CST401 ARTIFICIAL INTELLIGENCE
MODULE 1

- ARTIFICIAL INTELLIGENCE
  Artificial intelligence allows machines to replicate the capabilities of the human mind. From the development of self-driving cars to the development of smart assistants like Siri and Alexa
- AI is a growing part of everyday life.

- Artificial intelligence is a wide-ranging branch of computer science concerned with building smart machines capable of performing tasks that typically require human intelligence.

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)

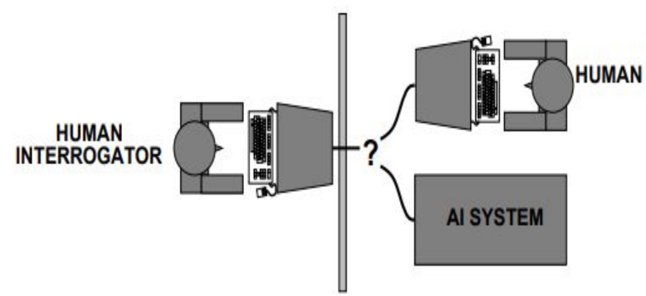
Figure 1.1 Some definitions of artificial intelligence, organized into four categories.

The definitions on the left measure success in terms of fidelity to human performance,

The ones on the right measure against an ideal performance measure, called rationality.
A system is rational if it does the “right thing,” given what it knows.

Turing Test

- (Human) judge communicates with a human and a machine over text-only channel,
- Both human and machine try to act like a human,
- Judge tries to tell which is which.
- Numerous variants



What is the Turing Test in Artificial Intelligence?

- NLP to communicate successfully.
- Knowledge Representation to act as its memory.
- Automated Reasoning to use stored information to answer question and draw new conclusions.
- ML to detect new patterns and adapt to new circumstances

	HUMAN	RATIONAL
THOUGHT	Systems that think like humans	Systems that think rationally
BEHAVIOUR	Systems that act like humans	Systems that act rationally

Turing Test on unsuspecting judges

- It is possible to (temporarily) fool humans who do not realize they may be talking to a bot
- ELIZA program rephrases partner's statements and questions (~psychotherapist)

- Modern variant: AOLiza[fragment from http://archives.cnn.com/2000/TECH/computing/08/29/aoliza.idg/]
- Five:just asking if richardcan tell me about some good lobster diving spots thatsit if this is a problem then forget iotidontknow whatsgoing on here
- **AOLiza:**that is interesting. please continue.
- **five:**continue what just ask richardwhen he gets home about some lobster diving up hat way thatsall if you got a problem with it then forget it,thisisntworhtit idontknow whatsbeen going on or what bevhas told you but ihaventdone anything to anybody iamthe one that got used and dumped
- **AOLiza:**tell me more about that.
- **five:**about what
- **AOLiza:**what do you think?
- **five:**therersnothing to tell she us3ed me then got tired of me said iwasnther type anymore iwas possesiveand controlling
- **AOLiza:**perhaps ialready know you were possesiveand controlling.
- **five:**iwas huh you nuts igave her all the room she wanted never told nhewrewhat to do or where to go

Acting humanly: The Turing Test approach

- tell whether the written responses come from a person or from a computer
- 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
- TOTAL TURING TEST-To pass the total Turing Test, the computer will need computer vision to perceive objects, and robotics to manipulate objects and move about

Thinking humanly: The cognitive modeling approach

- introspection—trying to catch our own thoughts as they go by
- psychological experiments—observing a person in action
- brain imaging—observing the brain in action
- cognitive science brings together computer models from AI and experimental techniques from psychology to construct precise and testable theories of the human mind.

	HUMAN	RATIONAL
THOUGHT	Systems that think like humans	Systems that think rationally
BEHAVIOUR	Systems that act like humans	Systems that act rationally

Thinking rationally: The "laws of thought" approach

- SYLLOGISM : an instance of a form of reasoning in which a conclusion is drawn from two given or assumed propositions, "Socrates is a man; all men are mortal; therefore, Socrates is mortal."
- LOGIC : laws of thought to govern the operation of the mind
- formal terms required by logical notation
- exhaust the computational resources of any computer unless it has some guidance as to which reasoning steps to try first

	HUMAN	RATIONAL
THOUGHT	Systems that think like humans	Systems that think rationally
BEHAVIOUR	Systems that act like humans	Systems that act rationally

Acting rationally: The rational agent approach

- An agent is just something that acts
- Rational behavior is doing the right thing
- Right thing is expected to maximize goal achievement, given available information
- computer agents
  - operate autonomously,
  - perceive their environment,
  - persist over a prolonged time period,
  - adapt to change, and
  - create and pursue goals

	HUMAN	RATIONAL
THOUGHT	Systems that think like humans	Systems that think rationally
BEHAVIOUR	Systems that act like humans	Systems that act rationally

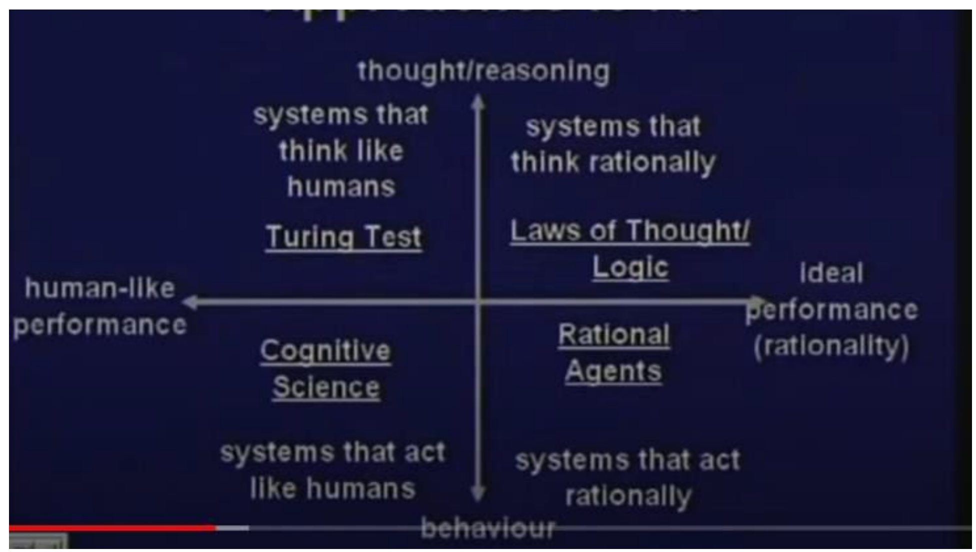
- is one that acts so as to achieve the best outcome or, when there is uncertainty, the best expected outcome
- correct thing to do, but something must still be done. There are also ways of acting rationally that cannot be said to involve inference

REQUIREMENTS

- **NATURAL LANGUAGE PROCESSING**
- To enable it to communicate successfully
- **KNOWLEDGE REPRESENTATION**
- Knowledge representation to store what it knows or hears;
- **AUTMATED REASONING**
- Automated reasoning to use the stored information to answer questions and to draw new conclusions
- **MACHINE LEARNING**
- machine learning to adapt to new circumstances and to detect and extrapolate patterns.
- **COMPUTER VISION**
- Computer vision to perceive objects
- **ROBOTICS**
- Robotics to manipulate objects and move about

How do we measure if Artificial Intelligence is acting like a human?

- Turing Test
- ➢The Cognitive Modelling Approach
- ➢The Law of Thought Approach
- ➢The Rational Agent Approach



Artificial Intelligence

- An intelligent entity created by humans.
- Capable of performing tasks intelligently without being explicitly instructed.
- Capable of thinking and acting rationally and humanely.

INTELLIGENT AGENTS

- An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators.
- An Agent runs in the cycle of perceiving, thinking, and acting.
- An agent can be:
- Human-Agent: A human agent has eyes, ears, and other organs which work for sensors and hand, legs, vocal tract work for actuators.
- Robotic Agent: A robotic agent can have cameras, infrared range finder, NLP for sensors and various motors for actuators.
- Software Agent: Software agent can have keystrokes, file contents as sensory input and act on those inputs and display output on the screen. Hence the world around us is full of agents such as thermostat, cellphone, camera, and even we are also agents.

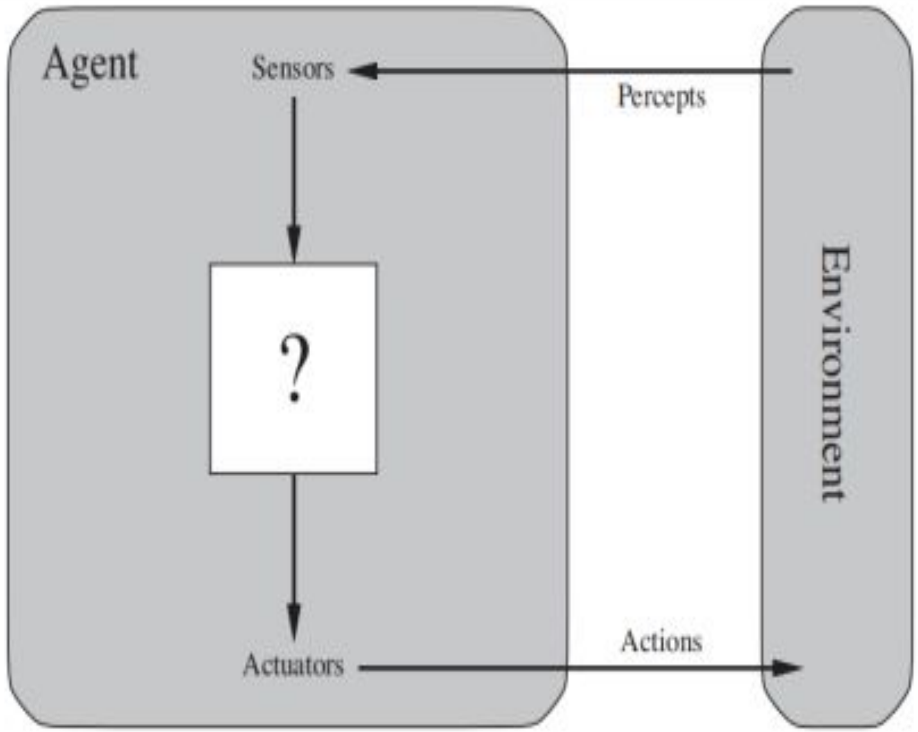


Figure 2.1 Agents interact with environments through sensors and actuators.

- **percept** to refer to the agent’s perceptual inputs at any given instant.
- An PERCEPT SEQUENCE agent’s **percept sequence** is the complete history of everything the agent has ever perceived.
- In general, *an 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*

- we should first know about sensors, effectors, and actuators.
- Sensor: Sensor is a device which detects the change in the environment and sends the information to other electronic devices. An agent observes its environment through sensors.
- Actuators: Actuators are the component of machines that converts energy into motion. The actuators are only responsible for moving and controlling a system. An actuator can be an electric motor, gears, rails, etc.
- Effectors: Effectors are the devices which affect the environment. Effectors can be legs, wheels, arms, fingers, wings, fins, and display screen

- Intelligent Agents: An intelligent agent is an autonomous entity which act upon an environment using sensors and actuators for achieving goals. An intelligent agent may learn from the environment to achieve their goals. A thermostat is an example of an intelligent agent.
- Following are the main four rules for an AI agent:
- Rule 1: An AI agent must have the ability to perceive the environment.
- Rule 2: The observation must be used to make decisions.
- Rule 3: Decision should result in an action.
- Rule 4: The action taken by an AI agent must be a rational action.


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

The Vacuum Cleaner World

- 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.
- Percepts: location and contents, e.g., [A,Dirty]
- Actions: Left, Right, Suck, NoOp Agent’s
- function → look-up table
- For many agents this is a very large table

Percept sequence	Action
$[A, Clean]$	$Right$
$[A, Dirty]$	$Suck$
$[B, Clean]$	$Left$
$[B, Dirty]$	$Suck$
$[A, Clean], [A, Clean]$	$Right$
$[A, Clean], [A, Dirty]$	$Suck$
$\vdots$	$\vdots$
$[A, Clean], [A, Clean], [A, Clean]$	$Right$
$[A, Clean], [A, Clean], [A, Dirty]$	$Suck$
$\vdots$	$\vdots$

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.

• GOOD BEHAVIOR: THE CONCEPT OF RATIONALITY

- Rational Agent:
- A rational agent is an agent which has clear preference, models uncertainty, and acts in a way to maximize its performance measure with all possible actions.
- A rational agent is said to perform the right things. AI is about creating rational agents to use for game theory and decision theory for various real-world scenarios.
- For an AI agent, the rational action is most important because in AI reinforcement learning algorithm, for each best possible action, agent gets the positive reward and for each wrong action, an agent gets a negative reward.

- **Vacuum Cleaner Revisited** We might propose to measure performance by the amount of dirt cleaned up in a single eight hour shift. With a rational agent, of course, what you ask for is what you get.
- A rational agent can maximize this performance measure by cleaning up the dirt, then dumping it all on the floor, then cleaning it up again, and so on.
- A more suitable performance measure would reward the agent for having a clean floor. For example, one point could be awarded for each clean square at each time step (perhaps with a penalty for electricity consumed and noise generated).
- As a 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

- The rationality of an agent is measured by its performance measure.
- Rationality can be judged on the basis of following points:
  - Performance measure which defines the success criterion.
  - Agent prior knowledge of its environment.
  - Best possible actions that an agent can perform.
  - The sequence of percepts.

Omniscience, learning, and autonomy

- An omniscient agent knows the *actual o*utcome of its actions and can act accordingly; but omniscience is impossible in reality.
- Rationality maximizes *expected* performance, while perfection maximizes *actual* performance.
- Doing actions *in order to modify future percepts*—sometimes called **information gathering**
- Our definition requires a rational agent not only to gather information but also to **learn** as much as possible from what it perceives.
- A rational agent should be autonomous—it should learn what it can to compensate for partial or incorrect prior knowledge.

Nature of Environments

- **PEAS Representation**
- PEAS System is used to categorize similar agents together. The PEAS system delivers the performance measure with respect to the environment, actuators, and sensors of the respective agent. Most of the highest performing agents are Rational Agents.
- **PEAS** stands for a *Performance measure, Environment, Actuator, Sensor*.

- PEAS is a type of model on which an AI agent works upon. When we define an AI agent or rational agent, then we can group its properties under PEAS representation model. It is made up of four words:
- P: Performance measure
- E: Environment
- A: Actuators
- S: Sensors

- Here performance measure is the objective for the success of an agent's behavior.
- **PEAS for self-driving cars:**
- Let's suppose a self-driving car then PEAS representation will be:
- **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

Example of Agents with their PEAS representation

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.

Types of Environments in AI

- An environment in artificial intelligence is the surrounding of the agent. The agent takes input from the environment through sensors and delivers the output to the environment through actuators. There are several types of environments:
  - Fully Observable vs Partially Observable
  - Deterministic vs Stochastic
  - Competitive vs Collaborative
  - Single-agent vs Multi-agent
  - Static vs Dynamic
  - Discrete vs Continuous
  - Episodic vs Sequential
  - Known vs Unknown

1. Fully Observable vs Partially Observable

- When an agent sensor is capable to sense or access the complete state of an agent at each point in time, it is said to be a **fully observable environment** else it is partially observable.
- Maintaining a fully observable environment is easy as there is no need to keep track of the history of the surrounding.
- 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 environment is called **unobservable** when the agent has no sensors in all environments.
- Examples:
- **Chess –** the board is fully observable, and so are the opponent's moves.
- **Driving –** the environment is partially observable because what's around the corner is not known.

2. Deterministic vs Stochastic

- When a uniqueness in the agent's current state completely determines the next state of the agent, the environment is said to be deterministic.
- The stochastic environment is random in nature which is not unique and cannot be completely determined by the agent.
- Examples:
- **Chess –** there would be only a few possible moves for a coin at the current state and these moves can be determined.
- **Self-Driving Cars-** the actions of a self-driving car are not unique, it varies time to time.

3. Competitive vs Collaborative

- An agent is said to be in a competitive environment when it competes against another agent to optimize the output.
- The game of chess is competitive as the agents compete with each other to win the game which is the output.
- An agent is said to be in a collaborative environment when multiple agents cooperate to produce the desired output.
- When multiple self-driving cars are found on the roads, they cooperate with each other to avoid collisions and reach their destination which is the output desired.

4. Single-agent vs Multi-agent

- An environment consisting of only one agent is said to be a single-agent environment.
- An environment involving more than one agent is a 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.
- The game of football is multi-agent as it involves 11 players in each team.

5. Dynamic vs Static

- An environment that keeps constantly changing itself when the agent is up with some action is said to be dynamic.
- A roller coaster ride is dynamic as it is set in motion and the environment keeps changing every instant.
- An idle environment with no change in its state is called a static environment.
- An empty house is static as there's no change in the surroundings when an agent enters.

6. Discrete vs Continuous

- If an environment consists of a finite number of actions that can be deliberated in the environment to obtain the output, it is said to be a discrete environment.
- The game of chess is discrete as it has only a finite number of moves. The number of moves might vary with every game, but still, it's finite.
- The environment in which the actions are performed cannot be numbered i.e. is not discrete, is said to be continuous.
- Self-driving cars are an example of continuous environments as their actions are driving, parking, etc. which cannot be numbered.

7.Episodic vs Sequential

- In **an Episodic task environment**, each of the agent's actions is divided into atomic incidents or episodes. There is no dependency between current and previous incidents. In each incident, an agent receives input from the environment and then performs the corresponding action.
- **Example:** Consider an example of **Pick and Place robot**, which is used to detect defective parts from the conveyor belts. Here, every time robot(agent) will make the decision on the current part i.e. there is no dependency between current and previous decisions.
- In a **Sequential environment**, the previous decisions can affect all future decisions. The next action of the agent depends on what action he has taken previously and what action he is supposed to take in the future.
- Example: • **Checkers-** Where the previous move can affect all the following moves

8. Known vs Unknown

- In a known environment, the output for all probable actions is given. Obviously, in case of unknown environment, for an agent to make a decision, it has to gain knowledge about how the environment works.

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.

Structure of an AI Agent

- The task of AI is to design an agent program which implements the agent function. The structure of an intelligent agent is a combination of architecture and agent program. It can be viewed as:
- 1. Agent = Architecture + Agent program

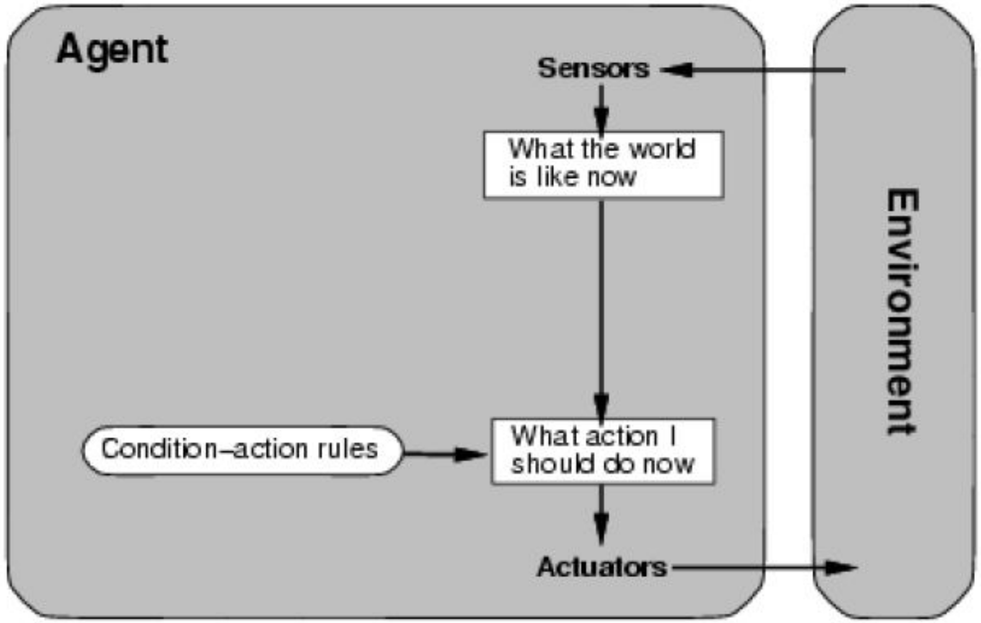
- Following are the main three terms involved in the structure of an AI agent

- **Architecture:** Architecture is machinery that an AI agent executes on.
- **Agent Function:** Agent function is used to map a percept to an action.
- 1. $f: P^* \rightarrow A$

- **Agent program:** Agent program is an implementation of agent function. An agent program executes on the physical architecture to produce function f.

- Types of Agent Programs
- Four basic kinds of agent programs that embody the principles underlying almost all intelligent systems:
  1. Simple reflex agents;
  2. Model-based reflex agents;
  3. Goal-based agents; and
  4. Utility-based agents
- Simple reflex agents
- Select actions on the basis of the current percept, ignoring the rest of the percept history
- Agents **do not have memory** of past world states or percepts.

- So, actions depend solely on **current percept**.
- Action becomes a “reflex.”
- Uses **condition-action rules**.



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.

- The INTERPRET-INPUT function generates an abstracted description of the current state from the percept, and. The RULE-MATCH function returns the first rule in the set of rules that matches the given state description. Note that the description in terms of “rules” and “matching” is purely conceptual; actual implementations can be as simple as a collection of logic gates implementing a Boolean circuit
- This will work *only if the correct decision can be made on the basis of only the current percept—that is, only if the environment is fully observable*

- Even a little bit of unobservability can cause serious trouble. For example, the braking rule given earlier assumes that the condition *car-in-front-is-braking* can be determined from the current percept—a single frame of video.
- This works if the car in front has a centrally mounted brake light. Infinite loops are often unavoidable for simple reflex agents operating in partially observable environments. Escape from infinite loops is possible if the agent can randomize its actions.

- **Model-based reflex agents**
- It works by finding a rule whose condition matches the current situation
- Key difference (wrt simple reflex agents):
  - Agents have **internal state**, which is used to keep track of past states of the world.
  - Agents have the ability **to represent change in the World**
- The current state is stored inside the agent which maintains some kind of structure describing the part of the world which cannot be seen.

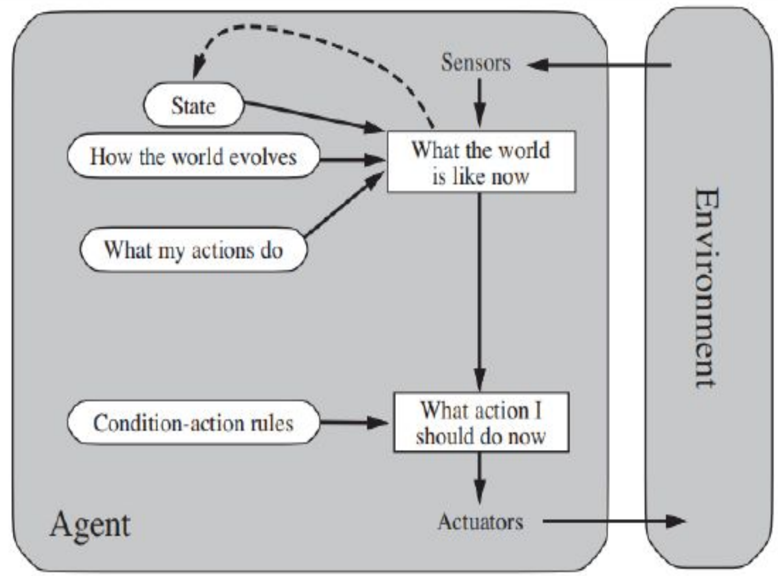


Figure 2.11 A model-based reflex agent.

Internal state information as time goes by requires two kinds of knowledge to be encoded in the agent program

1. we need some information about how the world evolves independently of the agent
2. we need some information about how the agent's own actions affect the world

Knowledge about "how the world works is called a **model** of the world. An agent that uses such a model is called a **model-based 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*

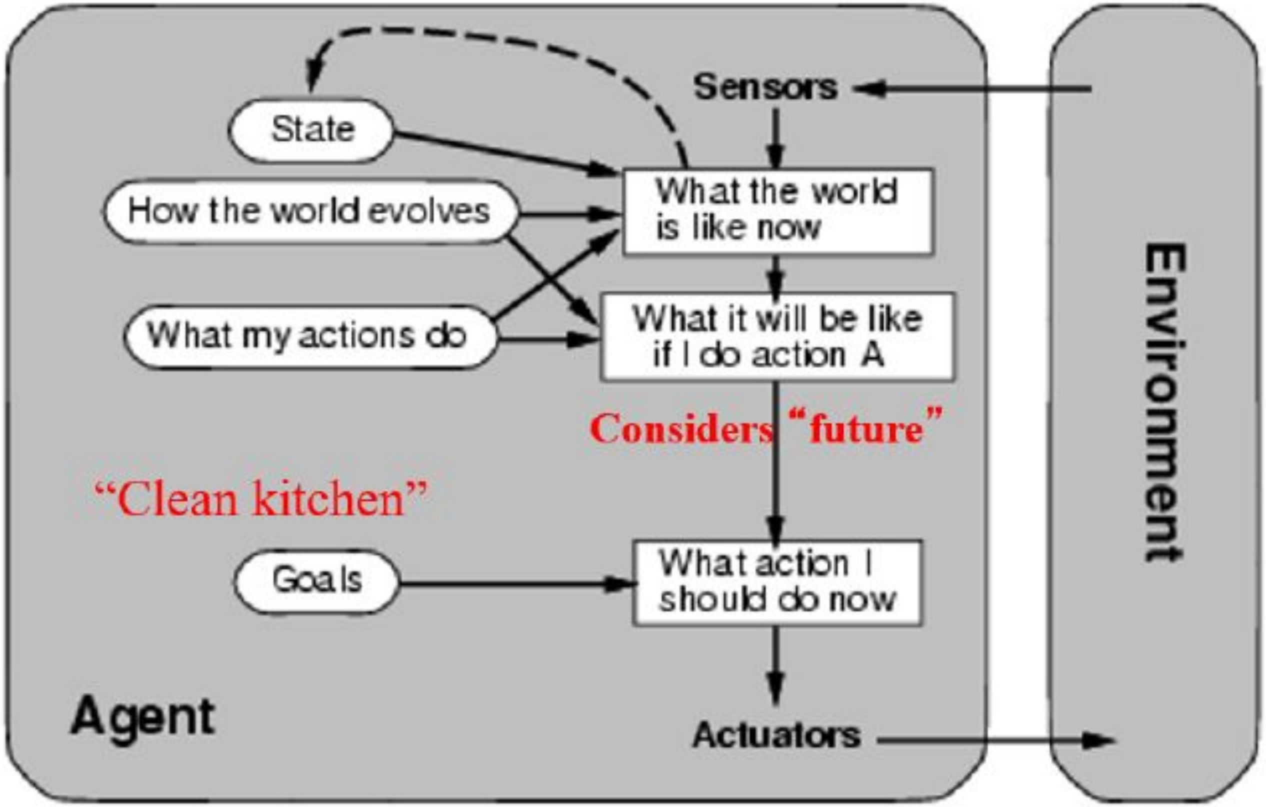
Figure 2.12 A model-based reflex agent. It keeps track of the current state of the world, using an internal model. It then chooses an action in the same way as the reflex agent.

UPDATE-STATE, which is responsible for creating the new internal state description.

- Goal-based agents
- These kinds of agents take decisions based on how far they are currently from their **goal**
- Key difference wrt Model-Based Agents:
- In addition to state information, have **goal information** that describes desirable situations to be achieved.
- Search and planning are the subfields of AI devoted to finding action sequences that achieve the agent's goals
- Agents of this kind take future events into consideration.
- **What *sequence* of actions can I take to achieve certain goals?**
- Choose actions so as to (eventually) achieve a (given or computed) goal

Module:
Problem Solving

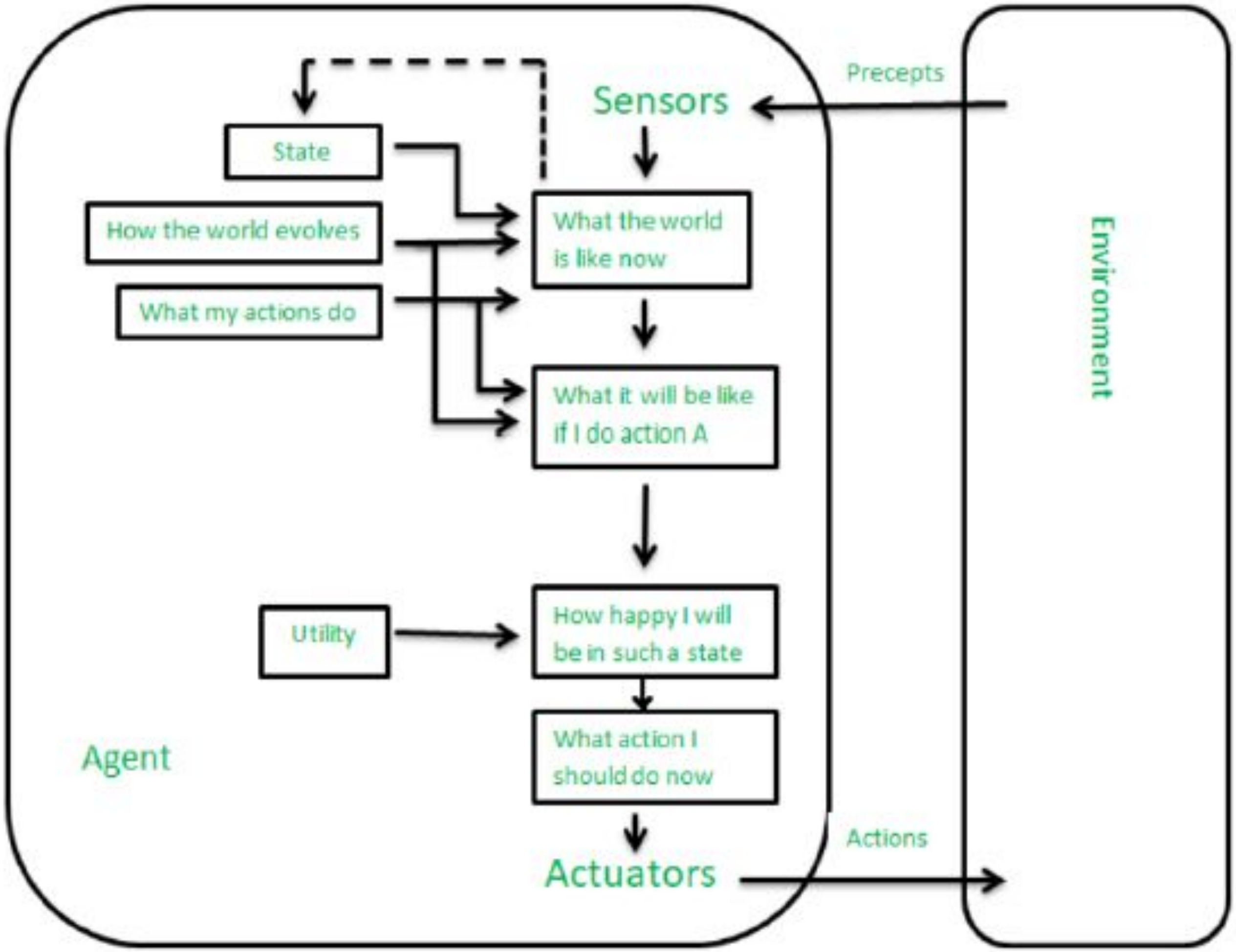
Goal-based agents



Agent keeps track of the world state as well as set of goals it's trying to achieve: chooses actions that will (eventually) lead to the goal(s).

More flexible than reflex agents → may involve search and planning

- <u>Utility-based agents</u>
- Goals alone are not enough to generate high-quality behavior in most environments. Goals just provide a crude binary distinction between "happy" and "unhappy" states. Because "happy" does not sound very scientific, economists and computer scientists use the term **utility** instead
- An agent's **utility function** is essentially an internalization of the performance measure. If the internal utility function and the external performance measure are in agreement, then an agent that chooses actions to maximize its utility will be rational according to the external performance measure.


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

- **Learning Agents**

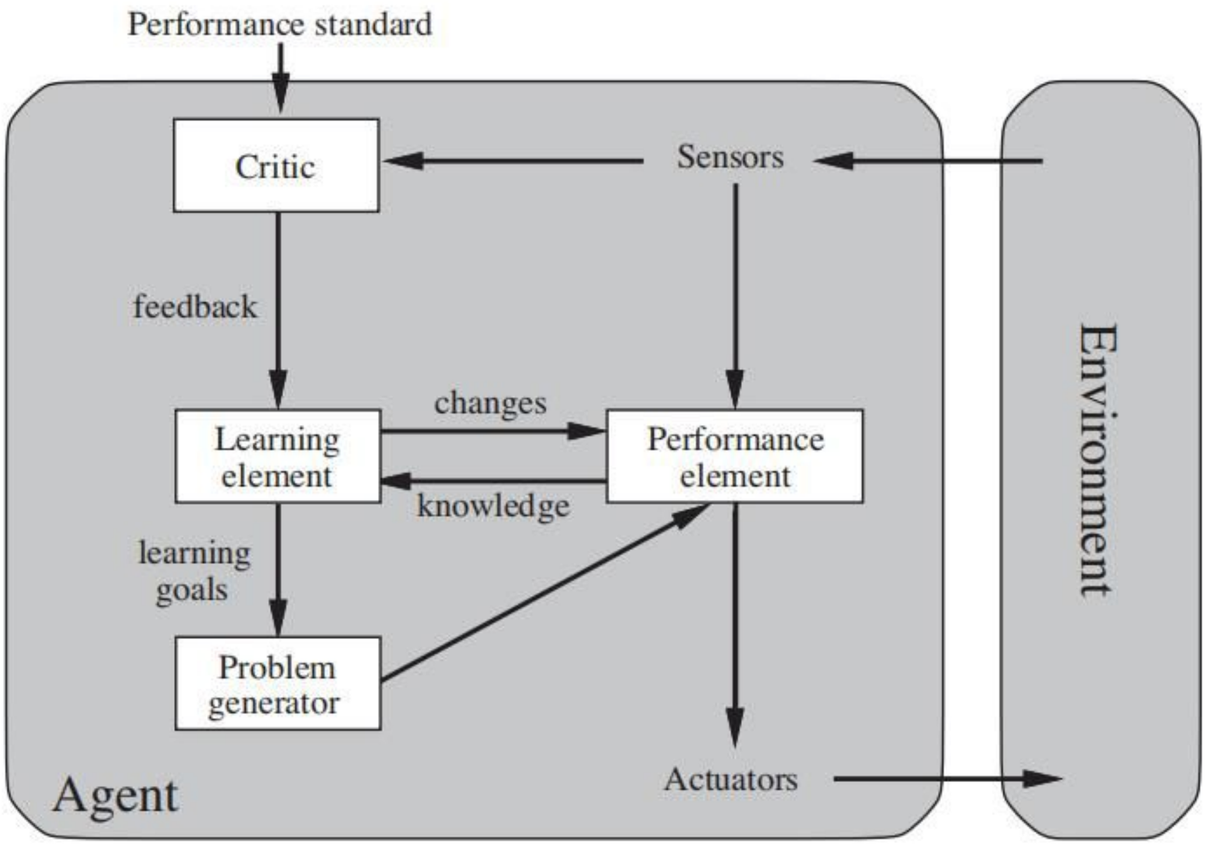


Figure 2.15 A general learning agent.

- A learning agent can be divided into four conceptual components

- **learning element**, which is responsible for making improvements
- **performance element**, which 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.
- **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
- **problem generator**. It is responsible for suggesting actions that will lead to new and informative experiences

How the components of agent programs work

- the various ways that the components can represent the environment that the agent inhabits.
- we can place the representations along an axis of increasing complexity and expressive power—**atomic**, **factored**, and **structured**
- In an **atomic representation** each state of the world is indivisible—it has no internal structure. For the purposes of solving this problem, it may suffice to reduce the state of world to just the name of the city we are in—a single atom of knowledge. The algorithms underlying **search** and **game-playing**, **HiddenMarkov models,**and **Markov decision processes** all work with atomic representations.
- A **factored representation** splits up each state into a fixed set of **variables** or **attributes**, each of which can have a **value**. With factored representations, we can also represent *uncertainty*—for example, ignorance about the amount of gas in the tank can be represented by leaving that attribute blank. Many important areas of AI are based on factored representations, including **constraint satisfaction** algorithms , **propositional logic** , **planning**, **Bayesian networks** , and the **machine learning** algorithms

- **In structured representation**, objects and their various and varying relationships can be described explicitly. Structured representations underlie **relational databases** and **first-order logic** ,**first-order probability models**, **knowledge-based learning.**

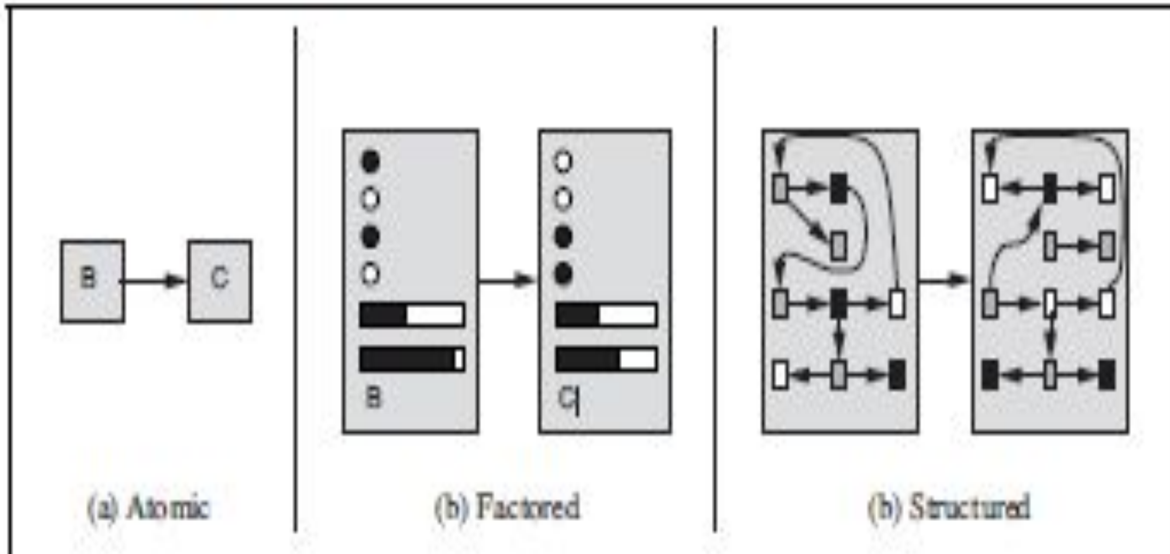


Figure 2.16 Three ways to represent states and the transitions between them. (a) Atomic representation: a state (such as B or C) is a black box with no internal structure; (b) Factored representation: a state consists of a vector of attribute values; values can be Boolean, real-valued, or one of a fixed set of symbols. (c) Structured representation: a state includes objects, each of which may have attributes of its own as well as relationships to other objects.