ARTIFICIAL INTELLIGENCE

UNIT-I

Syllabus

Introduction: What is AI, Foundations of AI, History of AI, The State of Art.

Intelligent Agents: Agents and Environments, Good Behavior: The Concept of Rationality, The Nature of Environments, The Structure of Agents.

CHAPTER-1 INTRODUCTION

- ✓ The field of artificial intelligence, or AI, goes further still: it attempts not just to understand but also to *build* intelligent entities.
- ✓ AI is one of the newest sciences. Work started in earnest soon after World War II, and the name itself was coined in 1956.
- ✓ Along with molecular biology, AI is regularly cited as the "field I would most like to be in" by scientists in other disciplines.
- ✓ A student in physics might reasonably feel that all the good ideas have already been taken by Galileo, Newton, Einstein, and the rest. AI, on the other hand, still has openings for several full-time Einsteins.
- ✓ AI systematizes and automates intellectual tasks and is therefore potentially relevant to any sphere of human intellectual activity. In this sense, it is truly a universal field.

1.1.1. What is AI?

Thinking Humanly	**Thinking Rationally**
"The exciting new effort to make computers 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 solving, learning . . ." (Bellman, 1978)	"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	**Acting Rationally**
"The art of creating machines that perform 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)	"Computational Intelligence is the study of the design of intelligent agents." (Poole *et al.*, 1998) "AI . . . is concerned with intelligent behavior in artifacts." (Nilsson, 1998)

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

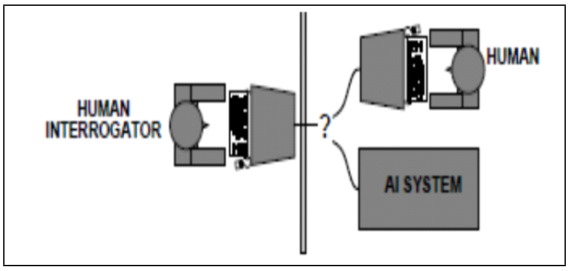
- ✓ Views of AI fall into four categories:

Thinking Humanly	Thinking Rationally
Acting Humanly	Acting Rationally

- ✓ Note: A system is rational if it does the "right

A. Acting Humanly : The Turing Test Approach

- ✓ Alan Turing (1950) - "Computing machinery and intelligence":
- ✓ Can machines think? Can machines behave intelligently?
- ✓ Operational test for intelligent behavior: the Imitation Game.



- ✓ Predicted that by 2000, a machine might have a 30% chance of fooling a lay person for 5 minutes.
- ✓ Anticipated all major arguments against AI in following 50 years.
- ✓ Suggested major components of AI: knowledge, reasoning, language understanding, learning, computer vision & Robotics.
- ➢ **Problem:** Turing test is not reproducible, constructive, or amenable to mathematical analysis.

B. Thinking humanly: The Cognitive modeling approach

- ✓ 1960s - cognitive revolution": information processing psychology replaced prevailing orthodoxy of behaviorism".
- ✓ Requires scientific theories of internal activities of the brain.
- ✓ What level of abstraction? Knowledge" or circuits"?
- ➢ **What is cognitive science??**

The study of thought, learning, and mental organization, which draws on aspects of psychology, linguistics, philosophy, and computer modeling.

- ➢ **How to validate? Requires**

  1) Predicting and testing behavior of human subjects (top-down) or

  2) Direct identification from neurological data (bottom-up).

- ✓ Both approaches (roughly, Cognitive Science and Cognitive Neuroscience) are now distinct from AI

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

- ✓ Normative (or prescriptive) rather than descriptive
- ✓ Aristotle: what are correct arguments/thought processes?
- ✓ Several Greek schools developed various forms of logic: **notation and rules of derivation for thoughts;**
- ✓ May or may not have proceeded to the idea of mechanization.
- ✓ Direct line through mathematics and philosophy to modern AI
- ➢ **Problems:**

  1) Not all intelligent behavior is mediated by logical deliberation

  2) What is the purpose of thinking? What thoughts should I have?

D. Acting rationally : The rational agent approach

- ✓ Rational behavior: doing the right thing
- ✓ The right thing: that which is expected to maximize goal achievement, given the available information.
- ✓ Doesn't necessarily involve thinking e.g., blinking reflex, but thinking should be in the service of

- ✓ Aristotle (Nicomachean Ethics): Every art and every inquiry, and similarly every action and pursuit, is thought to aim at some good.

1.1.2. Foundations of AI…

The different foundations which are contributors for AI are:

i. Philosophy

ii. Mathematics

iii. Economics

iv. Neuroscience

v. Psychology

vi. Computer engineering

vii.Control theory and cybernetics

viii.Linguistics

i. Philosophy(428 B .C .-present):

- ✓ Can formal rules be used to draw valid conclusions?
- ✓ How does the mental mind arise from a physical brain?
- ✓ Where does knowledge come from?
- ✓ How does knowledge lead to action?
- ✓ Aristotle (384-322 B.C.) was the first to formulate a precise set of laws governing the rational part of the mind.

ii. Mathematics(B.C 800-present):

- ✓ What are the formal rules to draw valid conclusions?
- ✓ What can be computed?
- ✓ How do we reason with uncertain information?
- ✓ Besides logic and computation, the third great contribution of mathematics to A1 is PROBABILITY - the theory of probability.
- ✓ The Italian Gerolamo Cardano (1501-1576) first framed the idea of probability, describing it in terms of the possible outcomes of gambling events.
- ✓ Thomas Bayes (1702-1 761) proposed a rule for updating probabilities in the light of new evidence.
- ✓ Bayes' rule and the resulting field called Bayesian analysis form the basis of most modern approaches to uncertain reasoning in A1 systems.

iii. Economics (1776-present):

- ✓ How should we make decisions so as to maximize payoff?
- ✓ How should we do this when others may not go along?
- ✓ How should we do this when the payoff may be fix in the future?
- ✓ Work in economics and operations research has contributed much to our notion of rational Agents.
- ✓ Herbert Simon (191 6- 2001), the pioneering AI researcher, won the Nobel prize in economics in 1978 for his early SATISFICING work showing that models based on satisficing-making decisions that are "good

enough“.

✓

iv Neuroscience (1861-present):

- ✓ How do brains process information?
- ✓ Neuroscience is the study of the nervous system, particularly the brain.
- ✓ Paul Broca's (1824-1880) study of aphasia (speech deficit) in brain-damaged patients.
- ✓ He proved speech production was localized to a portion of the left hemisphere now called NEURONS Broca's area.
- ✓ Carnillo Golgi (1843-1926) developed a staining technique allowing the observation of individual neurons in the brain.
- ✓ The recent development of functional magnetic resonance imaging (MRI) (Ogawa et al., 1990) is giving neuroscientists unprecedentedly detailed images of brain activity.

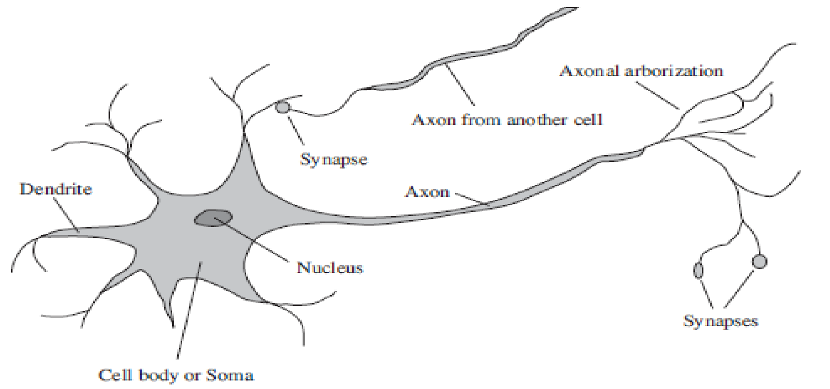


Figure 1.2 The parts of a nerve cell or neuron. Each neuron consists of a cell body, or soma, that contains a cell nucleus. Branching out from the cell body are a number of fibers called dendrites and a single long fiber called the axon. The axon stretches out for a long distance, much longer than the scale in this diagram indicates. Typically, an axon is 1 cm long (100 times the diameter of the cell body), but can reach up to 1 meter. A neuron makes connections with 10 to 100,000 other neurons at junctions called synapses. Signals are propagated from neuron to neuron by a complicated electrochemical reaction. The signals control brain activity in the short term and also enable long-term changes in the connectivity of neurons. These mechanisms are thought to form the basis for learning in the brain. Most information processing goes on in the cerebral cortex, the outer layer of the brain. The basic organizational unit appears to be a column of tissue about 0.5 mm in diameter, containing about 20,000 neurons and extending the full depth of the cortex about 4 mm in humans).

	Supercomputer	Personal Computer	Human Brain
Computational units	10^4 CPUs, 10^{12} transistors	4 CPUs, 10^9 transistors	10^{11} neurons
Storage units	10^{14} bits RAM	10^{11} bits RAM	10^{11} neurons
	10^{15} bits disk	10^{13} bits disk	10^{14} synapses
Cycle time	10^{-9} sec	10^{-9} sec	10^{-3} sec
Operations/sec	10^{15}	10^{10}	10^{17}
Memory updates/sec	10^{14}	10^{10}	10^{14}

Figure 1.3 A crude comparison of the raw computational resources available to the IBM BLUE GENE supercomputer, a typical personal computer of 2008, and the human brain. The brain’s numbers are essentially fixed, whereas the supercomputer’s numbers have been increasing by a factor of 10 every 5 years or so, allowing it to achieve rough parity with the brain. The personal computer lags behind on all metrics except cycle time.

- ➢ **NOTE :Moore’s Law predicts that the CPU’s gate count will equal the brain’s neuron count around 2020.**

v Psychology (1879-present):

- ✓ How do humans and animals think and act?
- ✓ The origin of scientific psychology are traced back to the wok if German physiologist Hermann von Helmholtz(1821-1894) and his student Wilhelm Wundt(1832 – 1920).
- ✓ In 1879,Wundt opened the first laboratory of experimental psychology at the university of Leipzig.
- ✓ In US, the development of computer modeling led to the creation of the field of cognitive science.
- ✓ The field can be said to have started at the workshop in September 1956 at MIT.

vi Computer Engineering (1940-present):

- ✓ 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 of choice.
- ✓ AI also owes a debt to the software side of computer science, which has supplied the operating systems, programming languages, and tools needed to write modern programs.

vii Control theory and Cybernetics (1948-present):

- ✓ How can artifacts operate under their own control?
- ✓ Ktesibios of Alexandria (c. 250 B.C.) built the first self-controlling machine: a water clock.
- ✓ It contains regulator that kept the flow of water running through it at a constant, predictable pace.
- ✓ 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.

Viii Linguistics:

- ✓ How does language relate to thought?
- ✓ Modem linguistics and AI, then, were "born" at about the same time, and grew up together, intersecting in a hybrid field called computational linguistics or natural language processing.

1.1.3. History of AI...

A. Gestation, 1943–56:

- ✓ McCulloch & Pitts (1943)

  Artificial neural net—proved equivalent to Turing machine;
- ✓ Shannon, Turing (1950)

  Chess playing programs
- ✓ Marvin Minsky (1951)

  First neural net computer—SNARC
- ✓ Dartmouth College (1956)

  Term “AI” coined by John McCarthy
- ✓ Newell & Simon presented LOGIC THEORIST program

B. Early enthusiasm, 1952–69:

- ✓ Lots of progress.
- ✓ Programs written that could:
  - – plan, learn
  - – play games,
  - – prove theorems, in general, solve problems.
- ✓ Major feature of the period were microworlds—toy problem domains.

Major centres established:

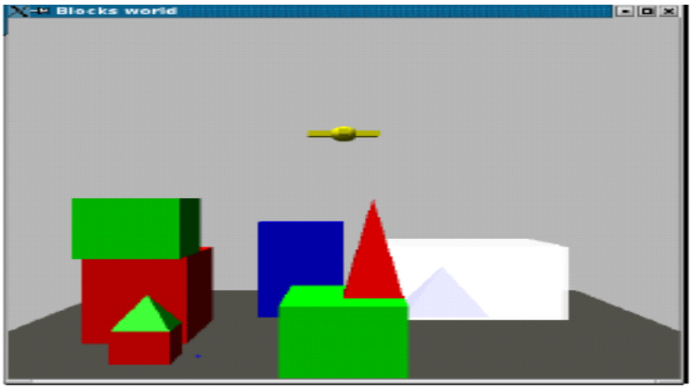
Minsky, MIT

McCarthy, Stanford

Newell & Simon, CMU

For the next 20 years these places dominated the field.

- ✓ Example: blocks world.
- ✓ Planners simulated manipulating simple shapes like this:



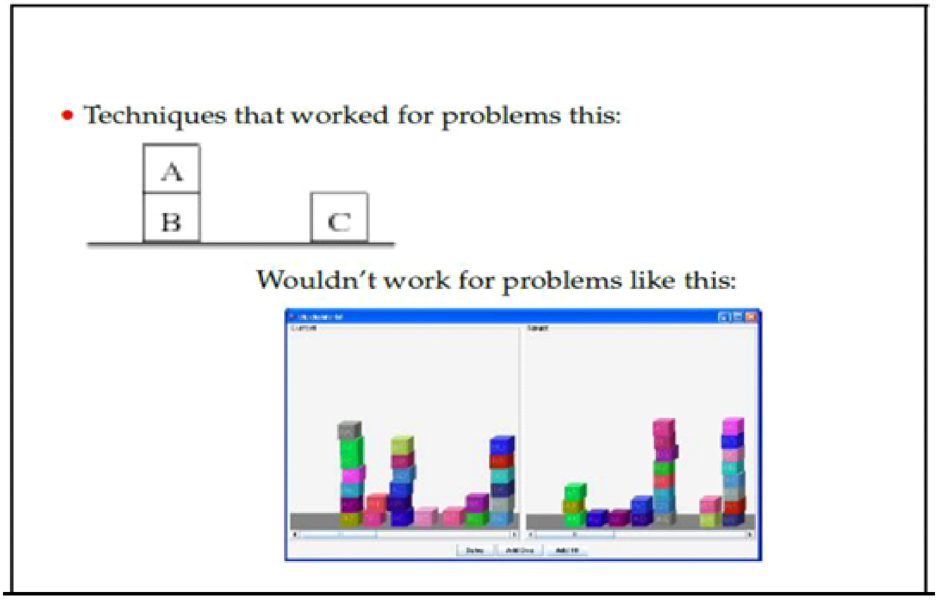
- ✓ Even some simple natural language handling:

```
Person: GRASP THE PYRAMID.
Computer: I DON'T UNDERSTAND WHICH PYRAMID YOU MEAN.
Person: FIND A BLOCK WHICH IS TALLER THAN THE ONE
        YOU ARE HOLDING AND PUT IT INTO THE BOX.
Computer: BY "IT", I ASSUME YOU MEAN THE BLOCK WHICH
          IS TALLER THAN THE ONE I AM HOLDING.
Computer: OK. (does it)
Person: WHAT DOES THE BOX CONTAIN?
Computer: THE BLUE PYRAMID AND THE BLUE BLOCK.
Person: WHAT IS THE PYRAMID SUPPORTED BY?
Computer: THE BOX.
Person: HOW MANY BLOCKS ARE NOT IN THE BOX?
Computer: FOUR OF THEM.
```

- The promise was "It'll scale, honest. . . "

C. A dose of reality, 1966–1973:

- ✓ Techniques developed on microworlds would not scale.
- ✓ Implications of complexity theory—developed in late 1960s, early 1970s—began to be appreciated:
    - o brute force techniques will not work.
    - o works in principle does not mean works in practice.
- ✓ Lots of early programs did symbol manipulation without any real understanding of the domain they were in.



D. Knowledge based systems, 1969–1979:

- ✓ General purpose, brute force techniques don't work, so use knowledge rich solutions.

✓ Early 1970s saw emergence of expert systems as systems capable of exploiting knowledge about tightly focused domains to solve problems normally considered the domain of experts.

- ✓ Ed Feigenbaum's knowledge principle: [Knowledge] is power, and computers that amplify that knowledge will amplify every dimension of power.
- ✓ Expert systems success stories:
    - o MYCIN — blood diseases in humans;
    - o DENDRAL — interpreting mass spectrometers;
    - o R1/XCON — configuring DEC VAX hardware;
    - o PROSPECTOR —finding promising sites for mineral deposits;
- ✓ Expert systems emphasised knowledge representation: rules, frames, semantic nets.
- ➢ **Problems:**

  – the knowledge elicitation bottleneck;

  – marrying expert system & traditional software;

  – breaking into the mainstream.

E. AI makes money, 1980–present:

- ✓ R1 was the first commercial expert system.
- ✓ Led to a boom in expert systems companies.

- Like an earlier dot com boom

✓ Most of those companies failed to deliver increased productivity for their customers.

✓ They went bust.

✓ AI suffered from all the broken promises, but didn't go away.

F. AI and the scientific method, 1987– present:

✓ Connection back to other fields.

- Probability/statistics
- Control theory
- Economics
- Operations research.

✓ New industries spawned.

- Data mining is machine learning.

✓ Low-key successes.

- Expert systems embedded in Windows.

✓ Increased emphasis on shared datasets, common problems, competitions.

G. Intelligent agents, 1993–present:

✓ Emphasis on understanding the interaction between agents and environments.

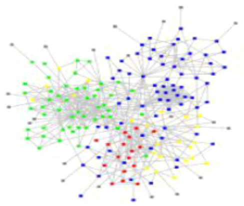
✓ AI as component, rather than as end in itself.

- "Useful first" paradigm — Etzioni (NETBOT, US$35m)
- "Raisin bread" model — Winston.

✓ More about interaction between components, emergent intelligence, and doing well enough to be useful.

H. Large datasets, 2001–present:

- ✓ Change in emphasis from algorithm to data.
  - o Perhaps with enough data, we don't have to be smart to be good.
- ✓ For example, how to fill in a gap in a picture.
  - o Perhaps you have a picture of your ex in front of a great landscape.
- ✓ Poll lots of similar pictures for a matching piece.
  - o Moving from 10,000 images to 2,000,000 led to a big jump in performance.
- ✓ Wisdom of the crowd.
- ✓ The growth of the internet makes this much easier in 2010 than it was in 1990.



- ✓ Currently a lot of interest in mining data from social networks.
- ✓ If people are connected, they are similar in some sense.

1.1.4. THE STATE OF THE ART:

Here we sample a few applications;

1. **Robotic vehicles**: A driverless robotic car named STANLEY sped through the rough terrain of the Mojave dessert at 22 mph, finishing the 132-mile course first to win the 2005 DARPA Grand Challenge.
2. **Speech recognition**: A traveler calling United Airlines to book a flight can have the entire conversation guided by an automated speech recognition and dialog management system.
3. **Autonomous planning and scheduling**: A hundred million miles from Earth, NASA's Remote Agent program became the first on-board autonomous planning program to control the scheduling of operations for a spacecraft.
4. **Game playing**: IBM's DEEP BLUE became the first computer program to defeat the world champion in a chess match when it bested Garry Kasparov by a score of 3.5 to 2.5 in an exhibition match.
5. **Spam fighting**: Each day, learning algorithms classify over a billion messages as spam, saving the recipient from having to waste time deleting what, for many users, could comprise 80% or 90% of all messages, if not classified away by algorithms.
6. **Logistics planning**: During the Persian Gulf crisis of 1991, U.S. forces deployed a Dynamic Analysis and Replanning Tool, DART (Cross and Walker, 1994), to do automated logistics planning and scheduling for transportation.
7. **Robotics**: The iRobot Corporation has sold over two million Roomba robotic vacuum cleaners for home use.
8. **Machine Translation**: A computer program automatically translates from Arabic to English .

1.2.1. AGENTS AND ENVIRONMENT :

➢ **Agent Definition…**

An agent is a system that:

– is situated in an environment,
– is capable of perceiving its environment, and
– is capable of acting in its environment with the goal of satisfying its design objectives.
(or)

Anything that can be viewed as perceiving its environment through sensors and SENSOR acting upon that environment through actuators.

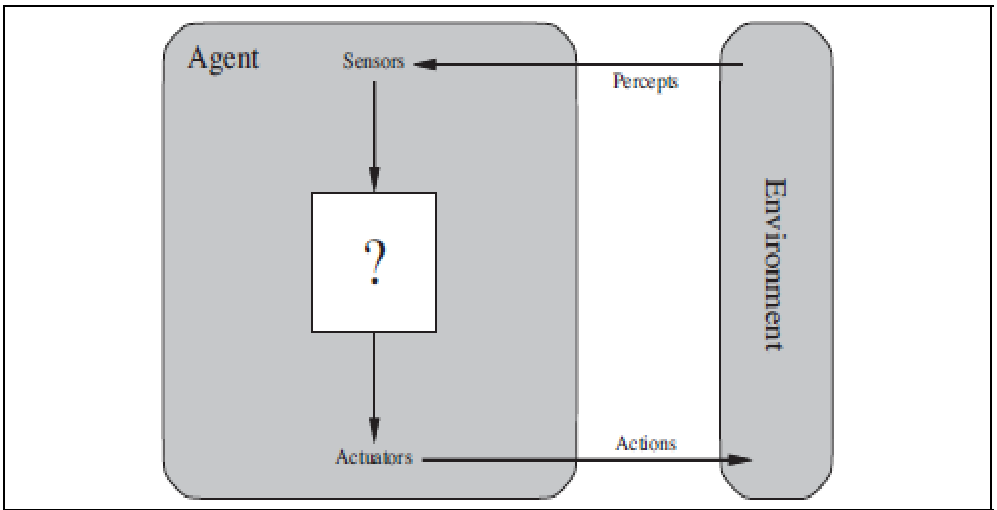


Figure 2.1 Agents interact with environments through sensors and actuators.

(OR)

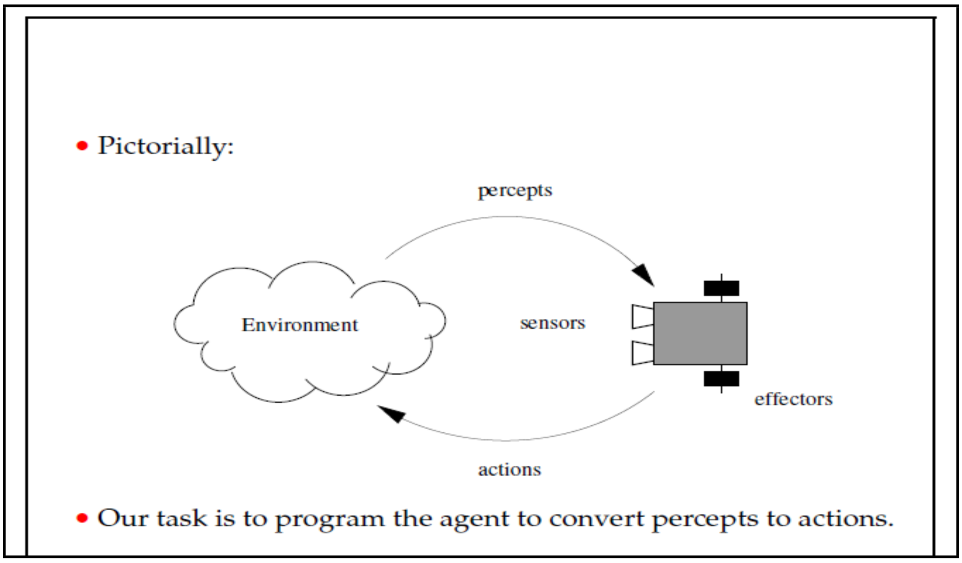


Figure : An agent example representation

➢ Examples for different agents:

i. Human "agent":

– environment: physical world;

– sensors: eyes, ears, . . .

– effectors: hands, legs, . . .

ii. Software agent:

– environment: (e.g.) UNIX operating system;

– sensors: keystrokes, file contents, n/w packets . . .

– effectors: rm, chmod, . . .

iii. Robot:

– environment: physical world;

– sensors: sonar, camera;

– effectors: wheels.

➢ Important terms…

✓ **Percept:**

We use the term percept to refer to the agent's perceptual inputs at any given instant.

✓ **Percept Sequence:**

An agent's percept sequence is the complete history of everything the agent has ever perceived.

✓ **Agent function: f:P* →A**

Mathematically speaking, we say that an agent's behavior is described by the agent function that **maps any given percept sequence to an action.**

➢ Agent function VS Agent program:

The agent function is an abstract mathematical description; the agent program is a concrete implementation, running on the agent architecture.

Example: The vacuum-cleaner world…

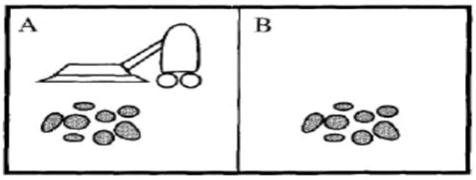


Figure 2.2 A vacuum-cleaner world with just two locations.

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.

1/23/2018 PVN RAJESWARI-CSE DEPT- VTA 54

1.2.2 Good behavior – Concept of Rationality…

- **Rational Agent…**

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.

Obviously, doing the right thing is better than doing the wrong thing. The right action is the one that will cause the agent to be most successful.

- **Performance measures:**

A performance measure embodies the criterion for success of an agent's behavior.

When an agent is plunked down in an environment, it generates a sequence of actions according to the percepts it receives.

- ✓ This sequence of actions causes the environment to go through a sequence of states. If the sequence is desirable, then the agent has performed well.

- **Rationality :**

What is 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.

- **Omniscience, learning, and autonomy…**
  - ✓ An omniscient agent knows the actual outcome of its actions and can act accordingly; but omniscience is impossible in reality.
  - ✓ Doing actions in order to modify future percepts-sometimes called information gathering-is an important part of rationality.
  - ✓ Our definition requires a rational agent not only to gather information, but also to learn as much as possible from what it perceives.
  - ✓ To the extent that an agent relies on the prior knowledge of its designer rather than on its own percepts, we say that the agent lacks autonomy. A rational agent should be autonomous-it should learn what it can to compensate for partial or incorrect prior knowledge.

1.2.3 The Nature of Environments :

- ✓ Task environments :

  These are essentially the "problems" to which rational agents are the "solutions.“

- ✓ Specifying the task environment : PEAS

  The rationality of the simple vacuum-cleaner agent, needs specification of …

- – the performance measure - P
- – the environment - E
- – the agent's actuators and sensors - A & S

PEAS Example…

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.

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.

➢**Properties of task environments…**

I. Fully observable vs. partially observable
II. Deterministic vs. stochastic
III. Episodic vs. sequential
IV. Static vs. dynamic
V. Discrete vs. continuous
VI. Single agent vs. Multiagent

I. Fully observable vs. partially observable…

- ✓ If an agent's sensors give it access to the complete state of the environment at each point in time, then we say that the task environment is fully observable.
- ✓ A task environment is effectively fully observable if the sensors detect all aspects that are relevant to the choice of action;
- ✓ An environment might be partially observable because of noisy and inaccurate sensors or because parts of the state are simply missing from the sensor.

II. Deterministic vs. 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; other-wise, it is stochastic.

III. Episodic vs. sequential…

- ✓ In an episodic task environment, the agent's experience is divided into atomic episodes.
- ✓ Each episode consists of the agent perceiving and then performing a single action.
- ✓ Crucially, the next episode does not depend on the actions taken in previous episodes.
- ✓ In sequential environments, on the other hand, the current decision could affect all future decisions. Chess and taxi driving are sequential.

IV. Static vs. 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.
- ✓ If the environment itself does not change with the passage of time but the agent’s performance score does, then we say the environment is **semidynamic**. Taxi driving is clearly dynamic: the other cars and the taxi itself keep moving while the driving algorithm dithers about what to do next. Chess, when played with a clock, is semidynamic. Crossword puzzles are static.

V. Discrete vs. continuous…

- ✓ A discrete-state environment such as a chess game has a finite number of distinct states.
- ✓ 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,.)

VI. Single agent vs. multiagent…

- ✓ 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.

- ✓ As one might expect, the hardest case is partially observable, stochastic, sequential, dynamic, continuous, and Multiagent.

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.

1.2.4 The Structure of Agents…

➢ Agent programs:

- ✓ The job of AI is to design the agent program that implements the agent function mapping percepts to actions.
- ✓ Agent = architecture +program
- ✓ The agent programs all have the same skeleton: they take the current percept as input from the sensors and return an action to the actuators.
- ✓ Notice the difference between the agent program, which takes the current percept as input, and the agent function, which takes the entire percept history.

➢ Agent Types/Categories…

i. Table-driven agents:

-- use a percept sequence/action table in memory to find the next action. They are implemented by a (large) lookup table.

ii. Simple reflex agents:

-- are based on condition-action rules, implemented with an appropriate production system. They are stateless devices which do not have memory of past world states.

iii. Agents with memory (Model):

-- have internal state, which is used to keep track of past states of the world.

iv. Agents with goals:

-- are agents that, in addition to state information, have goal information that describes desirable situations. Agents of this kind take future events into consideration.

v. Utility-based agents:

--base their decisions on classic axiomatic utility theory in order to act rationally.

vi. Learning Agents:

i. Table Driven Agent:

```
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.

➢ **Drawbacks:**

- ✓ Table lookup of percept-action pairs defining all possible condition-action rules necessary to interact in an environment.

➢ **Problems :**

– Too big to generate and to store (Chess has about 10^120 states, for example)

– No knowledge of non-perceptual parts of the current state

– Not adaptive to changes in the environment; requires entire table to be updated if changes occur

– Looping: Can't make actions conditional

– Take a long time to build the table

– No autonomy

– Even with learning, need a long time to learn the table entries

ii. Simple Reflex Agent:

- ✓ The simplest kind of agent is the simple reflex agent.
- ✓ These agents select actions on the basis of the current percept, ignoring the rest of the percept history. E.g. the vacuum-agent
- ✓ Large reduction in possible percept/action situations.
- ✓ Implemented through condition-action rules.
- ✓ Simple reflex behaviors occur even in more complex environments. Imagine yourself as the driver of the automated taxi. If the car in front brakes and its brake lights come on, then you should notice this and initiate braking.
- ✓ In other words, some processing is done on the visual input to establish the condition we call "The car in front is braking." Then, this triggers some established connection in the agent program to the action "initiate braking."
- ✓ We call such a connection a **condition–action rule**,5 written as **if** *car-in-front-is-braking* **then** *initiate-braking*.
- ✓ **Example : Vaccum Cleaner : If dirty then suck**

```
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.

```
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.

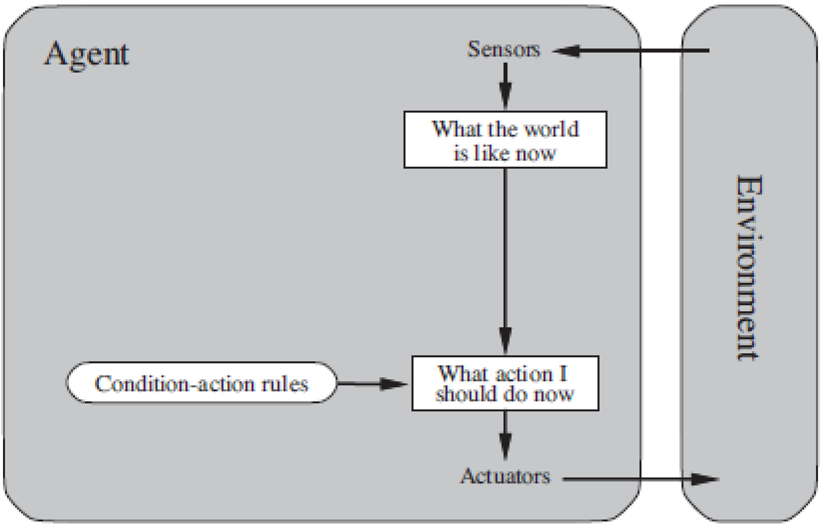


Figure 2.9 Schematic diagram of a simple reflex agent.

- **Characteristics:**
  - ✓ Only works if the environment is fully observable.
  - ✓ Lacking history, easily get stuck in infinite loops.
  - ✓ One solution is to randomize actions.

iii. Model based reflex Agents:

- ✓ The most effective way to handle partial observability is for the agent to keep track of the part of the world it can't see now.
- ✓ That is, the agent should maintain some sort of internal state that depends on the percept history and thereby reflects at least some of the unobserved aspects of the current state.
- ✓ 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 examp le, 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 working -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.

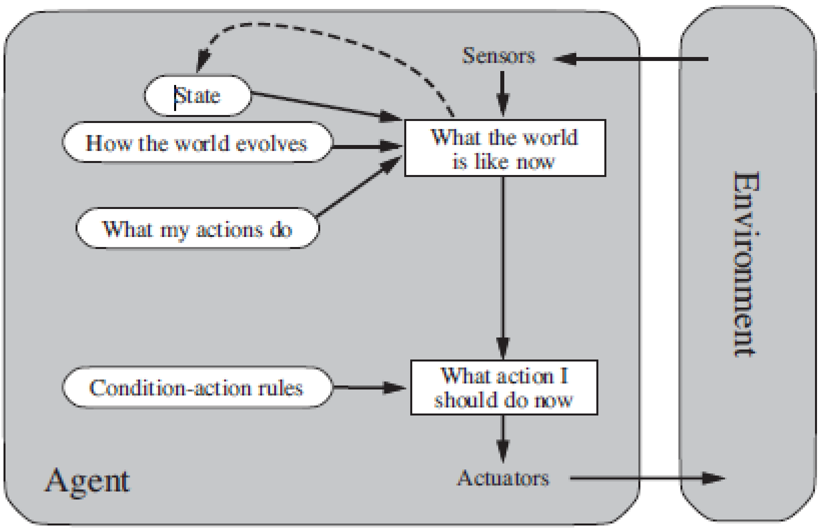


Figure 2.11 A model-based reflex agent.

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*

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.

iv 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, turn right, or go straight on.

- ✓ The correct decision depends on where the taxi is trying to get to.

- ✓ In other words, as well as a current state description, the agent needs some sort of goal information that describes situations that are desirable-for example, being at the passenger's destination.

- ✓ The agent program can combine this with information about the results of possible actions (the same information as was used to update internal state in the reflex agent) in order to choose actions that achieve the goal.

- ✓ Sometimes goal-based action selection is straightforward—for example, when goal satisfaction results immediately from a single action. Sometimes it will be more tricky—for example, when the agent has to consider long sequences of twists and turns in order to find a way to achieve the goal.

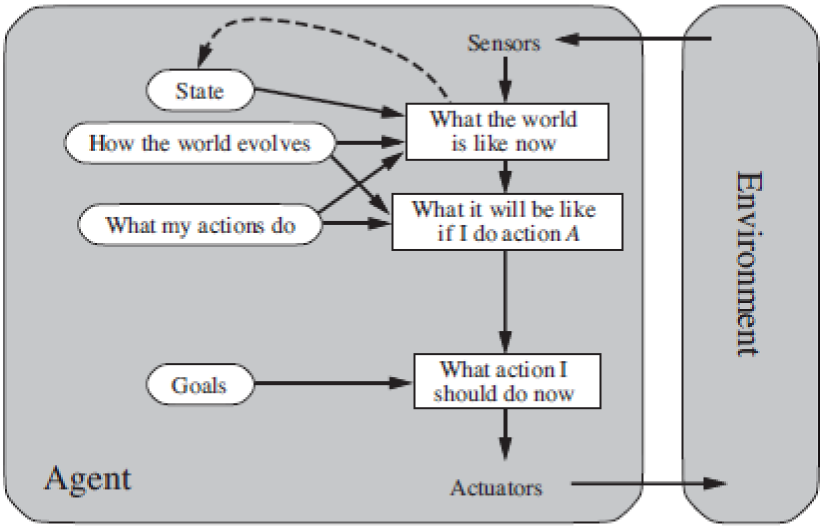


Figure 2.13 A model-based, goal-based agent. It keeps track of the world state as well as a set of goals it is trying to achieve, and chooses an action that will (eventually) lead to the achievement of its goals.

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

- ✓ Although 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.

V Utility based Agents:

- ✓ Goals alone are not really enough to generate high-quality behavior in most environments. For example, there are many 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.
- ✓ Goals just provide a crude binary distinction between "happy" and "unhappy" states, whereas a more general performance measure should allow a comparison of different world states according to exactly how happy they would make the agent if they could be achieved.

- ✓ Because "happy" does not sound very scientific, the customary terminology is to say that if one world state is preferred to another, then it has higher **utility** for the agent.

- ✓ **A utility function** maps a state (or a sequence of states) onto a real number, which describes the associated degree of happiness.
- ✓ A complete specification of the utility function allows rational decisions in two kinds of cases where goals are

inadequate.

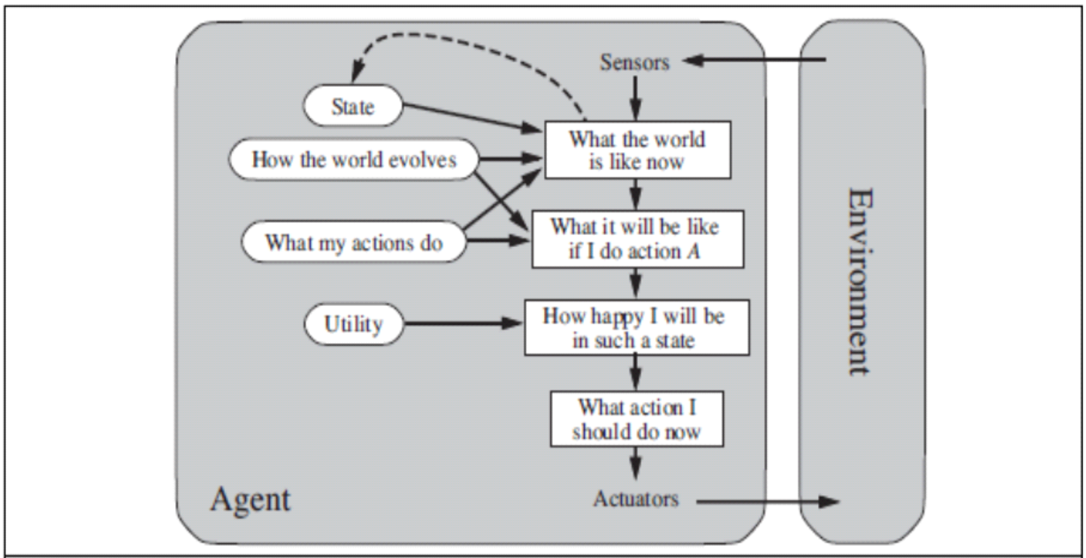


Figure 2.14 A model-based, utility-based agent. It uses a model of the world, along with a utility function that measures its preferences among states of the world. Then it chooses the action that leads to the best expected utility, where expected utility is computed by averaging over all possible outcome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

- ✓ First, when there are conflicting goals, only some of which can be achieved (for example, speed and safety), the utility function specifies the appropriate tradeoff.
- ✓ Second, 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.
- ✓ Partial observability and stochasticity are ubiquitous in the real world, and so, therefore, is decision making under uncertainty.
- ✓ Technically speaking, a rational utility-based agent chooses the action that maximizes the **expected utility** of the action outcomes—that is, the utility the agent expects to derive, on average, given the probabilities and utilities of each outcome.

Vi Learning Agents:

- ✓ All agents can improve their performance through learning.
- ✓ A learning agent can be divided into four conceptual components as shown in figure:
  - a. Learning element
  - b. Performance element
  - c. Critic
  - d. Problem generator

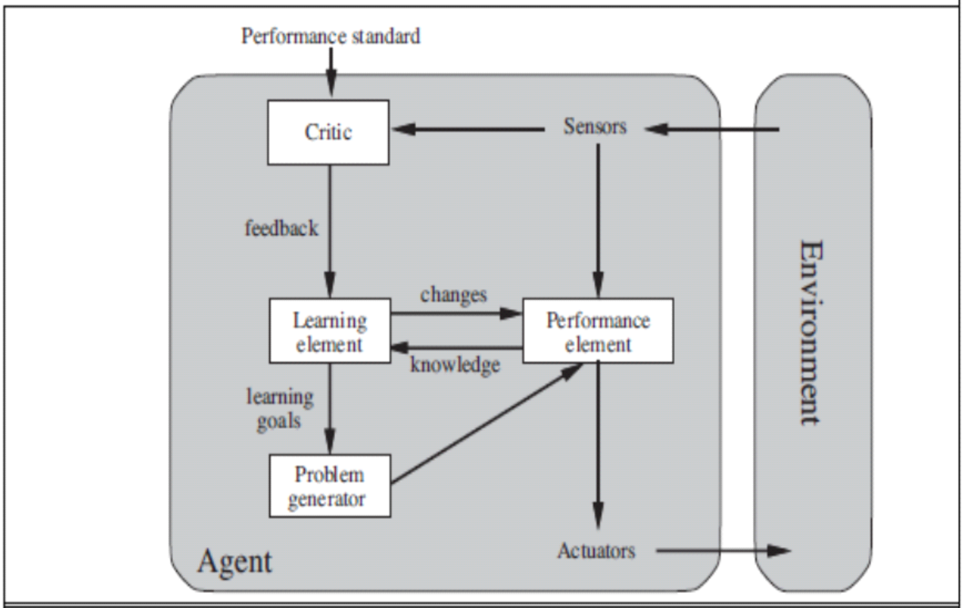


Figure 2.15 A general learning agent.

a. Learning element - is responsible for making improvements.

b. Performance element - is responsible for selecting external actions. The performance element is what we have previously considered to be the entire agent: it takes in percepts and decides on actions.

c. Critic - The learning element uses feedback from the critic on how the agent is doing and determines how the performance element should be modified to do better in the future.

d. Problem generator - It is responsible for suggesting actions that will lead to new and informative experiences.

****