left*

Intelligent Agents

- ❑ Agents and environments



Rationality

- ❑ PEAS (Performance measure, Environment, Actuators, Sensors)
- ❑ Environment types
- ❑ Agent types

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.

Agent Goals:

High Performance measure (an objective criterion for success of an agent's behavior) e.g. self driving car agent could be

Safety

Optimized Result (Decision)e.g. amount of time taken(shortest route, less traffic)

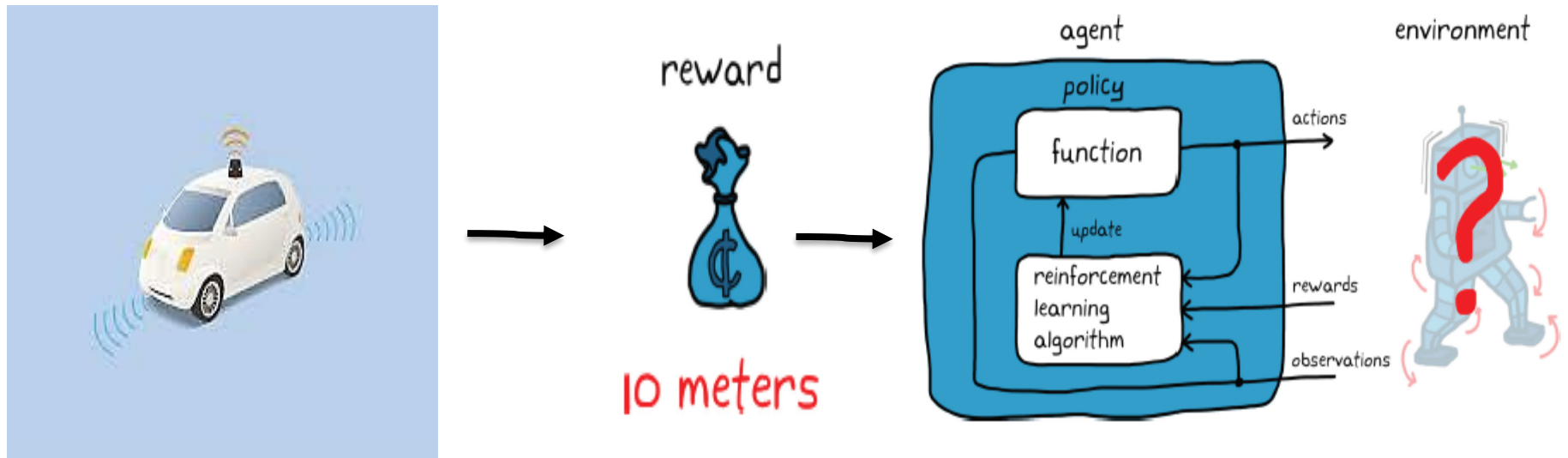
Rational Action (Right Action) e.g. Speed

Rational Agents

Agents can perform actions in order to modify future percepts so as to obtain useful information (information gathering, exploration, learn).

An agent is **autonomous** if its behavior is determined by its own experience (with ability to learn and adapt).

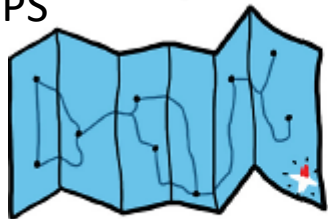
Autonomous Agent Example: Reinforcement Learning



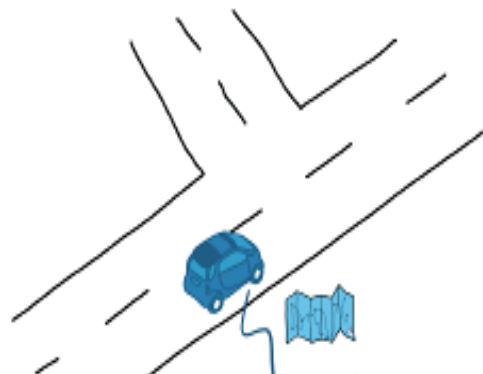
$$\text{reward} = \begin{cases} 1 & \text{for state} = 10 \text{ meters} \\ 0 & \text{for state} \neq 10 \text{ meters} \end{cases}$$

reward = $\begin{cases} 1 & \text{for state} = 10 \text{ meters} \\ 0 & \text{for state} \neq 10 \text{ meters} \end{cases}$

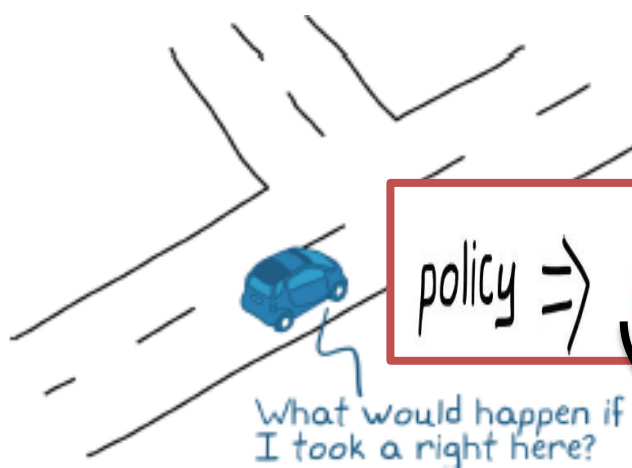
GPS



this road map should help!



I know right is low reward.
I'll explore straight and left.



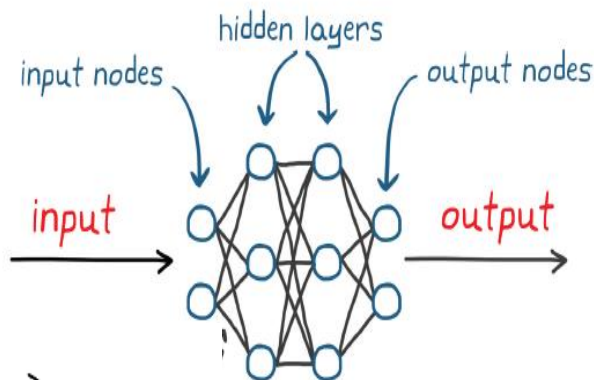
What would happen if
I took a right here?

policy \Rightarrow actions = function (state observations)

states
actions \rightarrow

	a1	a2	a3
s1	value	value	value
s2	value	value	value
s3	value	value	value

\rightarrow value



policy

current state \rightarrow

	a1	a2	a3	action with the highest value
s1	0	+1	-1	
s2	+1	0	-1	a1
s3	-1	-1	+1	

Intelligent Agents

- Agents and environments

- Rationality



PEAS (Performance measure,
Environment, Actuators, Sensors)

- Environment types

- Agent types

PEAS

When designing a rational/intelligent agent, we keep in mind PEAS.

PEAS: **P**erformance measure, **E**nvironment, **A**ctuators, **S**ensors

Consider, e.g., the task of designing an automated taxi driver:

- Performance measure
- Environment
- Actuators
- Sensors

PEAS

Agent: automated taxi driver

- **Performance measure:** *Safe, fast, legal, comfortable trip, maximize profits*
- **Environment:** *Roads, other traffic, people and objects in/around the street*
- **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

Intelligent Agents

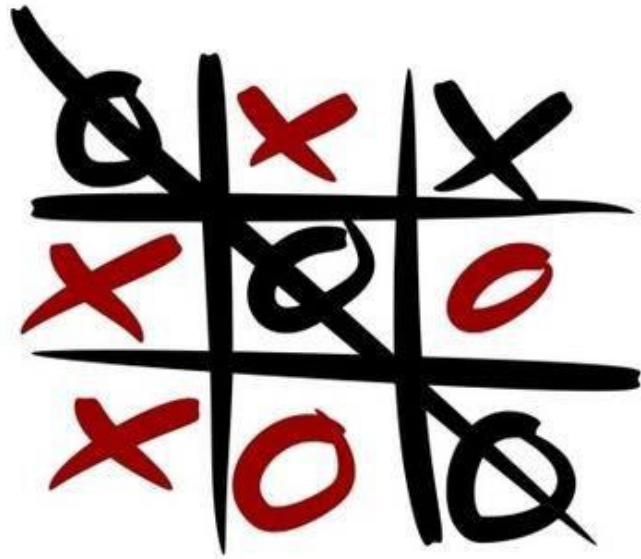
- ❑ Agents and environments
- ❑ Rationality
- ❑ PEAS (Performance measure, Environment, Actuators, Sensors)
-  Environment types
- ❑ Agent types

Environment Types

- 1) Fully observable vs. 2) Partially observable
 - Sensors capture all relevant information from the environment
- 3) Deterministic vs. 4) Stochastic (non-deterministic)
 - Changes in the environment are predictable
- 5) Episodic vs. 6) Sequential (non-episodic)
 - Independent perceiving-acting episodes
- 7) Static vs. 8) Dynamic
 - No changes while the agent is “thinking”
- 9) Discrete vs. 10) Continuous
 - Limited number of distinct percepts/actions
- 11) Single vs. 12) Multiple agents
 - Interaction and collaboration among agents
 - Competitive, cooperative

Environment Types

❖ **Fully observable** (vs. partially observable): An agent's sensors can measure all relevant aspects of the environment at each point in time.



Tic Tac Toe is Fully Observable. Cards are Partially Observable.

Environment Types

- ❖ **Deterministic** (vs. stochastic): The next state of the environment is completely determined by the current state and the action executed by the agent. (Stochastic: next state cannot be predicted with certainty)



Chess is deterministic. Taxi driver is stochastic.

Environment Types

❖ **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.



Robot is Episodic. Taxi driver is sequential.

Environment Types

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



Taxi driver is dynamic. Chess is static.

Environment Types

❖ **Discrete** (vs. continuous): A limited number of distinct, clearly defined percepts and actions (**Defined Rules and Actions**).



Chess has a finite number of distinct states. thus it is discrete; however the Taxi-driving is not.

Environment Types

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



Crossword is Single agent, while Chess is a multi-agent environment.

Environment Types

Task Environment	Oberservable	Deterministic	Episodic	Static	Discrete	Agents
<i>Crossword puzzle</i>	fully	deterministic	sequential	static	discrete	single
<i>Chess with a clock</i>	fully	strategic	sequential	semi	discrete	multi
<i>Taxi driver</i>	partially	stochastic	sequential	dynamic	conti.	multi
<i>mushroom-picking</i>	partially	stochastic	episodic	dynamic	conti.	single

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

Intelligent Agents

- ❑ Agents and environments
- ❑ Rationality
- ❑ PEAS (Performance measure, Environment, Actuators, Sensors)
- ❑ Environment types

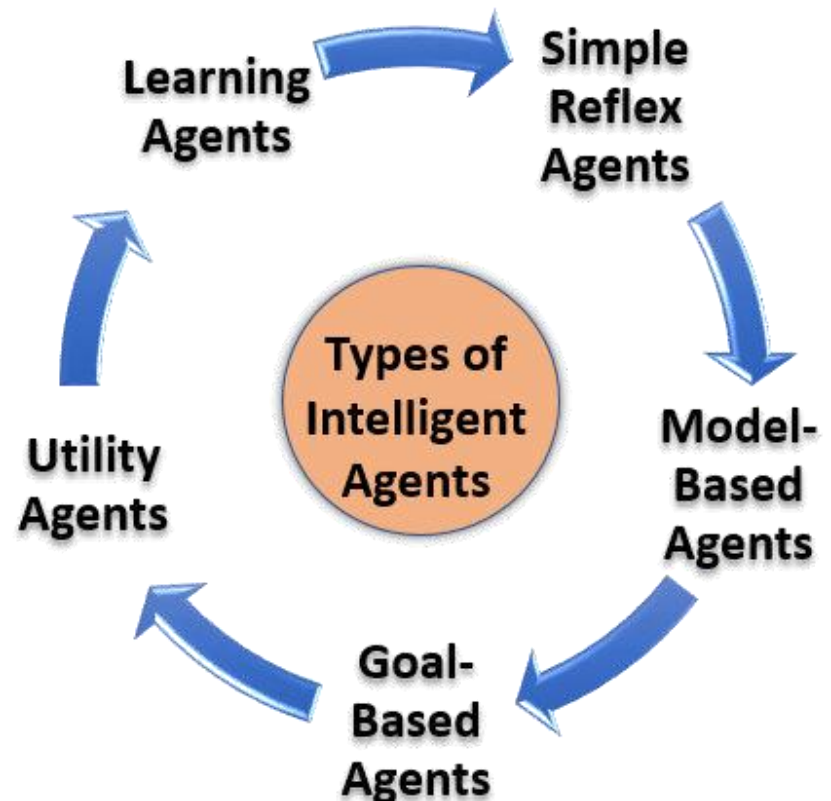


Agent types

Agent Types

Agents can be divided into five (05) basic types according to the degree of **perceived intelligence** and **capacity to change the environment**:

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

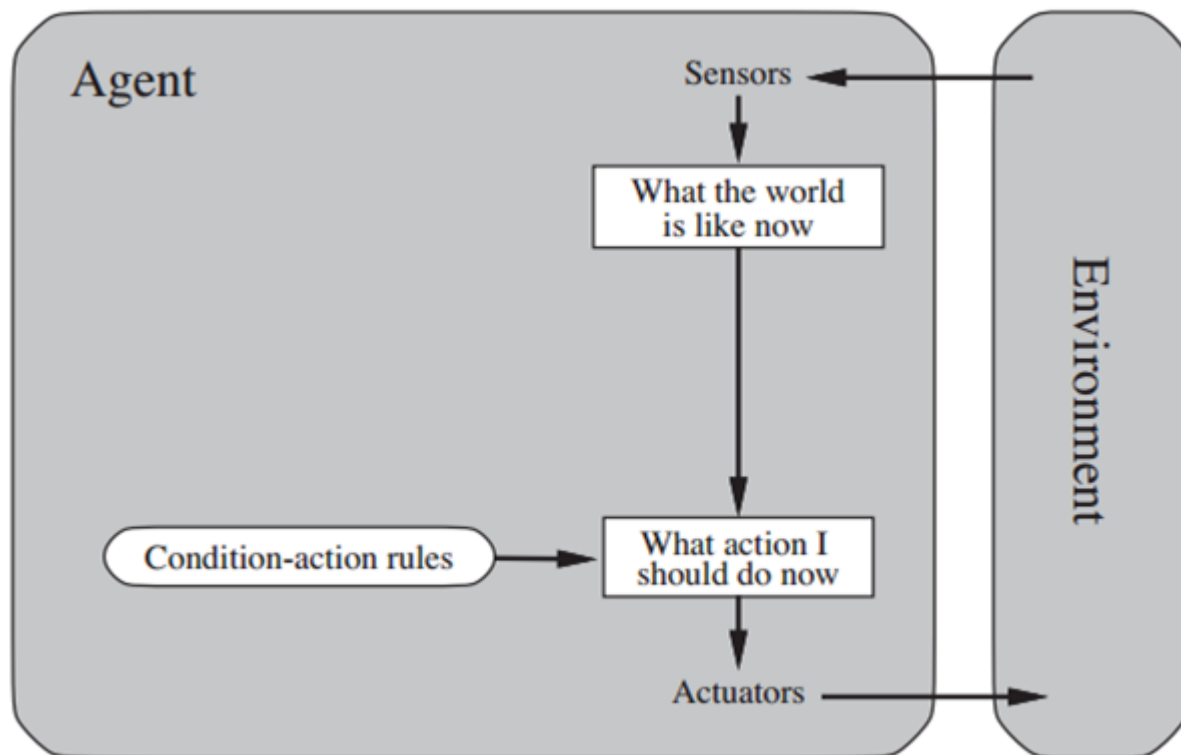


Simple Reflex Agents

The agent selects an action(s) based on the **current precept and conditions**, ignoring the rest of the precept history (**previous state**).

if x happens, do y

e.g. Poker



Simple Reflex Agents

- Very limited Intelligence
- No Knowledge/Perception about the previous or next state
- Operates in Partially Observable Environments
- Infinite loops are unavoidable

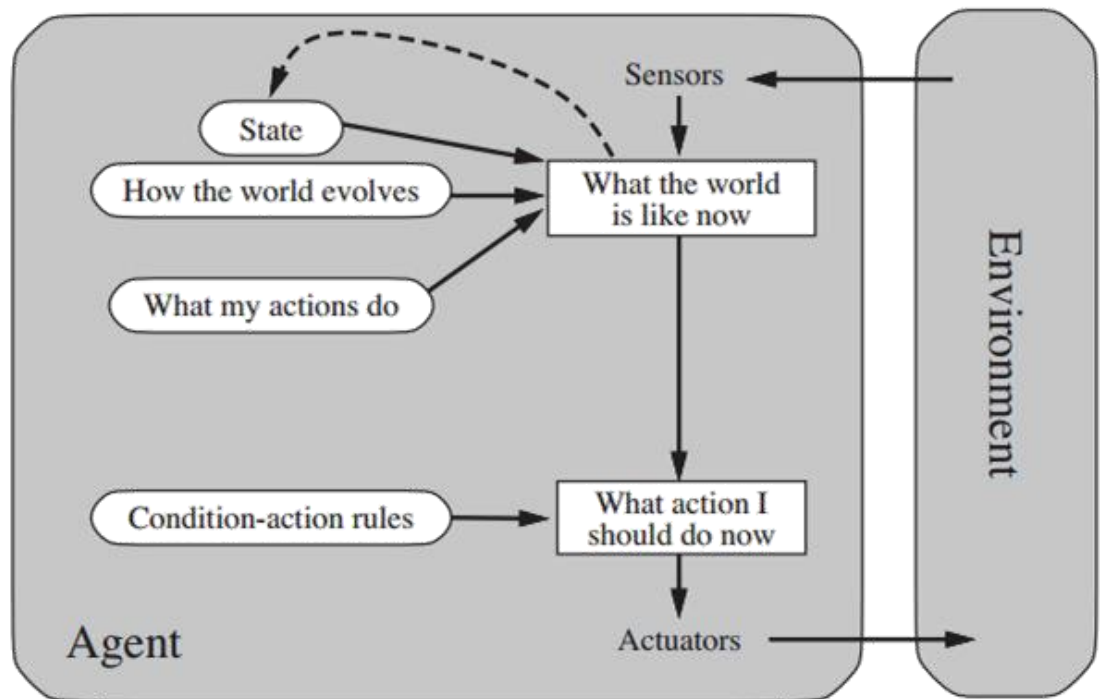
Model-based Reflex Agents

The agent decides its action(s) based on a predefined set of condition-action rules.

Depending on the state of the world, different actions are appropriate.

State is evaluated in terms of how it changed from the **previous state**.

e.g., A Roomba Cleaner Robot, a telephone operator/answering machine.

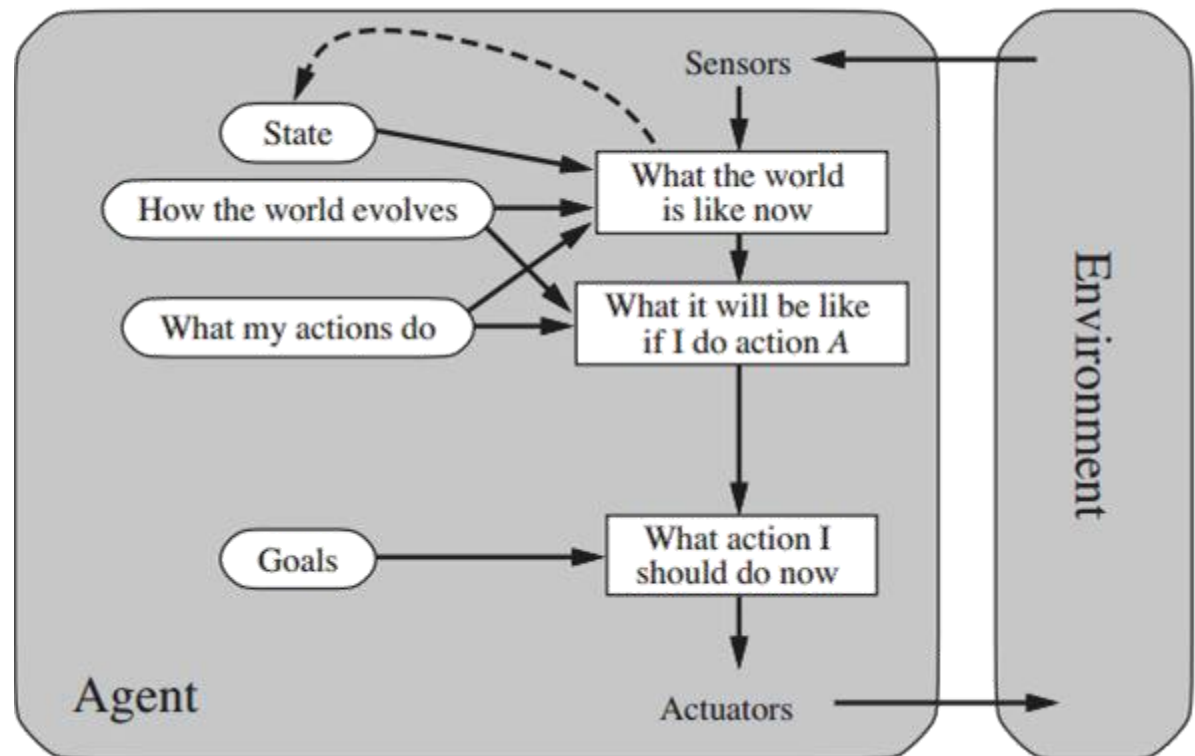


Goal-based Agents

The agent decides its action(s) based on a **known goal**.

These agents have all of the above and **goal**.

Involves consideration of the **previous and future states**.



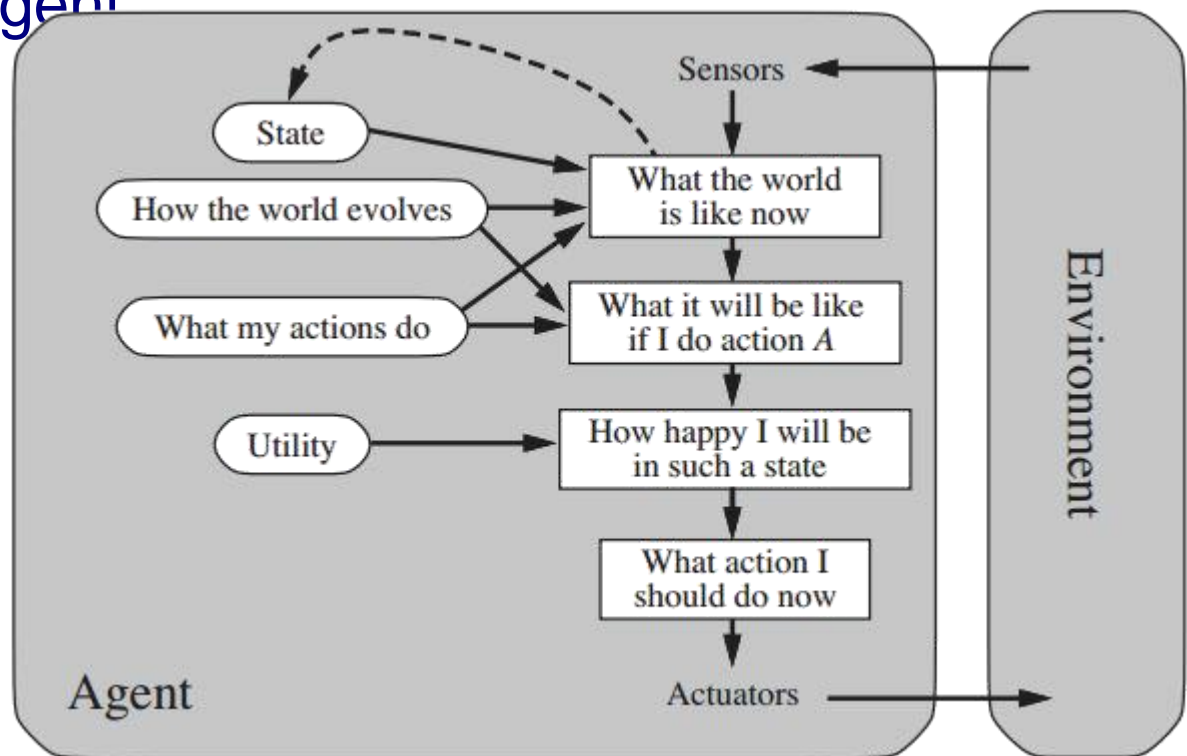
e.g., a GPS system finding a path to certain destination.

Utility-based Agents

The agent decides its action(s) based on utilities/
preferences.

Utility function to decide which world state (**Optimality**) is
better for an agent

e.g., A GPS
system finding a
shortest/fastest/sa
fer path to certain
destination.



Learning Agents

The agent adapts its action(s) based on **feedback** (not only sensors).

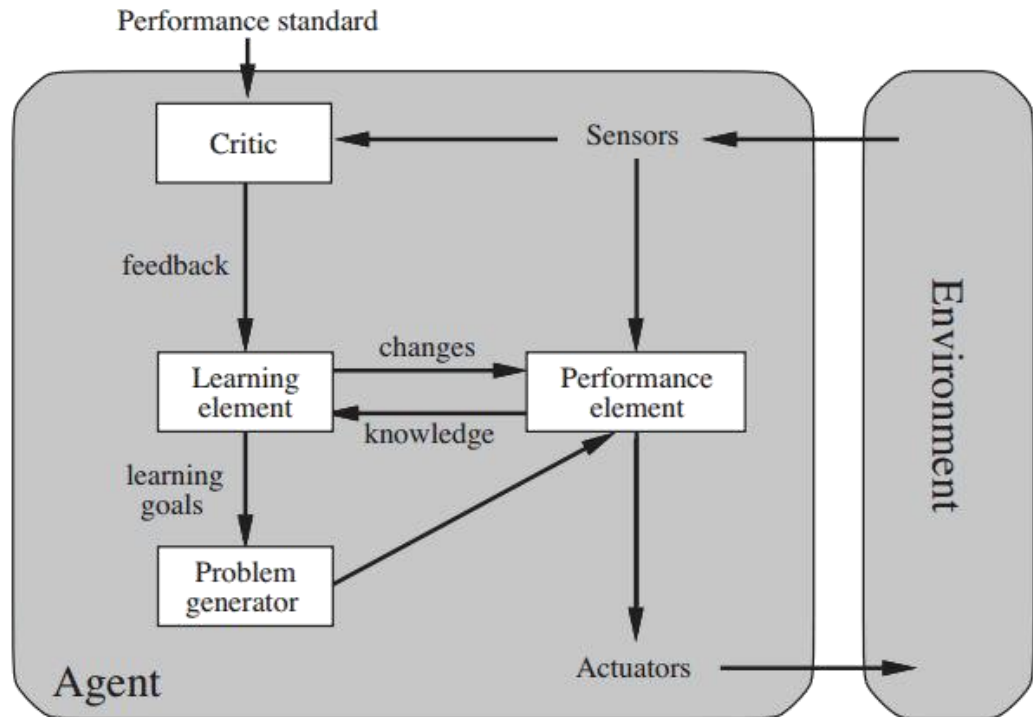
Learning element - responsible for **making improvements** from past events.

Performance element - what we have up to now considered to be the entire agent. Takes percepts and then decides on actions.

Critic - gives the learning element feedback on how the performance element is doing and if it needs to be modified.

Problem generator - it suggests actions that lead to new and informative experiences.

e.g., human agent



Summary

- **Intelligent Agents**
- **IA Behavior**
- **IA Structure**
- **Environment types**
- **Agent Types:** Reflex, state-based, goal-based, utility-based