Module 1

Introduction to Artificial Intelligence

- Homo Sapiens: The name is Latin for "*wise man*"
- Philosophy of AI - "*Can a machine think and behave like humans do?*"
- In Simple Words - *Artificial Intelligence is a way of **making a computer, a computer-controlled robot, or a software think intelligently**, in the similar manner the intelligent humans think.*
- **Artificial intelligence** (**AI**) is an area of computer science that emphasizes the creation of **intelligent** machines that work and react like humans.
- AI is accomplished by studying how the human brain thinks and how humans learn, decide, and work while trying to solve a problem, and then using the outcomes of this study as a basis of developing intelligent software and systems.

1. What is AI ?

Views of AI fall into four categories:

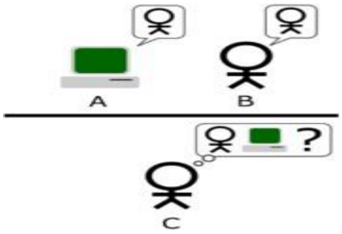
i. Thinking humanly
ii. Thinking rationally
iii. Acting humanly
iv. **Acting rationally**

Thinking Humanly "The exciting new effort to make comput- ers think . .. *machines with minds*, in the full and literal sense." (Haugeland, 1985) "[The automation of] activities that we associate with human thinking, activities such as decision-making, problem solv- ing, learning .. ." (Bellman, 1978)	**Thinking Rationally** "The study of mental faculties through the use of computational models." (Charniak and McDermott, 1985) "The study of the computations that make it possible to perceive, reason, and act." (Winston, 1992)
Acting Humanly "The art of creating machines that per- form functions that require intelligence when performed by people." (Kurzweil, 1990) "The study of how to make computers do things at which, at the moment, people are better." (Rich and Knight, 1991)	**Acting Rationally** "Computational Intelligence is the study of the design of intelligent agents." (Poole *et al.*, 1998) "AI . . . is concerned with intelligent be- havior in artifacts." (Nilsson, 1998)

Figure 1.1 Some definitions of artificial intelligence, organized into four categories.

i. Acting humanly: The Turing Test approach

- Turing (1950) developed "Computing machinery and intelligence":
- "Can machines think?" or "Can machines behave intelligently?"
- Operational test for intelligent behavior: the Imitation Game
- A computer passes the test if a human interrogator, after posing some written questions, cannot tell whether the written responses come from a person or from a machine.
- Suggested major components of AI: knowledge, reasoning, language understanding, learning.



The computer would need to possess the following capabilities:

- Natural Language Processing: To enable it to communicate successfully in English.
- Knowledge representation: To store what it knows or hears.
- Automated reasoning: To use the stored information to answer questions and to draw new conclusions.
- Machine Learning: To adapt to new circumstances and to detect and extrapolate patterns.

To pass the Total Turing Test

- Computer vision: To perceive objects.
- Robotics: To manipulate objects and move about.

ii. Thinking humanly: The cognitive modeling approach

- If we are going to say that *a given program thinks like a human,* we must have some way of determining how humans think.
- We need to get inside the actual working of human minds.
- There are 3 ways to do it:
    - i. Through introspection
      Trying to catch our own thoughts as they go
    - ii. Through psychological experiments
      Observing a person in action
    - iii. Through brain imaging
      Observing the brain in action
- Comparison of the trace of computer program reasoning steps to traces of human subjects solving the same problem.
- Cognitive Science brings together computer models from AI and experimental techniques from psychology to try to construct precise and testable theories of the working of the human mind.
- Now distinct from AI
    - AI and Cognitive Science fertilize each other in the areas of vision and natural language.
- Once we have a sufficiently precise theory of the mind, it becomes possible to express the theory as a computer program.
- If the program's input-output behaviour matches corresponding human behaviour, that is evidence that the program's mechanisms could also be working in humans.
- For example, Allen Newell and Herbert Simon, who developed GPS, the "General Problem Solver".

iii. Thinking rationally: The "laws of thought" approach

Aristotle was one of the first to attempt to codify ―right thinking,‖ that is, irrefutable reasoning processes. His **syllogisms** provided patterns for argument structures that always yielded correct conclusions when given correct premises.

Eg.

Socratesis a man;

All men are mortal;

Therefore, Socrates is mortal.-- logic

There are two main obstacles to this approach.

1. It is not easy to take informalknowledge and state it in the formal terms required by logical notation, particularly whenthe knowledge is less than 100% certain.
2. Second, there is a big difference between solvinga problem ―in principle and solving it in practice.

iv. Acting rationally: The rational agent approach

- An **agent** is just something that acts.
- All computer programs do something, but computer agents are expected to do more: operate autonomously, perceive their environment, persist over a prolonged time period, and adapt to change, and create and pursue goals.
- A **rational agent** is one that acts so as to achieve the best outcome or, when there is uncertainty, the best expected outcome.
- In the ―laws of thought‖ approach to AI, the emphasis was on correct inferences.
- On the other hand, correct inference is not all of rationality; in some situations, there is no provably correct thing to do, but something must still be done.
- For example, recoiling from a hot stove is a reflex action that is usually more successful than a slower action taken after careful deliberation.

- **What means "behave rationally" for a person/system:**
  - Take the right/ best action to achieve the goals, based on his/its knowledge and belief
  - Example: Assume I don't like to get wet in rain (my goal), so I bring an umbrella (my action). Do I behave rationally?
  - The answer is dependent on my knowledge and belief.
  - If I've heard the forecast for rain and I believe it, then bringing the umbrella is rational.
  - If I've not heard the forecast for rain and I do not believe that it is going to rain, then bringing the umbrella is not rational.
- **"Behave rationally" does not always achieve the goals successfully**

Example:
  - My goals – (i) do not get wet if rain; (ii) do not looked stupid (such as bring an umbrella when not raining)
  - My knowledge/belief – weather forecast for rain and I believe it
  - My rational behaviour – bring an umbrella
  - The outcome of my behaviour: If rain, then my rational behaviour achieves both goals; If no rain, then my rational behaviour fails to achieve the 2nd goal
- The successfulness of "behave rationally" is limited by my knowledge and belief.

2. Foundations of Artificial Intelligence

Philosophy

- Can formal rules be used to draw valid conclusions?
- How does the mind arise from a physical brain? Where does knowledge come from?
- How does knowledge lead to action?
- Aristotle was the first to formulate a precise set of laws governing the **rational part of the mind**. He developed an informal system of **syllogisms** for **proper reasoning**, which in principle allowed one to generate **conclusions** mechanically, given initial **premises**.
  - *All dogs are animals;*

- *all animals have four legs;*
- *therefore all dogs have four legs*

- Descartes was a strong advocate of the **power of reasoning** in understanding the world, philosophy now called as **rationalism**.

Mathematics

- What are the formal rules to draw valid conclusions? What can be computed?
- How do we reason with uncertain information?
- **Formal representation and proof algorithms:** Propositional logic **Computation:** Turing tried to characterize exactly which functions are computable - capable of being computable.
- **(un)decidability: Incompleteness theory** showed that in any formal theory, there are **true statements that are undecidable** i.e. they have no proof within the theory.
  - **"** *a line can be extended infinitely in both directions* **"**
- **(in)tractability:** A problem is called intractable if the time required to solve instances of the problem grows exponentially with the size of the instance.
- **probability:** Predicting the future.

Economics

- How should we make decisions so as to maximize payoff?
- Economics is the study of how people make choices that lead to **preferred outcomes**(utility).
- **Decision theory:** It combines **probability theory** with **utility theory**, provides a formal and complete framework for decisions made under uncertainty.

Neuroscience

- How do brains process information?
- Neuroscience is the study of the **nervous system**, particularly brain.
- Brain consists of nerve cells or **neurons**. 10^11 neurons.
- Neurons are considered as **Computational units**.

Psychology

- Behaviorism movement, led by John Watson(1878-1958). Behaviorists insisted on studying only objective measures of the percepts(stimulus) given to an animal and its resulting actions(or response). Behaviorism discovered a lot about rats and pigeons but had less success at understanding humans.
- Cognitive psychology views the brain as an information processing device. A common view among psychologists is that cognitive theory should be like a computer program. (Anderson 1980) i.e. It should describe a detailed information processing mechanism whereby some cognitive function might be implemented.

Computer engineering:

How can we build an efficient computer?

- For artificial intelligence to succeed, we need two things: intelligence and an artifact. The computer has been the artifact(object) of choice.
- The first operational computer was the electromechanical Heath Robinson, built in 1940 by Alan Turing's team for a single purpose: deciphering German messages.
- The first operational programmable computer was the Z-3, the invention of KonradZuse in Germany in 1941.
- The first electronic computer, the ABC, was assembled by John Atanasoff and his student Clifford Berry between 1940 and 1942 at Iowa State University.

- The first programmable machine was a loom, devised in 1805 by Joseph Marie Jacquard (1752-1834) that used punched cards to store instructions for the pattern to be woven.

Control theory and cybernetics

How can artifacts operate under their own control?

- Ktesibios of Alexandria (c. 250 B.C.) built the first self-controlling machine: a water clock with a regulator that maintained a constant flow rate. This invention changed the definition of what an artifact could do.
- Modern control theory, especially the branch known as stochastic optimal control, has as its goal the design of systems that maximize an objective function over time. This roughly OBJECTIVE FUNCTION matches our view of Al: designing systems that behave optimally.
- Calculus and matrix algebra- the tools of control theory
- The tools of logical inference and computation allowed AI researchers to consider problems such as language, vision, and planning that fell completely outside the control theorist's purview.

Linguistics

How does language relate to thought?

- In 1957, B. F. Skinner published Verbal Behaviour. This was a comprehensive, detailed account of the behaviourist approach to language learning, written by the foremost expert in the field.
- Noam Chomsky, who had just published a book on his own theory, Syntactic Structures. Chomsky pointed out that the behaviourist theory did not address the notion of creativity in language.
- Modern linguistics and AI were ―born‖ at about the same time, and grew up together, intersecting in a hybrid field called computational linguistics or natural language processing.
- The problem of understanding language soon turned out to be considerably more complex than it seemed in 1957. Understanding language requires an understanding of the subject matter and context, not just an understanding of the structure of sentences.
- Knowledge representation (the study of how to put knowledge into a form that a computer can reason with)- tied to language and informed by research in linguistics.

3. History of Artificial Intelligence

1. The gestation of artificial intelligence (1943–1955)

   The gestation of artificial intelligence (AI) during the period from 1943 to 1955 marked the early theoretical and conceptual groundwork for the field. This period laid the foundation for the subsequent development of AI.

2. The birth of artificial intelligence (1956)

   The birth of artificial intelligence (AI) in 1956 is commonly associated with the Dartmouth Conference, a seminal event that took place at Dartmouth College in Hanover, New Hampshire.

3. Early enthusiasm, great expectations (1952–1969)

   The period from 1952 to 1969 in the history of artificial intelligence (AI) was characterized by early enthusiasm and great expectations. Researchers during this time were optimistic about the potential of AI and believed that significant progress could be made in creating machines with human-like intelligence.

4. A dose of reality (1966–1973)

   The period from 1966 to 1973 in the history of artificial intelligence (AI) is often referred to as "A Dose of Reality." During this time, researchers faced challenges and setbacks that led to a reevaluation of the initial optimism and expectations surrounding AI.

5. Knowledge-based systems: The key to power? (1969–1979)

   The period from 1969 to 1979 in the history of artificial intelligence (AI) is characterized by a focus on knowledge-based systems, with researchers exploring the use of symbolic representation of knowledge to address challenges in AI. This era saw efforts to build expert systems, which were designed to emulate human expertise in specific domains.

6. AI becomes an industry (1980–present)

   The period from 1980 to the present marks the evolution of artificial intelligence (AI) into an industry, witnessing significant advancements, increased commercialization, and widespread applications across various domains.

7. The return of neural networks (1986–present)

   The period from 1986 to the present is characterized by the resurgence and dominance of neural networks in the field of artificial intelligence (AI). This era is marked by significant advancements in the development of neural network architectures, training algorithms, and the widespread adoption of deep learning techniques.

8. AI adopts the scientific method (1987–present)

   The period from 1987 to the present has seen the adoption of the scientific method in the field of artificial intelligence (AI), reflecting a more rigorous and empirical approach to research. This shift has involved the application of experimental methodologies, reproducibility, and a greater emphasis on evidence-based practices.

9. The emergence of intelligent agents (1995–present)

   The period from 1995 to the present has been marked by the emergence and evolution of intelligent agents in the field of artificial intelligence (AI). Intelligent agents are autonomous entities that perceive their environment, make decisions, and take actions to achieve goals.

10. The availability of very large data sets (2001–present)

    The period from 2001 to the present has been characterized by the availability and utilization of very large datasets in the field of artificial intelligence (AI). This era has witnessed an unprecedented growth in the volume and diversity of data, providing a foundation for training and enhancing increasingly sophisticated AI models.

Intelligent Agents

1. Agents and environment

An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators. simple idea is illustrated in Figure 2.1

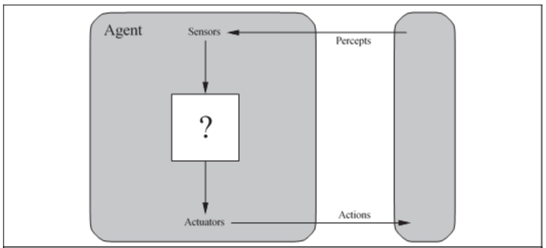


Figure 2.1 Agents interact with environments through sensors and actuators.

Percept − It is agent's perceptual inputs at a given instance.
Percept Sequence − It is the history of all that an agent has perceived till date.
Agent Function − It is a map from the precept sequence to an action.
Performance Measure of Agent − It is the criteria which determines how successful an agent is.
Behavior of Agent − It is the action that agent performs after any given sequence of percepts.

The vacuum-cleaner world shown in Figure 2.2.

This particular world has just two locations: squares *A* and *B*. The vacuum agent perceives which square it is in and whether there is dirt in the square. It can choose to move left, move right, suck up the dirt, or do nothing. One very simple agent function is the following: if the current square is dirty, then suck; otherwise, move to the other square. A partial tabulation of this agent function is shown in Figure 2.3 and an agent program that implements is given in Figure 2.8.

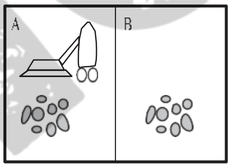


Figure 2.2 A vacuum-cleaner world with just two locations.

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.

```
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.

2. Concept of Rationality

A rational agent is one that does the right thing—conceptually speaking, every entry in the table for the agent function is filled out correctly.

Rationality

Rational at any given time depends on four 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.

A definition **of a rational agent:** *For each possible percept sequence, a rational agent should select an action that is ex-pected to maximize its performance measure, given the evidence provided by the percept sequence and whatever built-in knowledge the agent has.*

Consider the simple vacuum-cleaner agent that cleans a square if it is dirty and moves to the other square if not; this is the agent function tabulated in Figure 2.3. Is this a rational agent? That depends! First, we need to say what the performance measure is, what is known about the environment, and what sensors and actuators the agent has. Let us assume the following:

- The performance measure awards one point for each clean square at each time step, over a "lifetime" of 1000-time steps.
- The "geography" of the environment is known *a priori* (Figure 2.2) but the dirt distribution and the initial location of the agent are not. Clean squares stay clean and sucking cleans the current square. The *Left* and *Right* actions move the agent left and right except when this would take the agent outside the environment, in which case the agent remains where it is.
- The only available actions are *Left*, *Right*, and *Suck*.
- The agent correctly perceives its location and whether that location contains dirt.

We claim that *under these circumstances* the agent is indeed rational.

3. The nature of environment

Task environments, which are essentially the "problems" to which rational agents are the "solutions."

To specify the performance measure, the environment, and the agent's actuators and sensors called the **PEAS** (**P**erformance, **E**nvironment, **A**ctuators, **S**ensors) description.

In designing an agent, the first step must always be to specify the task environment as fully as possible.

PEAS description of an automated taxi driver.

Agent Type	Performance Measure	Environment	Actuators	Sensors
Taxi driver	Safe, fast, legal, comfortable trip, maximize profits	Roads, other traffic, pedestrians, customers	Steering, accelerator, brake, signal, horn, display	Cameras, sonar, speedometer, GPS, odometer, accelerometer, engine sensors, keyboard

Figure 2.4 PEAS description of the task environment for an automated taxi.

The **performance measure** to which we would like our automated driver to aspire? Desirable qualities include getting to the correct destination; minimizing fuel consumption and wear and tear; minimizing the trip time or cost; minimizing violations of traffic laws and disturbances to other drivers; maximizing safety and passenger comfort; maximizing profits. Obviously, some of these goals conflict, so tradeoffs will be required.

What is the driving environment that the taxi will face? Any taxi driver must deal with a variety of roads, ranging from rural lanes and urban alleys to 12-lane freeways. The roads contain other traffic, pedestrians, stray animals, road works, police cars, puddles, and potholes. The taxi must also interact with potential and actual passengers.

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.

4. Properties of Task Environments:

Fully observable vs. partially observable

- A task environment is (effectively) fully observable iff the sensors detect the complete state of the environment
  - ”relevant” depends on the performance measure
  - no need to maintain internal state to keep track of the environment
- A task environment may be partially observable (Ex: Taxi driving):
  - noisy and inaccurate sensors
  - parts of the state are not accessible for sensors
- A task environment might be even unobservable (no sensors)

- e.g. fully-deterministic actions

Deterministic vs. stochastic

- A task environment is deterministic iff its next state is completely determined by its current state and by the action of the agent. (Ex: a crossword puzzle).
- If not so:
  - A task environment is stochastic if uncertainty about outcomes is quantified in terms of probabilities (Ex: dice, poker game, component failure,...)
  - A task environment is nondeterministic iff actions are characterized by their possible outcomes, but no probabilities are attached to them.

In a multi-agent environment, we ignore uncertainty that arises from the actions of other agents (Ex: chess is deterministic even though each agent is unable to predict the actions of the others).

A partially observable environment could appear to be stochastic. =⇒ for practical purposes, when it is impossible to keep track of all the unobserved aspects, they must be treated as stochastic. (Ex: Taxi driving).

Episodic vs. sequential

In an episodic 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

In episodes do not depend on the actions taken in previous episodes, and they do not influence future episodes.

- Ex: an agent that has to spot defective parts on an assembly line,

In sequential environments the current decision could affect future decisions ⇒ actions can have long-term consequences

- Ex: chess, taxi driving, ...

Episodic environments are much simpler than sequential ones

- No need to think ahead!

Static vs. dynamic

The task environment is dynamic iff it can change while the agent is choosing an action, static otherwise ⇒ agent needs keep looking at the world while deciding an action

- Ex: crossword puzzles are static, taxi driving is dynamic

The task environment is semidynamic if the environment itself does not change with time, but the agent's performance score does.

- Ex: chess with a clock

Static environments are easier to deal wrt. [semi]dynamic ones.

Discrete vs. continuous

The state of the environment, the way time is handled, and agents percepts & actions can be discrete or continuous

- Ex: Crossword puzzles: discrete state, time, percepts & actions
- Ex: Taxi driving: continuous state, time, percepts & actions

Note:

- The simplest environment is fully observable, single-agent, deterministic, episodic, static and discrete. Ex: simple vacuum cleaner

- Most real-world situations are partially observable, multi-agent, stochastic, sequential, dynamic, and continuous. Ex: taxi driving

Example properties of task Environments:

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

Properties of the Agent's State of Knowledge

Known vs. unknown

- Describes the agent's (or designer's) state of knowledge about the "laws of physics" of the environment
  - if the environment is known, then the outcomes (or outcome probabilities if stochastic) for all actions are given.
  - if the environment is unknown, then the agent will have to learn how it works in order to make good decisions
- Orthogonal wrt. task-environment properties.

Known not equal to Fully observable

- a known environment can be partially observable (Ex: a solitaire card games)
- an unknown environment can be fully observable (Ex: a game I don't know the rules of)

5. The structure of agents

Agent = Architecture + Program

- AI Job: design an agent program implementing the agent function
- The agent program runs on some computing device with physical sensors and actuators: the agent architecture
- All agents have the same skeleton:
  - Input: current percepts
  - Output: action
  - Program: manipulates input to produce output.

- The agent function takes the entire percept history as input
- The agent program takes only the current percept as input.
- if the actions need to depend on the entire percept sequence, the agent will have to remember the percepts

The Table-Driven Agent

The table represents explicitly the agent function Ex: the simple vacuum cleaner

```
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
```

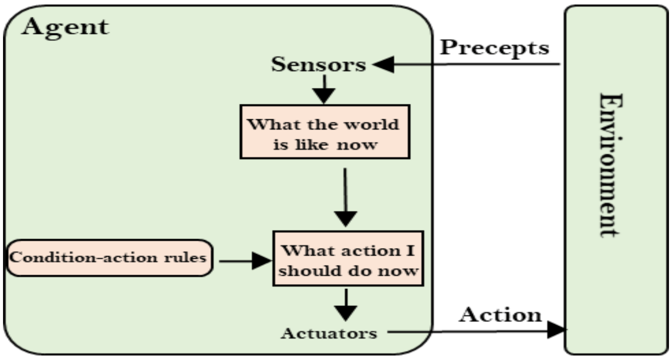
Figure 2.7 The TABLE-DRIVEN-AGENT program is invoked for each new percept and returns an action each time. It retains the complete percept sequence in memory.

Agents can be grouped into five classes based on their degree of perceived intelligence and capability. All these agents can improve their performance and generate better action over the time. These are given below:

- Simple Reflex Agent
- Model-based reflex agent
- Goal-based agents
- Utility-based agent
- Learning agent

Simple reflex agents

- The Simple reflex agents are the simplest agents. These agents take decisions on the basis of the current percepts and ignore the rest of the percept history.
- These agents only succeed in the fully observable environment.
- The Simple reflex agent does not consider any part of percepts history during their decision and action process.
- The Simple reflex agent works on Condition-action rule, which means it maps the current state to action. Such as a Room Cleaner agent, it works only if there is dirt in the room.
- Problems for the simple reflex agent design approach:
  - They have very limited intelligence
  - They do not have knowledge of non-perceptual parts of the current state
  - Mostly too big to generate and to store.
  - Not adaptive to changes in the environment.



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.

Model-based reflex agent

The Model-based agent can work in a partially observable environment, and track the situation.

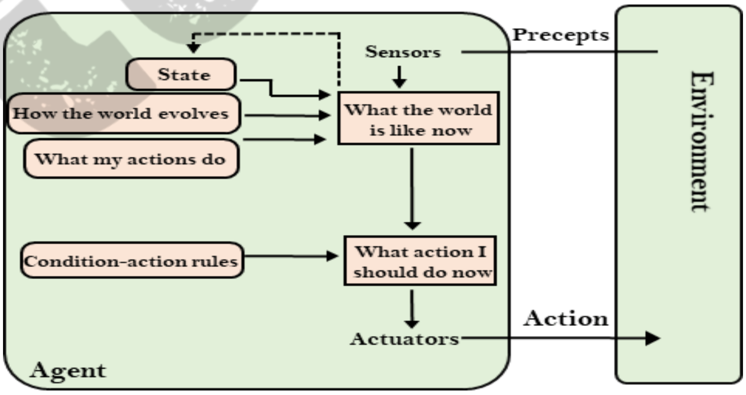
A model-based agent has two important factors:

- **Model:** It is knowledge about "how things happen in the world," so it is called a Model-based agent.
- **Internal State:** It is a representation of the current state based on percept history.

These agents have the model, "which is knowledge of the world" and based on the model they perform actions.

Updating the agent state requires information about:

- How the world evolves
- How the agent's action affects the world.



```
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
```

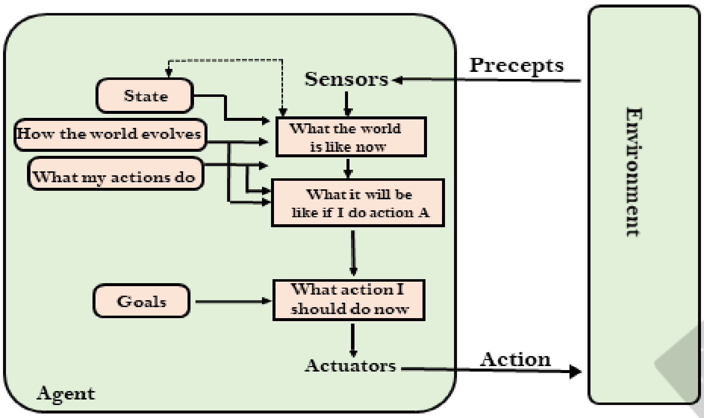
Figure 2.12 A model-based reflex agent. It keeps track of the current state of the world, using an internal model. It then chooses an action in the same way as the reflex agent.

- For the braking problem, the internal state is not too extensive— just the previous frame from the camera, allowing the agent to detect when two red lights at the edge of the vehicle go on or off simultaneously.
- For other driving tasks such as changing lanes, the agent needs to keep track of where the other cars are if it can't see them all at once. And for any driving to be possible at all, the agent needs to keep track of where its keys are.
- Updating this internal state information as time goes by 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.
- This 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.**

Goal-based agents

- The knowledge of the current state environment is not always sufficient to decide for an agent to what to do.
- The agent needs to know its goal which describes desirable situations.
- Goal-based agents expand the capabilities of the model-based agent by having the "goal" information.
- They choose an action, so that they can achieve the goal.
- These agents may have to consider a long sequence of possible actions before deciding whether the goal is achieved or not. Such considerations of different scenario are called searching and planning, which makes an agent proactive.
- **Sometimes goal-based action selection is straightforward:** for example when goal satisfaction results immediately from a single action.
- **Sometimes it will be trickier:** for example, when the agent has to consider long sequences of twists and turns to find a way to achieve the goal.

- **Search and planning** are the subfields of AI devoted to finding action sequences that achieve the agent's goals.



Reflex Agent	**Goal Based**
For the reflex agent, on the other hand, we would have to rewrite many condition–action rules.	The goal-based agent appears less efficient, it is more flexible because the knowledge that supports its decisions is represented explicitly and can be modified. If it starts to rain, the agent can update its knowledge of how effectively its brakes will operate; this will automatically cause all of the relevant behaviors to be altered to suit the new conditions
The reflex agent's rules for when to turn and when to go straight will work only for a single destination; they must all be replaced to go somewhere new.	The goal-based agent's behavior can easily be changed to go to a different destination, simply by specifying that destination as the goal.
Example: The reflex agent brakes when it sees brake lights	Example: A goal-based agent, in principle, could reason that if the car in front has its brake lights on, it will slow down.

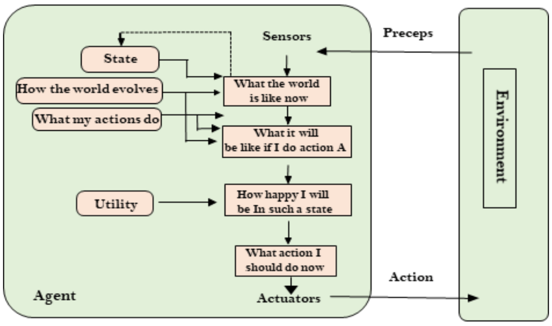
Utility-based agents

- 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 advantages wrt. goal-based:

- with conflicting goals, utility specifies and appropriate tradeoff
- with several goals none of which can be achieved with certainty, utility selects proper tradeoff between importance of goals and likelihood of success

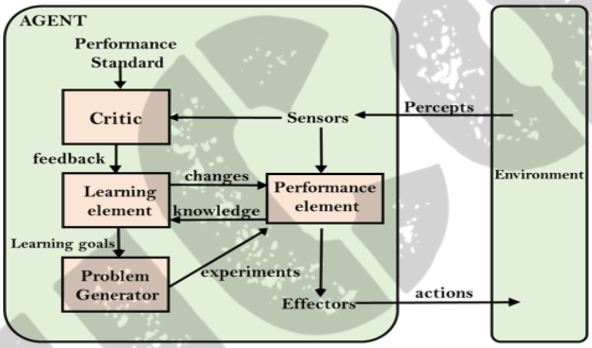
- still complicate to implement
- require sophisticated perception, reasoning, and learning
- may require expensive computation.



Learning Agents

Problem Previous agent programs describe methods for selecting actions

- How are these agent programs programmed?
- Programming by hand inefficient and ineffective!
- Solution: build learning machines and then teach them (rather than instruct them)
- Advantage: robustness of the agent program toward initially-unknown environments



- Performance element: selects actions based on percepts Corresponds to the previous agent programs
- Learning element: introduces improvements uses feedback from the critic on how the agent is doing determines improvements for the performance element
- Critic tells how the agent is doing w.r.t. performance standard
- Problem generator: suggests actions that will lead to new and informative experiences forces exploration of new stimulating scenarios

Example: Taxi Driving

- After the taxi makes a quick left turn across three lanes, the critic observes the shocking language used by other drivers.
- From this experience, the learning element formulates a rule saying this was a bad action.
- The performance element is modified by adding the new rule.
- The problem generator might identify certain areas of behavior in need of improvement, and suggest trying out the brakes on different road surfaces under different conditions.