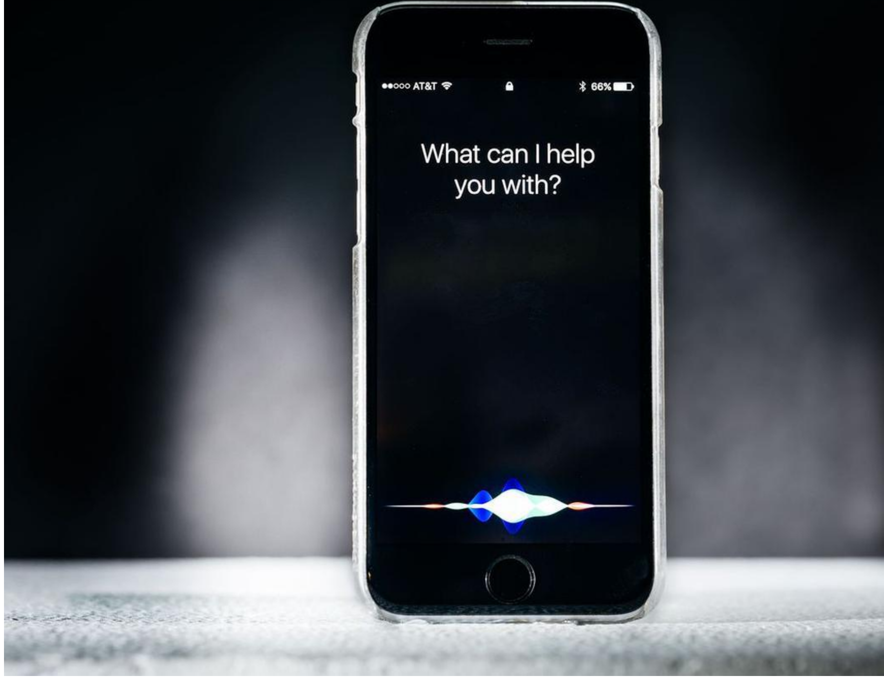
Module 2

Chapter 2

2	**Intelligent Agents**		**5**	**CO1**
	2.1	Agents and Environments, The concept of rationality, The nature of environment, The structure of Agents, Types of Agents, Learning Agent, function of agent program		



AI assistants- Alexa and Siri → Examples of **Intelligent Agents**

- They use sensors such as microphones and other inputs, to **perceive** a request and they draw conclusions on their collective experience and knowledge via supercomputers and data banks all over the world to make a decision from the internet without the user's help.

- **RATIONAL AGENT** = Makes decisions → as a person, firm, machine, or software.
  - Carries out an action with the best outcome after considering **past and current percepts** (agent’s perceptual inputs at a given instance).
- (Rational)**AI system** = **Agent + its environment**.
  - Agents act in their environment.
  - Environment may contain other agents.
- An agent :
  - perceives its environment through **sensors**
  - acts upon that environment through **actuators**

ARCHITECTURE=Sensors + Actuators

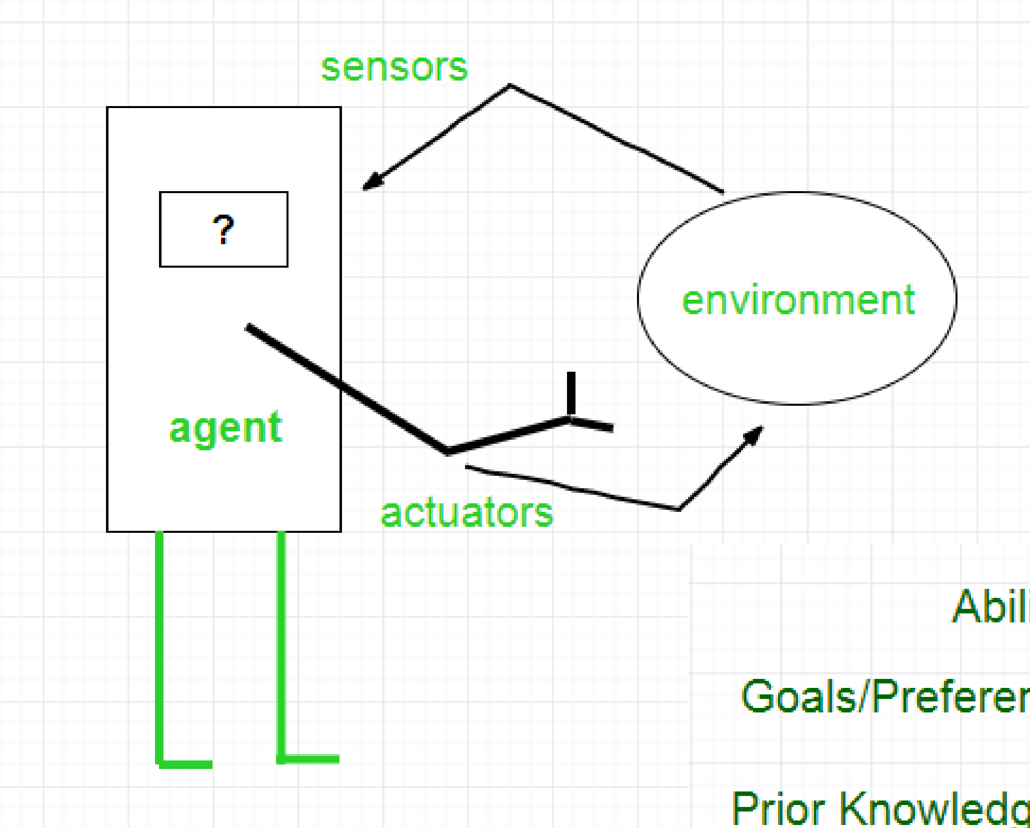
AGENT= ARCHITECTURE+PROGRAM

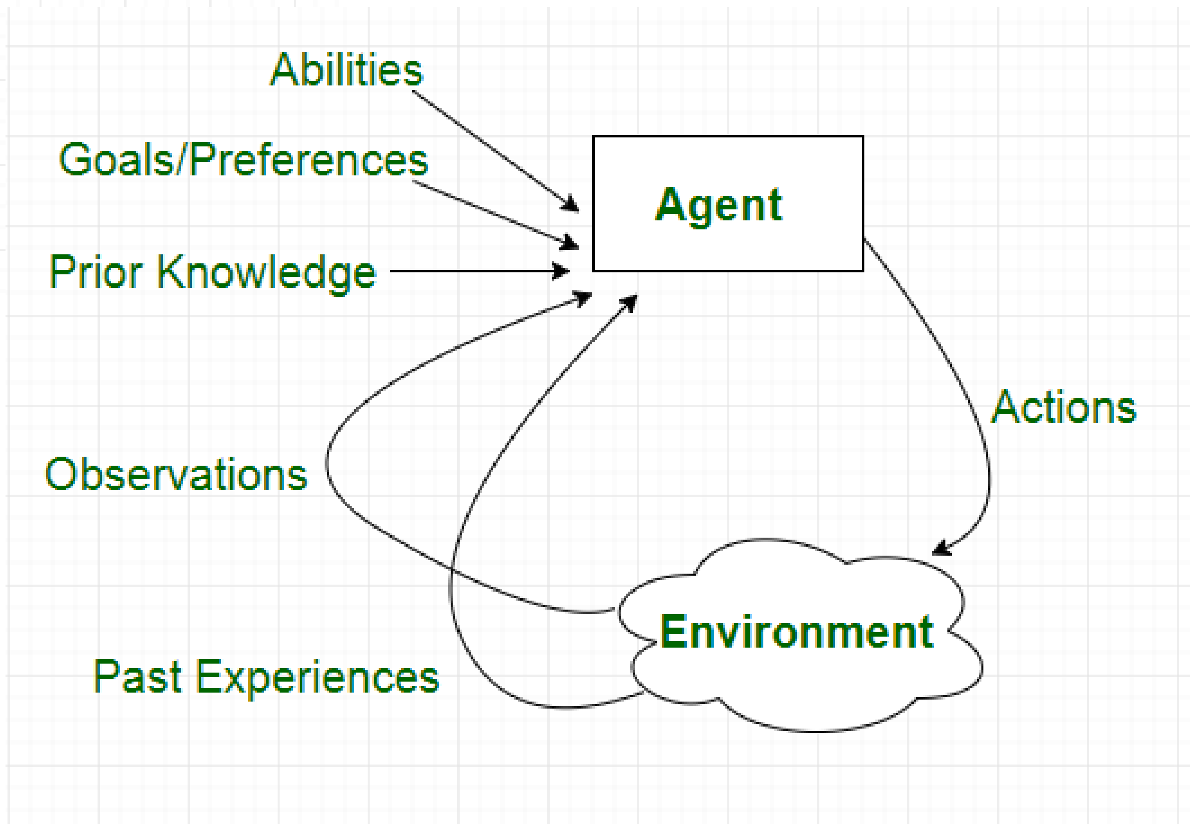
Agents and Environments

– A human agent has eyes,ears, and other organs for **sensors** and hands, legs, vocal tract, and so on for **actuators**.

– A robotic agent might have cameras and infrared range finders for **sensors** and various motors for **actuators**.

– A software agent receives keystrokes, file contents, and network packets as **sensory** inputs and **acts** on the environment by displaying on the **screen**, writing files, and sending network packets.


sensors
?
environment
agent
actuators


Abilities
Goals/Preferences
Prior Knowledge
Agent
Actions
Observations
Environment
Past Experiences

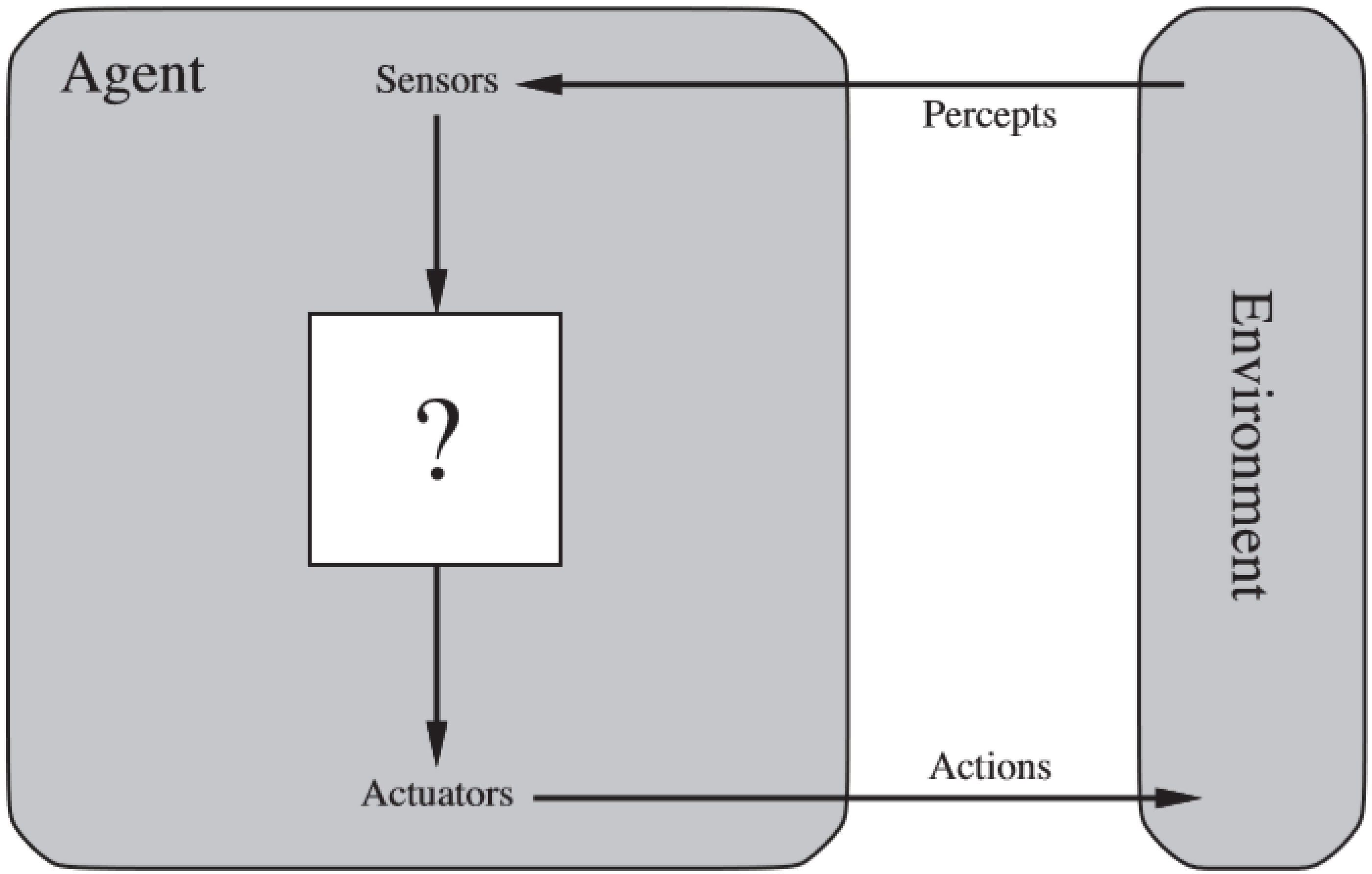


Figure 2.1 Agents interact with environments through sensors and actuators.

- **PERCEPT**: Agent's perceptual inputs at any given instant.
  - Agent's percept sequence is the **complete history** of everything the agent has ever perceived.
  - Agent's choice of action at any given instant can depend on the entire percept sequence observed to date, **but not on anything it hasn't perceived!**
- **AGENT FUNCTION: Maps** any given **percept** sequence to an **action**
- **AGENT PROGRAM:** The agent function for an artificial agent will be implemented by an agent program.

Classic Vacuum cleaner problem

- **Vacuum cleaner problem** is a well-known search problem for an agent which works on A I.
- Vacuum cleaner is our **agent**.
- **Goal** of this agent, is to clean up the whole area.
- We have two rooms and one vacuum cleaner.
- There is dirt in both the rooms and it is to be cleaned.
- The vacuum cleaner is present in any one of these rooms.
- Aim :- Reach a state in which both the rooms are clean and are dust free.
- **Operations**: Move Left , Move Right, Suck Dirt.

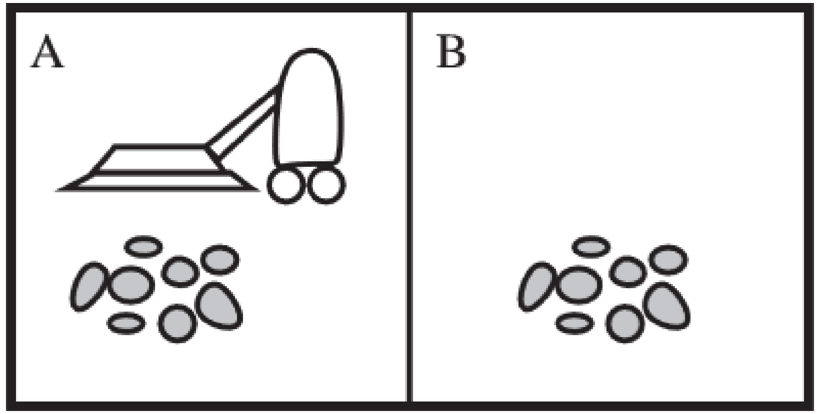


Figure 2.2 A vacuum-cleaner world with just two locations.

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*
⋮	⋮
$[A, Clean], [A, Clean], [A, Clean]$	*Right*
$[A, Clean], [A, Clean], [A, Dirty]$	*Suck*
⋮	⋮

Figure 2.3 Partial tabulation of a simple agent function for the vacuum-cleaner world shown in Figure 2.2.

The concept of rationality

- A rational agent is one that does the **RIGHT THING**
- **PERFORMANCE MEASURE** evaluates any given sequence of "environment states".
  - Notice that we said environment states, **not "agent" states**
  - Not one fixed performance measure for all tasks and agents

– Example:- (Notorious)**Human Agents**[Nobel Prize](Sour Grapes !)

v/s **Vacuum Cleaner Agent** (Rational Agent)-

clean,dump,clean…

General Rule:- It is better to design performance measures according to what one actually wants in the "**environment**", rather than according to how one thinks the agent should behave

Rationality of Agent depends on 4 things

- The **PERFORMANCE MEASURE** that defines the criterion of success.
- The **agent's prior knowledge of the ENVIRONMENT.**
- The **ACTIONS that the agent can perform**
- The **agent's percept SEQUENCE to date**

Defn: Percept sequence= 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.]

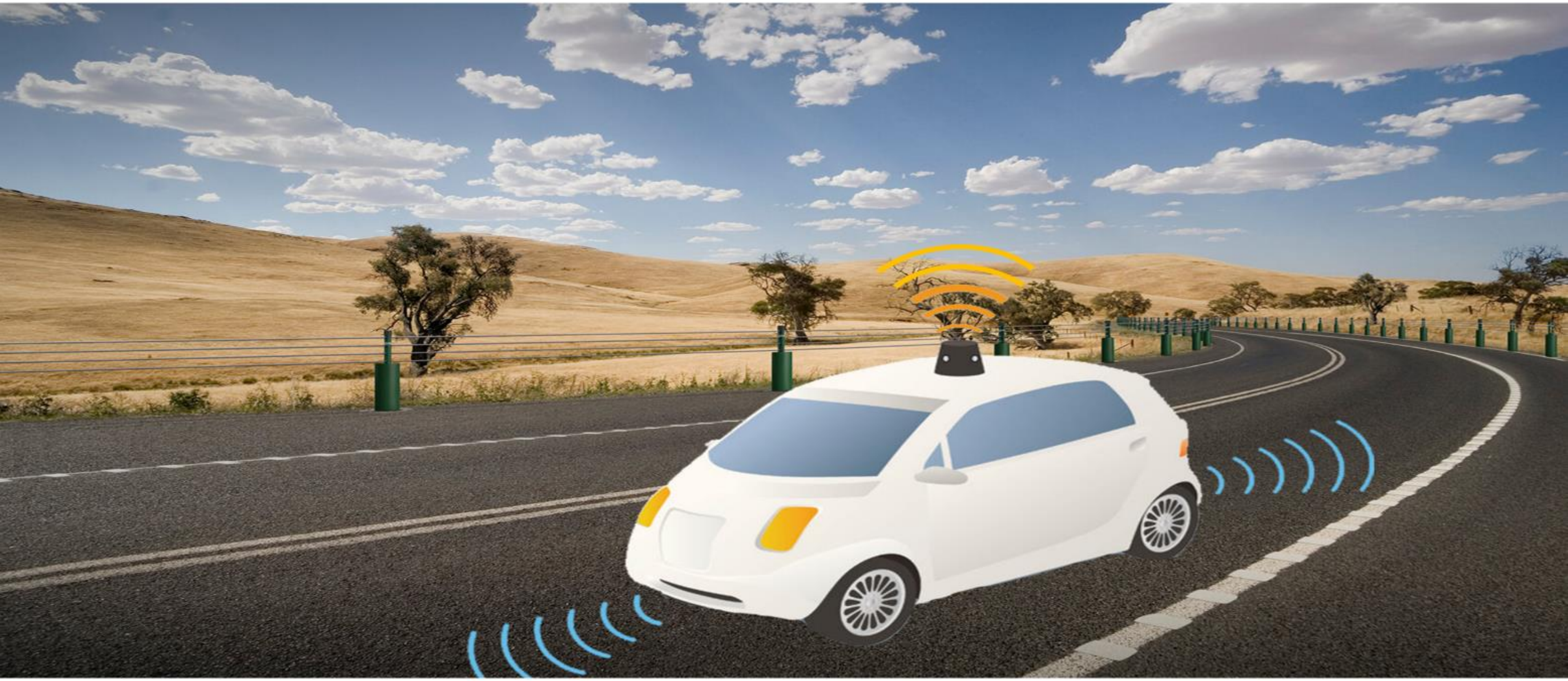
* PEAS -Performance Measures, Environment, Actuators, and Sensors

PEAS is a type of model on which an AI agent works upon.

- **Performance Measure**: Judge the performance of the agent.
  - For example, in case of **pick and place robot**, nos of correct parts in a bin can be the performance measure.
- **Environment**: The real environment where the agent needs to deliberate actions.

- **Actuators:**
  - Tools, equipment using which agent performs **actions** in the environment.
  - This works as **OUTPUT** of the agent.
- **Sensors**:
  - Tools using which agent captures the **state** of the environment.
  - This works as **INPUT** to the agent

PEAS descriptor for Automated Car Driver



PEAS descriptor for Automated Car Driver

- **Performance Measure:**
  - Safety: Automated system should be able to drive the car safely without dashing anywhere.
  - Optimum speed: Automated system should be able to maintain the optimal speed depending upon the surroundings.
  - Comfortable journey: Automated system should be able to give a comfortable journey to the end user.

- **Environment**:
  - Roads: Automated car driver should be able to drive on any kind of a road ranging from city roads to highway.
  - Traffic conditions: You will find different sort of traffic conditions for different type of roads.

- **Actuators**:
  - Steering wheel: used to direct car in desired directions.
  - Accelerator, brake,gear: To increase or decrease speed of the car.
- **Sensors**: To take i/p from environment in car driving example:- cameras.

- Refer Vacuum Cleaner Agent Example on Pg 38- Russell Norvig

PEAS (Performance, Environment, Actuators, Sensors) description

Performance: Safety, time, legal drive, comfort

Environment: Roads, other vehicles, road signs, pedestrian

Actuators: Steering, accelerator, brake, signal, horn

Sensors: Camera, GPS, speedometer, odometer, accelerometer, sonar.

- Performance –which qualities it should have?
- Environment –where it should act?
- Actuators –how will it perform actions?
- Sensors –how will it perceive environment?

Mathematician's theorem-proving assistant

- P: good math knowledge, can prove theorems accurately and in minimal steps/time
- E: Internet, library
- A: display
- S: keyboard

Autonomous Mars rover

- P: Terrain explored and reported, samples gathered and analysed
- E: Launch vehicle, lander, Mars
- A: Wheels/legs, sample collection device, analysis devices, radio transmitter
- S: Camera, touch sensors, accelerometers, orientation sensors, wheel/joint encoders, radio receiver

Internet book-shopping agent

- P: Obtain requested/interesting books, minimize expenditure
- E: Internet
- A: Follow link, enter/submit data in fields, display to user
- S: Web pages, user requests

Robot soccer player

- P: Winning game, goals for/against
- E: Field, ball, own team, other team, own body
- A: Devices (e.g., legs) for locomotion and kicking
- S: Camera, touch sensors, accelerometers, orientation sensors, wheel/joint encoders

Agent Type	Performance Measure	Environment	Actuators	Sensors
Medical diagnosis system	Healthy patient, reduced costs	Patient, hospital, staff	Display of questions, tests, diagnoses, treatments, referrals	Keyboard entry of symptoms, findings, patient's answers
Satellite image analysis system	Correct image categorization	Downlink from orbiting satellite	Display of scene categorization	Color pixel arrays
Part-picking robot	Percentage of parts in correct bins	Conveyor belt with parts; bins	Jointed arm and hand	Camera, joint angle sensors
Refinery controller	Purity, yield, safety	Refinery, operators	Valves, pumps, heaters, displays	Temperature, pressure, chemical sensors
Interactive English tutor	Student's score on test	Set of students, testing agency	Display of exercises, suggestions, corrections	Keyboard entry

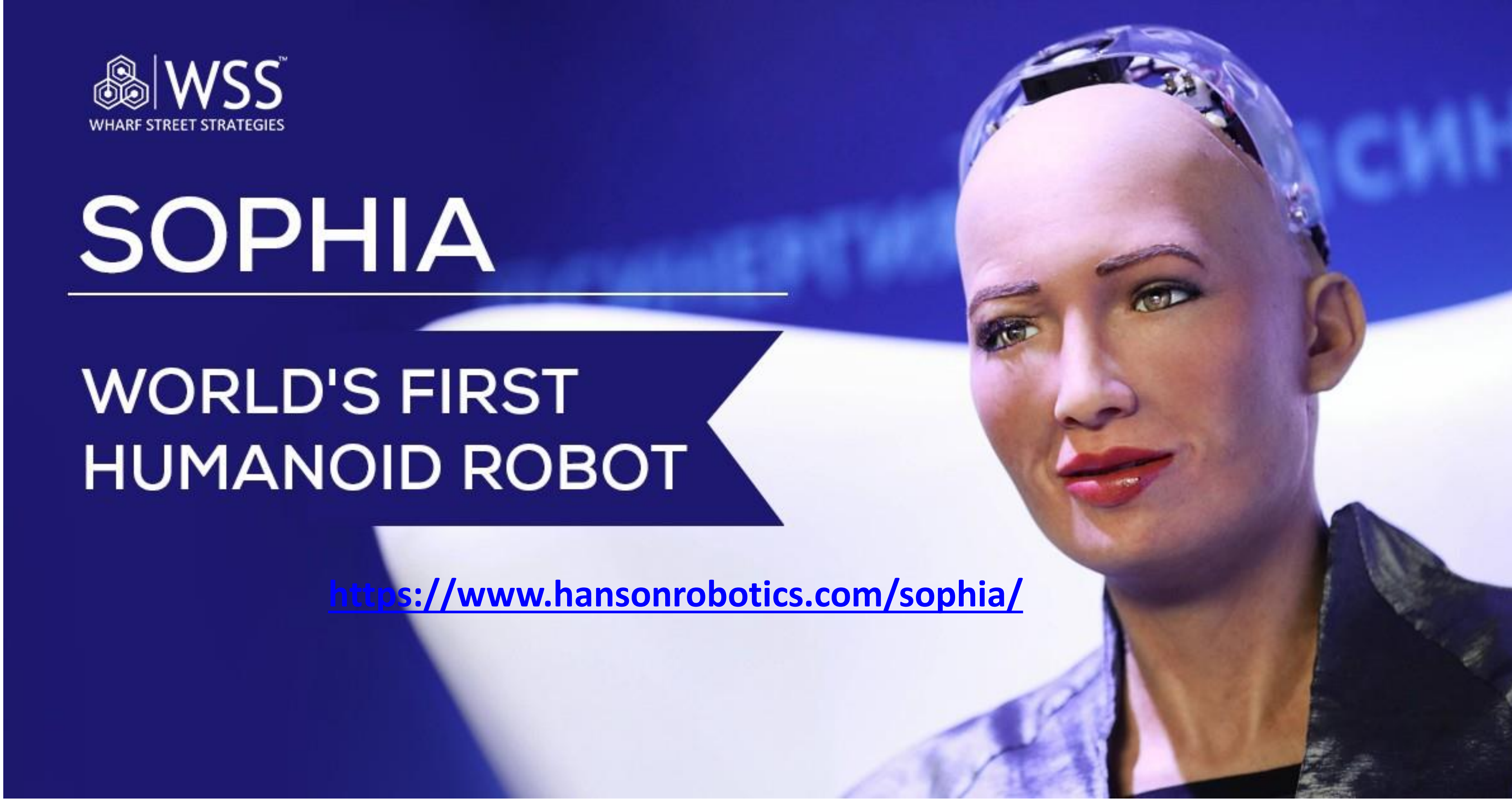
Figure 2.5 Examples of agent types and their PEAS descriptions.

Omniscience, learning, and autonomy

- An **OMNISCIENT AGENT** knows the actual outcome of its actions and can act accordingly;
  - but omniscience is impossible in reality.

Consider the following example:

I am walking along the road one day and I see an old friend across the street. There is no traffic nearby and I'm not otherwise engaged, so, being rational, I start to cross the street. Meanwhile, at 33,000 feet, a cargo door falls off a passing airliner, and before I make it to the other side of the street I am flattened. Was I irrational to cross the street? It is unlikely that my obituary would read "Idiot attempts to cross street."

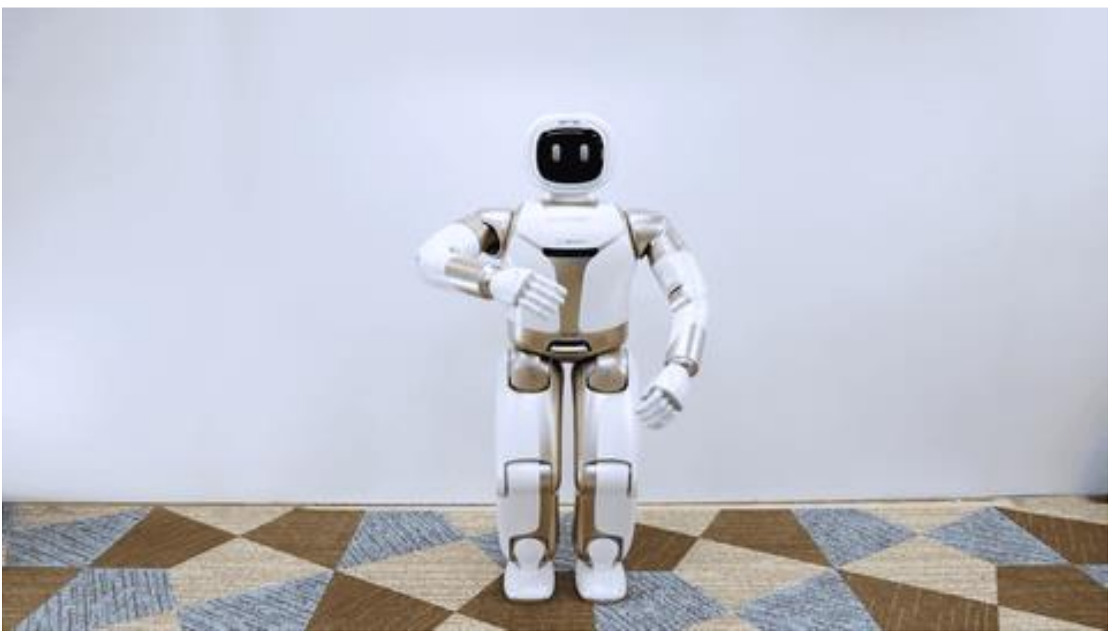


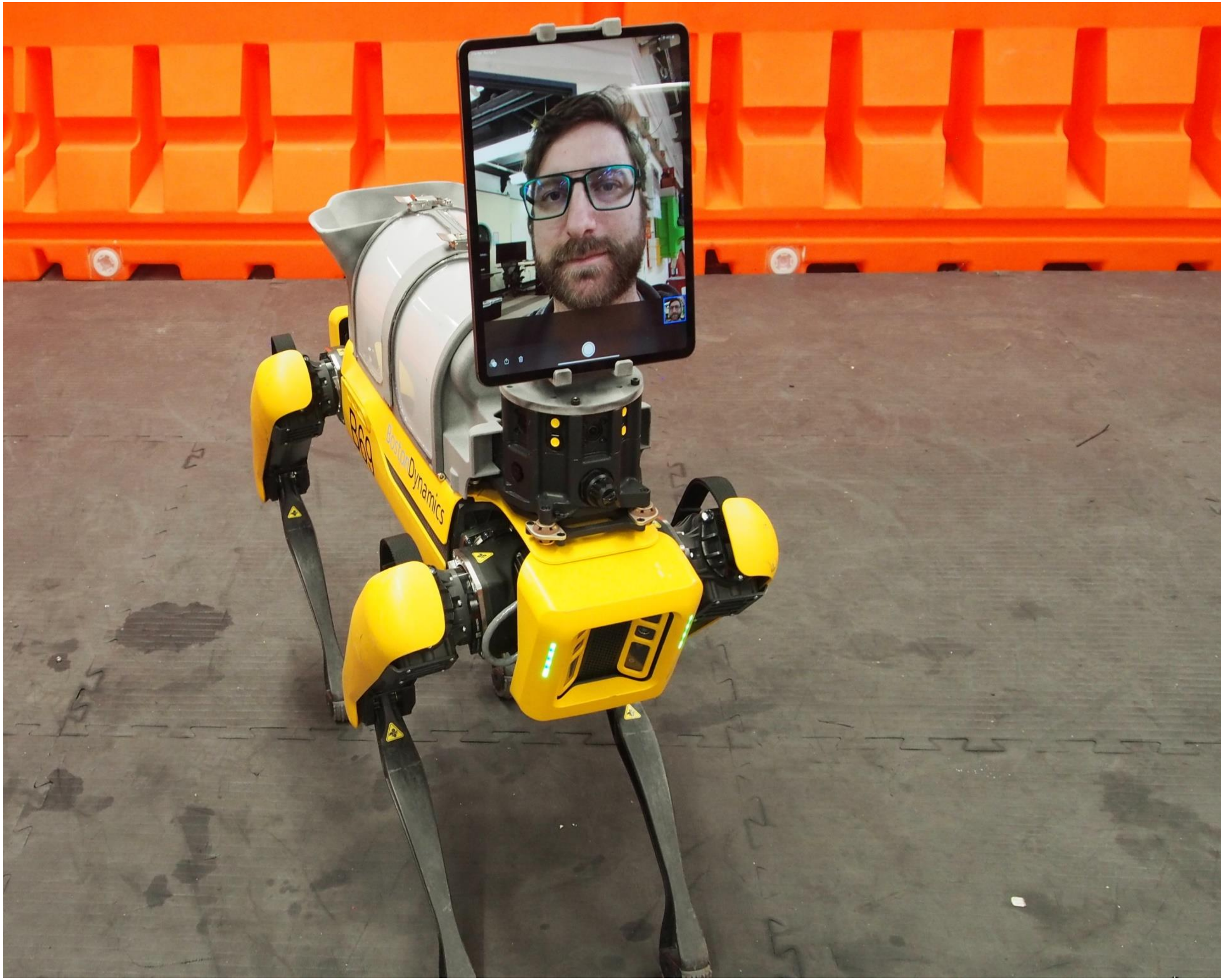
https://www.hansonrobotics.com/sophia/

Sophia's intelligence software is designed by Hanson **Robotics**.

The AI program analyses conversations and extracts data that allows it to improve responses in the future.

Hanson designed **Sophia** to be a suitable companion for the elderly at nursing homes, or to help crowds at large events or parks.




Boston Dynamics

TASK ENVIRONMENT

A task environment refers to the choices, actions and outcomes a given user has for a given task.

Properties of task environments

- Fully observable v/s partially observable
- Single Agent v/s Multi Agent
- Deterministic v/s Stochastic
- Episodic v/s Sequential
- Static v/s Dynamic
- Discrete v/s Continuous
- Known v/s Unknown

FULLY OBSERVABLE VS. PARTIALLY OBSERVABLE

- Agent's **sensors** gives **access** to complete state of the environment at each point in time→ **Fully observable else partially observable.**
  - Easy as there is no need to maintain the internal state to keep track history of the world.
- An agent with no sensors in all environments then such an environment is called as **unobservable**.

- An environment might be **partially observable** because of noisy and inaccurate sensors or because parts of the state are simply missing from the sensor data
- For example
  - a vacuum agent with only a local dirt sensor cannot tell whether there is dirt in other squares,
  - an automated taxi cannot see what other drivers are thinking.

Fully observable (accessible) vs. partially observable (inaccessible):

- Chess – the board is fully observable, as are opponent’s moves.
- Driving – what is around the next bend is not observable (yet).

SINGLE AGENT v/s MULTI AGENT

- If only one agent is **involved** in an environment, and **operating** by itself then such an environment is called **SINGLE AGENT ENVIRONMENT.**
- Multiple agents are operating in an environment- **MULTI-AGENT ENVIRONMENT.**

- For example
  - An agent solving a **crossword puzzle** by itself is clearly in a single-agent environment, whereas an agent playing **chess** is in a two- agent environment.
  - **Chess** is a competitive multi agent environment.
  - **Taxi-driving**- avoiding collisions maximizes the performance measure of all agents, so it is a partially cooperative multi agent environment

DETERMINISTIC v/s STOCHASTIC

- If the **next state of the environment is completely determined by the current state** and the action executed by the agent, then we say the environment is **DETERMINISTIC**
  - otherwise, it is **STOCHASTIC**
  - Most real situations are so complex that it is impossible to keep track of all the unobserved aspects;
    - for practical purposes, they must be treated as stochastic.

EXAMPLE

- **Taxi driving** is clearly **stochastic** in this sense, because one can never predict the **behaviour of traffic** exactly; moreover, one's **tyres blow out** and one's engine seizes up without warning.
- **Vacuum world** as we described it is **deterministic**, but variations can include stochastic elements such as **randomly appearing dirt and an unreliable suction mechanism**

EPISODIC VS. SEQUENTIAL

- E**pisodic task environment**, the agent's experience is divided into atomic episodes.
  - In each episode the agent receives a percept and then performs a single action.
- For example

An agent that has to spot defective parts on an assembly line bases each decision on the current part, regardless of previous decisions; moreover, the current decision doesn't affect whether the next part is defective.

- In **sequential environments,** the current decision could affect all future decisions.
  - **Chess and taxi driving are sequential: in both cases, short-term actions can have long-term consequences.**

Episodic environments are much simpler than sequential environments because the agent does not need to think ahead.

STATIC v/s DYNAMIC

- If the environment can change while an agent is deliberating, then we say the environment is **dynamic** for that agent;
- Otherwise, it is **static.**
- **Static environments** are **easy to deal** with because the agent need not keep looking at the world while it is deciding on an action, nor need it worry about the passage of time.
- **Dynamic environments**, on the other hand, are continuously asking the agent what it wants to do; if it hasn't decided yet, that counts as deciding **to do nothing**!

Playing football- other players make it **dynamic**,

Mowing a lawn is **static**

DISCRETE VS. CONTINUOUS

- The discrete/continuous distinction applies to
  - State of the **environment**,
  - Way **time** is handled, and
  - **Percepts and actions** of the agent.

Example(env , time , percepts & actions)

- The **chess environment** has a **FINITE** number of distinct states (excluding the clock).
  - Chess also has a discrete set of percepts and actions.
- **Taxi driving** is a continuous-state and continuous-time problem:
  - the **speed and location of the taxi** and of the other vehicles sweep through a range of continuous values and do so **smoothly over time**.
  - Taxi-driving actions are also **continuous** (steering angles, etc.).
  - Input from digital cameras is discrete, strictly speaking, but is typically treated as representing continuously varying intensities and locations.

KNOWN VS. UNKNOWN

- Known environment, the **outcomes** for all actions are **given**.
- If the environment is unknown, the agent will have to learn how it works in order to make good decisions.
  - Note that the distinction between known and unknown environments is not the same as the one between fully and partially observable environments

- **Fully/partially observable** –complete of partial state of the environment?
- •**Deterministic/stochastic** –can next state be completely determined by the current state and the action?
- •**Episodic/sequential** –can actions be divided into episodes, and are episodes independent?
- •**Static/dynamic**–does environment changes?
- •**Discrete/continuous** –is there limited number of distinct, well defined percepts and actions?
- •**Single/multi-agent**–does agent operates alone or in collaboration with other agents?

- Non-deterministic environment: physical world: Robot on Mars ,
- Deterministic environment: Tic Tac Toe game
- Episodic environment: Mail sorting system
- Non-episodic environment: Chess game
- Discrete-chess game,
- Continuos – driving a car

Task Environment	Observable	Deterministic	Episodic	Static	Discrete	Agents
Robot soccer	Partially	Stochastic	Sequential	Dynamic	Continuous	Multi
Internet book-shopping	Partially	Deterministic*	Sequential	Static*	Discrete	Single
Autonomous Mars rover	Partially	Stochastic	Sequential	Dynamic	Continuous	Single
Mathematician's assistant	Fully	Deterministic	Sequential	Semi	Discrete	Multi

Task Environment	Observable	Agents	Deterministic	Episodic	Static	Discrete
Crossword puzzle	Fully	Single	Deterministic	Sequential	Static	Discrete
Chess with a clock	Fully	Multi	Deterministic	Sequential	Semi	Discrete
Poker	Partially	Multi	Stochastic	Sequential	Static	Discrete
Backgammon	Fully	Multi	Stochastic	Sequential	Static	Discrete
Taxi driving	Partially	Multi	Stochastic	Sequential	Dynamic	Continuous
Medical diagnosis	Partially	Single	Stochastic	Sequential	Dynamic	Continuous
Image analysis	Fully	Single	Deterministic	Episodic	Semi	Continuous
Part-picking robot	Partially	Single	Stochastic	Episodic	Dynamic	Continuous
Refinery controller	Partially	Single	Stochastic	Sequential	Dynamic	Continuous
Interactive English tutor	Partially	Multi	Stochastic	Sequential	Dynamic	Discrete

Figure 2.6 Examples of task environments and their characteristics.

Four basic kinds of agent programs

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

Learning Agents

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*

Figure 2.7 The TABLE-DRIVEN-AGENT program is invoked for each new percept and returns an action each time. It retains the complete percept sequence in memory.

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*

Figure 2.8 The agent program for a simple reflex agent in the two-state vacuum environment. This program implements the agent function tabulated in Figure 2.3.

SIMPLE REFLEX AGENT

- Simplest kind of agent .
- Agents select actions on the basis of the current percept, ignoring the rest of the percept history.
  - For example, the vacuum agent is a simple reflex agent, because its decision is based only on the current location and on whether that location contains dirt.
  - Ex-Car initiating a brake
- CONDITION–ACTION RULE written as

if car-in-front-is-braking **then** initiate-braking.

A reflex is a quick reaction based on a current situation.

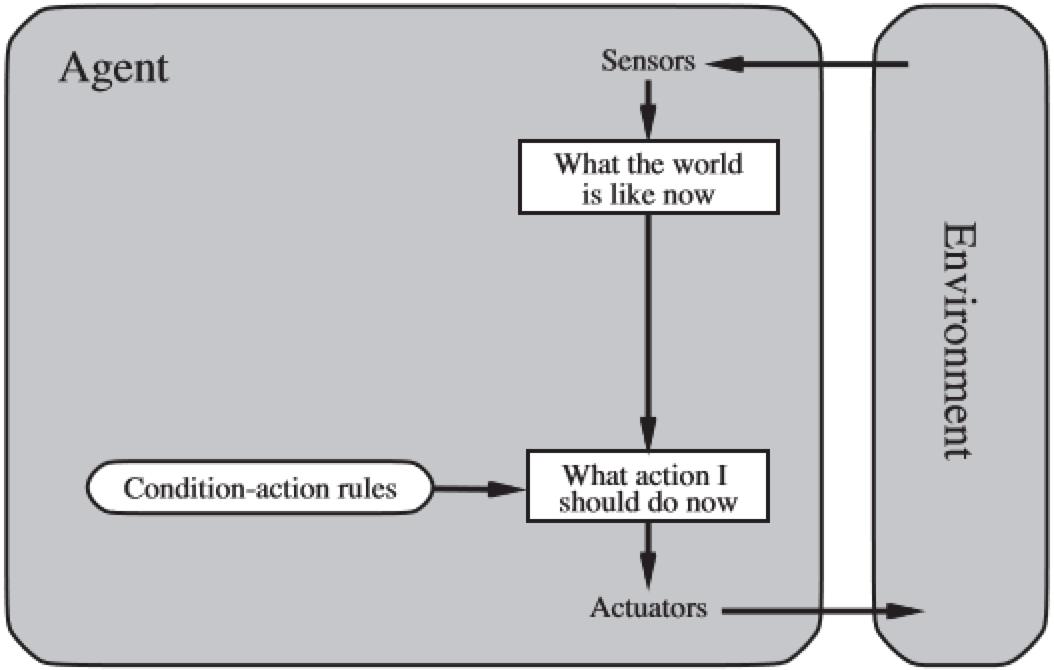


Figure 2.9 Schematic diagram of a simple reflex agent.

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*

Figure 2.10 A simple reflex agent. It acts according to a rule whose condition matches the current state, as defined by the percept.

- **INTERPRET-INPUT** function generates an **abstracted** description of the current state from the percept
- **RULE-MATCH** function **returns** the first rule in the set of rules that matches the given state description.

PROBLEMS WITH SIMPLE REFLEX AGENTS ARE :

- Very limited intelligence.
- No knowledge of non-perceptual parts of state.
- If there occurs any change in the environment, then the collection of rules need to be updated.

MODEL-BASED REFLEX AGENTS

- Most effective way to handle partial observability, agent keeps track of the part of the world it can't see now.
  - Agent should maintain some sort of internal state(depends on the <u>percept history</u> and thereby reflects at least some of the unobserved aspects of the current state)

- Updating INTERNAL STATE INFORMATION requires two kinds of knowledge to be encoded in the agent program.

  First, 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.

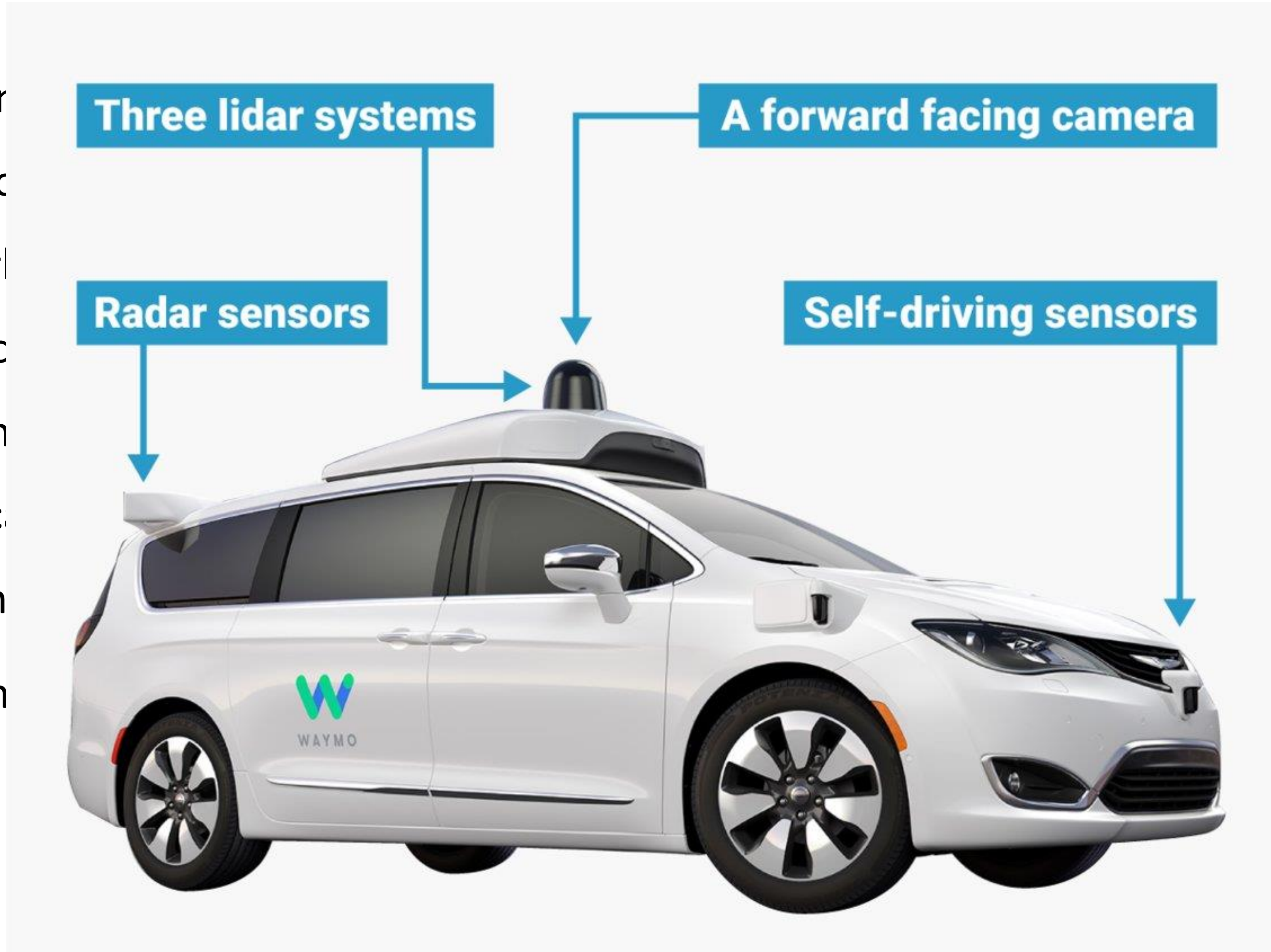
  Second, we need some information about how the agent’s own actions affect the world

  —for example, that when the agent turns the steering wheel clockwise, the car turns to the right, 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..

- Knowledge about “how the world works”—whether implemented in simple Boolean circuits or in complete scientific theories—is called a **Model of the world.**
- An agent that uses such a model is called a **MODEL-BASED AGENT**

Waymo

- Imagi[r] ... to
  somec ... y of
  driverl ... car
  projec ... e in
  Phoen ... tity
  that c ... ake
  action
- Waym ... nt.



COST ? , WAYMO _______ ?

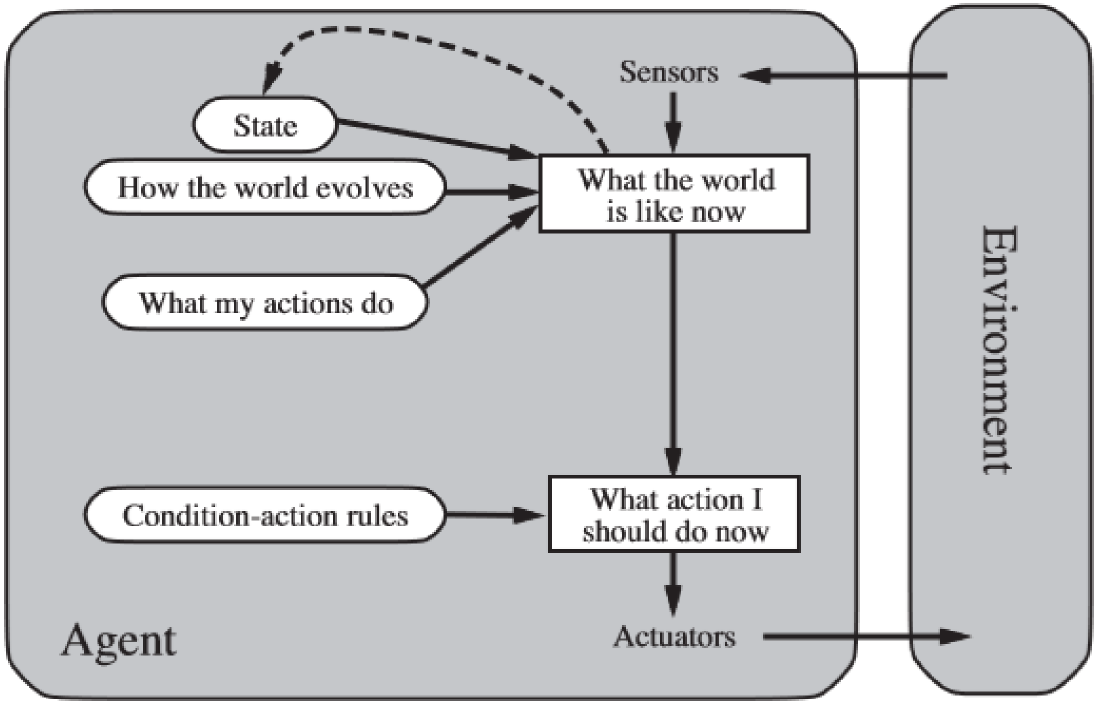


Figure 2.11 A model-based reflex agent.

function MODEL-BASED-REFLEX-AGENT(*percept*) **returns** an action
  persistent: *state*, the agent's current conception of the world state
    model, a description of how the next state depends on current state and action
    rules, a set of condition–action rules
    action, the most recent action, initially none

  state ← UPDATE-STATE(*state*, *action*, *percept*, *model*)
  rule ← RULE-MATCH(*state*, *rules*)
  action ← *rule*.ACTION
  return *action*

- UPDATE-STATE-is responsible for creating the new internal state description
  - Rarely possible for the agent to determine the current state of a partially observable environment exactly.

GOAL BASED AGENTS

"Knowing something 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, turn right, or go straight on. **The correct decision depends on where the taxi is trying to get to.**

GOAL SITUATIONS THAT ARE DESIRABLE

» "What will happen if I do such-and-such?" and "Will that make me happy?"

New Year Resolutions,etc..

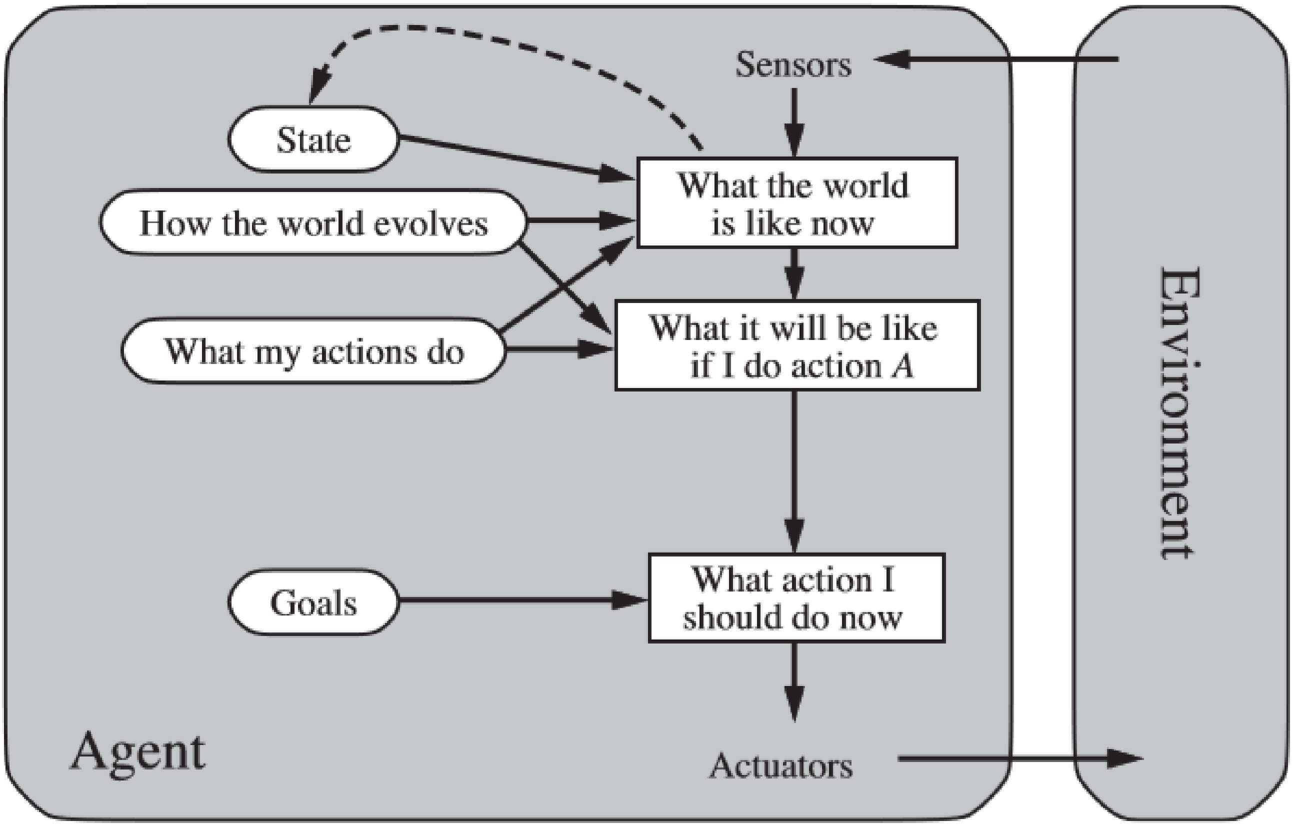


Figure 2.13 A model-based, goal-based agent. It keeps track of the world state as well as a set of goals it is trying to achieve, and chooses an action that will (eventually) lead to the achievement of its goals.

UTILITY-BASED AGENTS

- Goals just provide a crude binary distinction between "happy" and" unhappy" states.
  - For example, many action sequences will get the taxi to its destination (thereby achieving the goal) but some are quicker, safer, more reliable, or cheaper than others.
- A utility-based agent has to model and keep track of its environment, tasks that have involved a great deal of research on **perception, representation, reasoning, and learning.**

A utility based agent looks for the best solution to a scenario.

Ex: Vacation and driving off

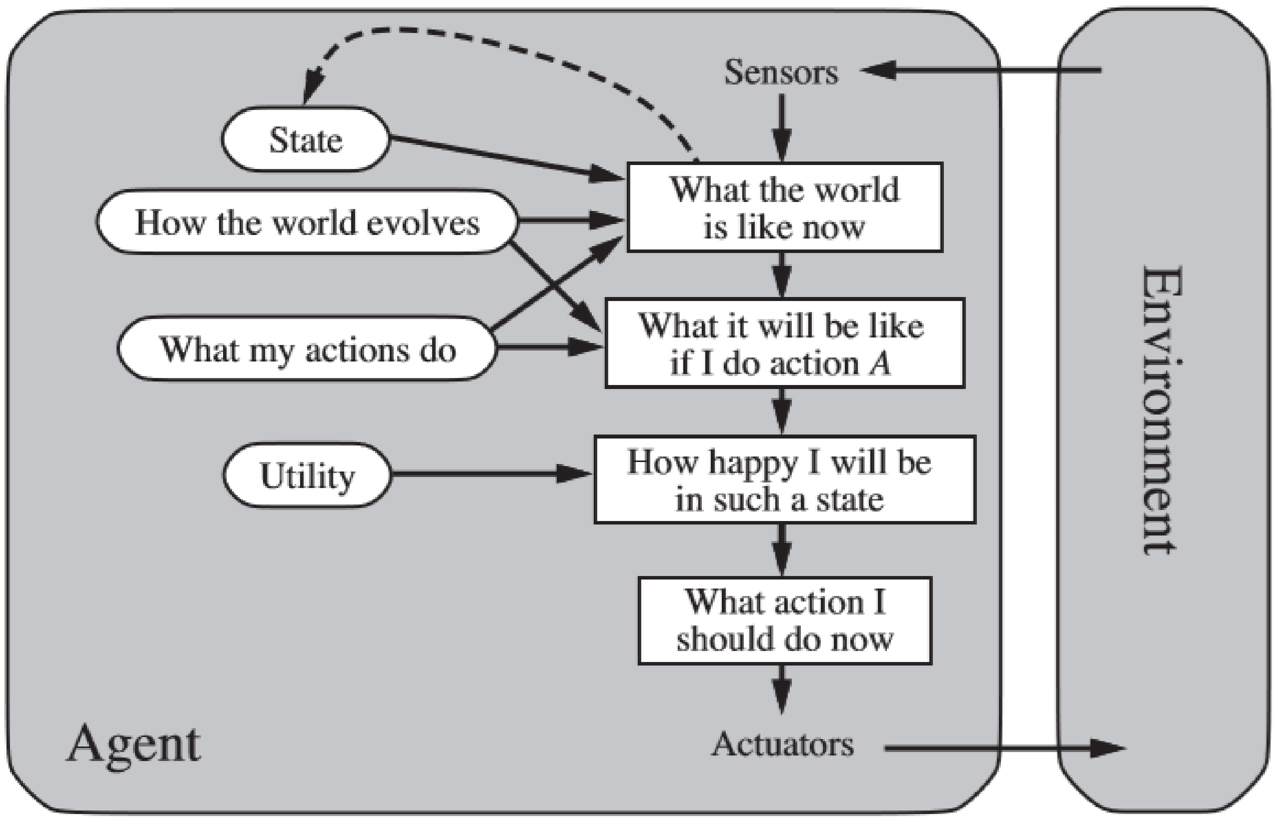


Figure 2.14 A model-based, utility-based agent. It uses a model of the world, along with a utility function that measures its preferences among states of the world. Then it chooses the action that leads to the best expected utility, where expected utility is computed by averaging over all possible outcome states, weighted by the probability of the outcome.

LEARNING AGENTS

A **learning agent** is a tool in **AI** that is capable of **learning** from its experiences.

It starts with some basic knowledge and is then able to act and adapt autonomously, through **learning**, to **improve** its own performance.

Example: **Learning Useful Information**

- Imagine you've decided to start driving for a cab service. It's a Friday night and you're running your typical route from the nightlife scene to many of the hotels away from the downtown area. You've become accustomed to taking the Maple Street exit to pick up your customers, but tonight you opt for Palace Road for a change of pace. To your surprise, you discover that Palace Road is not only quicker, but you can avoid the accident-prone intersection at Maple and Vine. You decide that, in the future, you'll be sure to take Palace Road.
- You probably guessed there's a deeper meaning to the example. It's an illustration of an idea in artificial intelligence (AI) known as a learning agent.

LEARNING AGENT CAN BE DIVIDED INTO FOUR CONCEPTUAL COMPONENTS

1. LEARNING ELEMENT → responsible for making **improvements**

2. PERFORMANCE ELEMENT → ***selecting*** **external actions**.

3. CRITIC → Learning element uses **feedback** from the **CRITIC .**

- **how the performance element should be modified to do better in the future.**

4. PROBLEM GENERATOR→ ***Suggesting*** actions that will lead to new and **informative experiences**

- This is what scientists do when they carry out experiments.

- When you were in school you would do a test and it would be marked the test is the critic. The teacher would mark the test and see what could be improved and instructs you how to do better next time.
- The **teacher** is the **LEARNING ELEMENT** and **you are the PERFORMANCE ELEMENT.**
- **PROBLEM GENERATOR**: - For example coming back to the school analogy, in science with your current knowledge at that time you would not have thought of placing a mass on a spring but the teacher suggested an experiment and you did it and this taught you more and **added to knowledge base.**

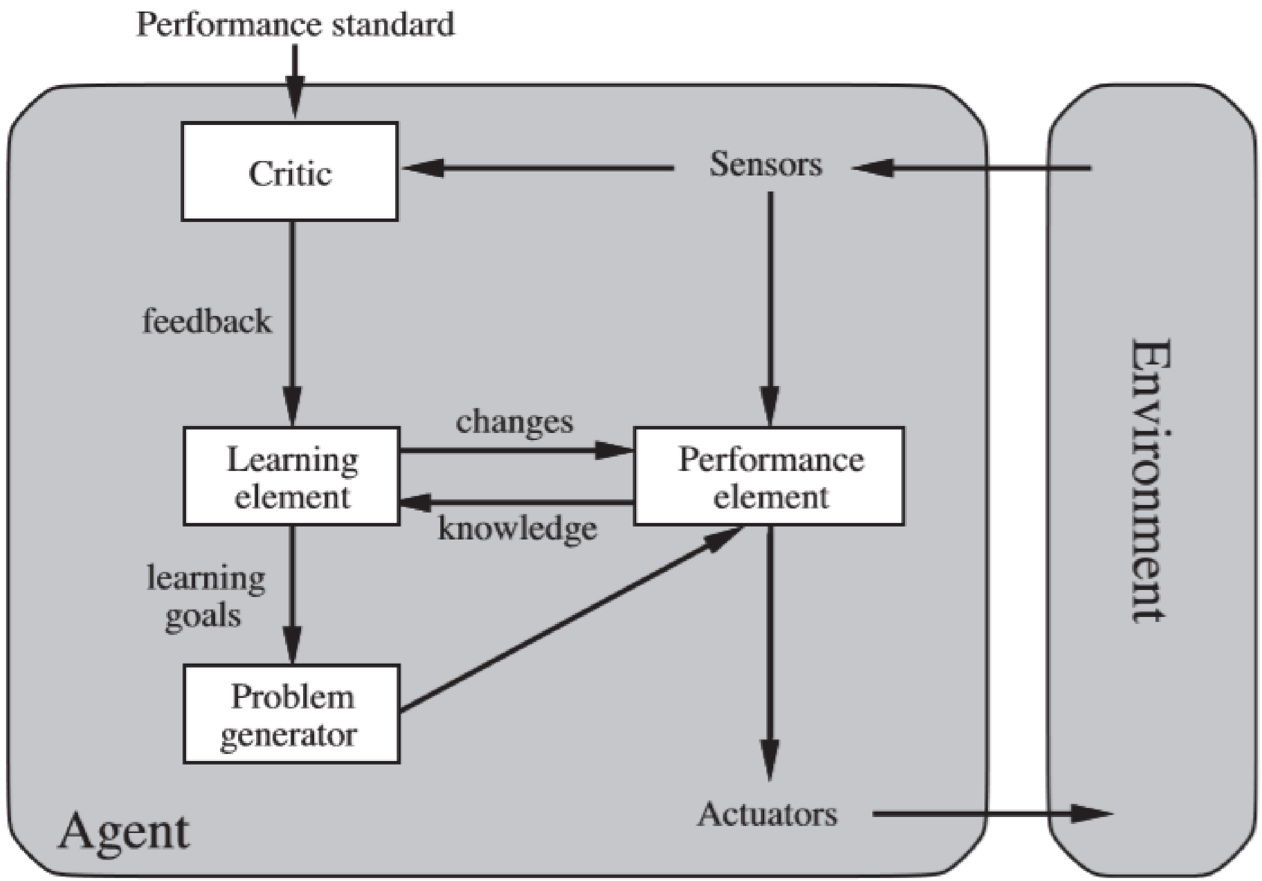


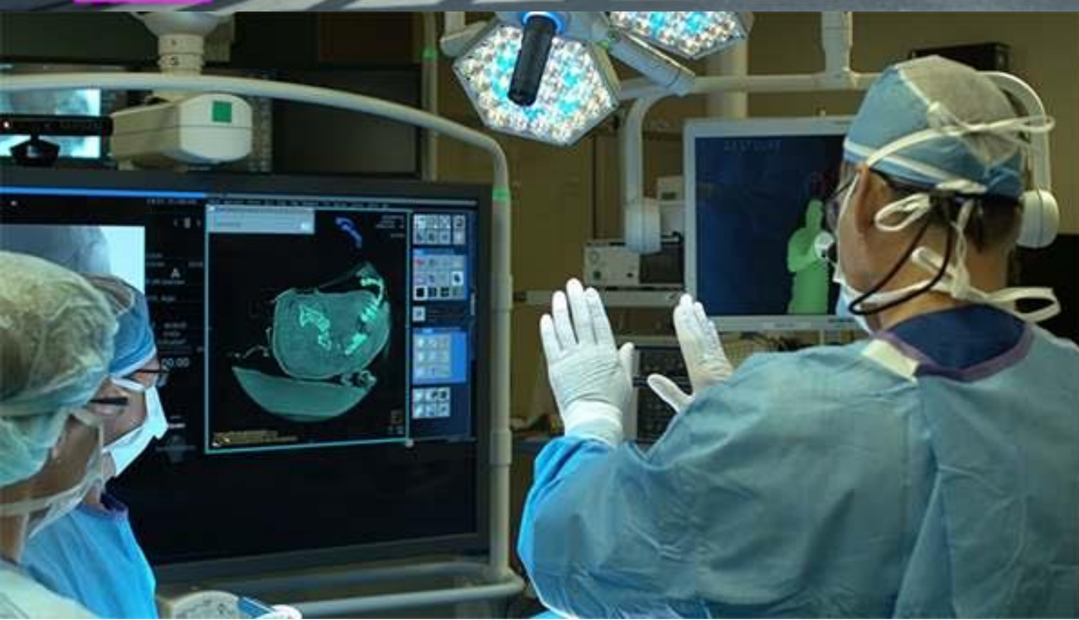
Figure 2.15 A general learning agent.

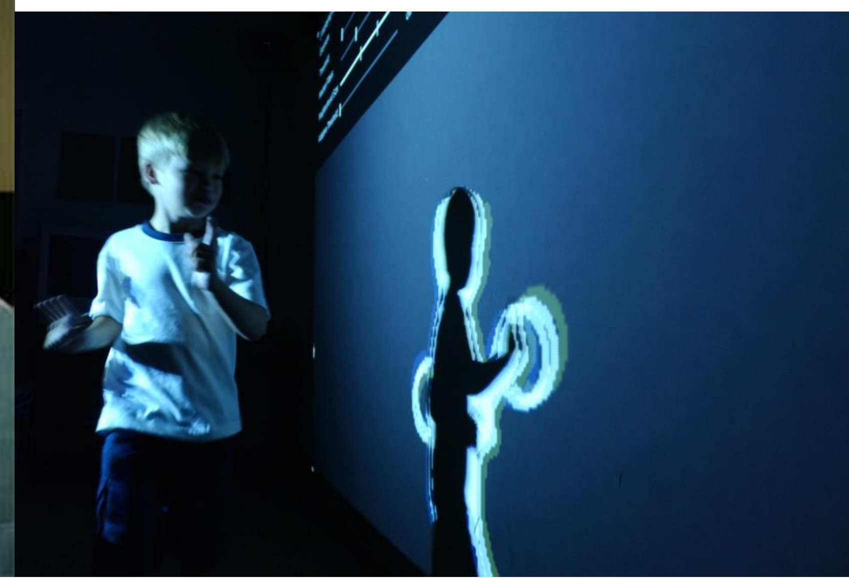
Learning in intelligent agents can be **summarized** as a process of **modification of each component of the agent to bring the components into closer agreement with the available feedback information, thereby improving the overall performance of the agent**.

Human is an example of a learning agent.

For example, a human can learn to ride a bicycle, even though, at birth, no human possesses this skill.


traffic light
traffic light
traffic light
traffic light
traffi
traffic light
traffic light
bus
truck
car
car
car
car
truck
truck
person
perso
person
erson
handbag
perso
person
person





EXAMPLES

PEAS description of the Online shopping agent

PEAS for the “shopping for DataWarehousing books on the internet” activity.

Performance measures:

Price of the book

Author of the book

Quality of the book

Book reviews on google.

Obtain interested/desired books.

Cost minimization.

Environment:

Internet websites.

Web pages of a particular website

Vendors/Sellers

Shippers

Actuators:

Filling in the forms.

Display to the user

Follow URL

Sensors:

Keyboard entry

Browser used to find web pages

HTML

Task Environment

- **1. Observable (Fully or Partial):** This environment is partially observable. When an agent can't determine the complete state of the environment at all points of time, then it is called a partially observable environment.
  - Here, the shopping agent can't see all types of books on one webpage. For example, on the current webpage, all the books have similar ratings and prices. If the user wants to see the books with high ratings then the agent has to follow a different webpage or set the filter in the search bar. Thus, the agent is interacting with a partially observable environment.
- **2. Deterministic or non-deterministic:** The environment is deterministic. A task environment is said to be deterministic if the current state and actions performed in the current state completely determines the next state, otherwise, it will be a non-deterministic task environment.
  - Here, if the shopping agent likes a book and wants to purchase it, then the next state will be followed for the same book. The next stages will be: payment, filling in the delivery address, and order confirmation. The agent will make the payment for the selected book only. Thus, the next state is determined by the current state.
- **3. Episodic/Sequential:** This is a sequential environment. An environment is said to be episodic if it consists of independent episodes and actions performed in one episode don't affect the other episodes. In a sequential environment, the actions performed in the current state will affect the next states.
  - Here, if the current book is rejected by the agent then the agent will not see the same book again. The webpage will not show the same book again, once it is rejected by the agent. Therefore, the action in the current state completely changed the next possible state.

- **4. Static/Dynamic:** It is a static environment. An environment is static if it does not change over time. A car driving environment is dynamic because vehicles are running continuously. The agent doesn’t know what is going to come next. But in the static environment, a particular state is completely unchangeable over time, like a web page.
  - Here, the details of the books or the list of the books displayed on the website is not going to change over time. Details of the book don’t depend on the actions of the agent.
- **5. Discrete/Continuous:** It is a discrete environment. An environment is discrete if it consists of a finite number of states. A chess-playing environment is discrete while the car driving environment is continuous.
  - Here, the number of states is finite. The possible states are:
    - See the book details
    - See the price
    - Fill the form
    - Place the order and make payment.
- **6. Single-agent/Multi-agent**: It is a single-agent system. Only one agent is interacting with the environment and no other robots or AI agent is present in the environment. An environment is said to be a single agent environment if only one agent is interacting and acting on it, otherwise multi-agent. A chess-playing environment is multi-agent since two agents (human or robot) are required to play the chess game.
  - Here, the shopping agent alone is acting on the website.

Describe the PEAS for the “bidding on an item at an auction” activity.

PEAS

- **Performance measures:**
  - Cost of the item
  - Quality of the item
  - Value of the item
  - Necessity of the item
- **Environment:**
  - Auctioneer
  - Bidders
  - Bidders Items which are to be bid
- **Actuators: (means to perform the activity)**
  - Speakers
  - Microphones
  - Display items
  - Budget
- **Sensors: (means to perceive the environment)**
  - Camera
  - Price monitor, where prices are being displayed.
  - Eyes
  - Ears of the attendees.

TASK ENV

- **Observable (Fully/Partially):** It is a partially observable environment. When an agent can't determine the complete state of the environment at all points of time, then it is called a partially observable environment. Here, the auctioneering agent is not capable of knowing the state of the environment fully at all points in time. We can say that wherever the agent has to deal with humans in the task environment, it can't observe the state fully.
- **Agents (Single/Multi):** It is single-agent activity. Because only one agent is involved in this environment and is operating by itself. There are other human agents involved in the activity but they all are passing their percept sequence to the central agent – our auction agent. So, it is still a single-agent environment.
- **Deterministic (Deterministic/Stochastic):** It is stochastic activity. Because in bidding the outcome can't be determined base on a specific state of the agent. It is the process where the outcome involves some randomness and has some uncertainty

- **Episodic (Episodic/Sequential):** It is a sequential task environment. In the episodic environment, the episodes are independent of each other. The action performed in one episode doesn't affect subsequent episodes. Here in auction activity, if one bidder set the value X then the next bidder can't set the lesser value than X. So, the episodes are not independent here. Therefore, it is a sequential activity. There is high uncertainty in the environment.
- **Static (Static/Semi/Dynamic):** It is a dynamic activity. The static activity is the one in which one particular state of the environment doesn't change over time. But here in the auction activity, the states are highly subjective to the change. A static environment is the crossword solving problem where numbers don't change.
- **Discrete (Discrete/Continuous):** It is a continuous activity. The discrete environment is one that has a finite number of states. But here in auction activity, bidders can set the value forever. The number of states can be 1 or 1000. There is randomness in the environment. Thus, it is a continuous environment.