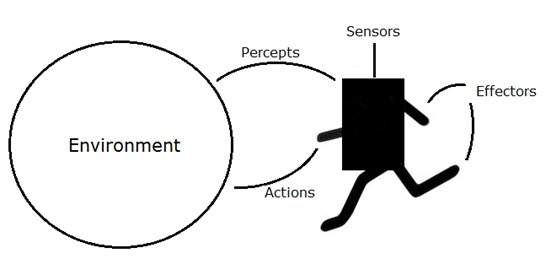
|  |
| --- |
| **Agents and Environment Types** |
|  |
|  |

An AI system is composed of an agent and its environment. The agents act in their environment. The environment may contain other agents.

**What are Agent and Environment?**

An **agent** is anything that can perceive its environment through **sensors** and acts upon that environment through **effectors.**

* A **human agent** has sensory organs such as eyes, ears, nose, tongue and skin parallel to the sensors, and other organs such as hands, legs, mouth, for effectors.
* A **robotic agent** replaces cameras and infrared range finders for the sensors, and various motors and actuators for effectors.
* A **software agent** has encoded bit strings as its programs and actions.



**Agent Terminology**

* **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.
* **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.

**Rationality**

Rationality is nothing but status of being reasonable, sensible, and having good sense of judgment. Rationality is concerned with expected actions and results depending upon what the agent has perceived. Performing actions with the aim of obtaining useful information is an important part of rationality.

**What is Ideal Rational Agent?**

An ideal rational agent is the one, which is capable of doing expected actions to maximize its performance measure, on the basis of −

* Its percept sequence
* Its built-in knowledge base

Rationality of an agent depends on the following −

* The **performance measures**, which determine the degree of success.
* Agent’s **Percept Sequence** till now.
* The agent’s **prior knowledge about the environment**.
* The **actions** that the agent can carry out.

A rational agent always performs right action, where the right action means the action that causes the agent to be most successful in the given percept sequence. The problem the agent solves is characterized by Performance Measure, Environment, Actuators, and Sensors (PEAS).

|  |
| --- |
| **Intelligent Agents - Agents and environments**  Structure of agents-Problem |
|  |

The Structure of Intelligent Agents

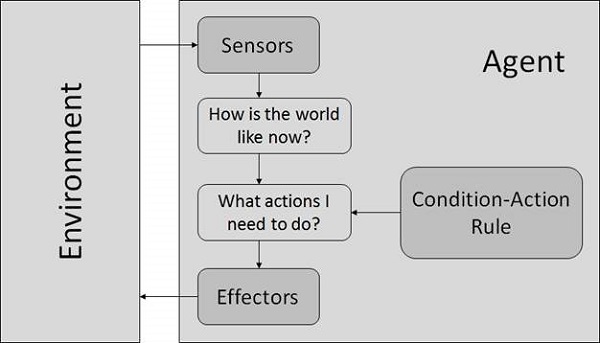
Agent’s structure can be viewed as −

* Agent = Architecture + Agent Program
* Architecture = the machinery that an agent executes on.
* Agent Program = an implementation of an agent function.

Simple Reflex Agents

* They choose actions only based on the current percept.
* They are rational only if a correct decision is made only on the basis of current precept.
* Their environment is completely observable.

**Condition-Action Rule** − It is a rule that maps a state (condition) to an action.



*Fig.-Model Based Reflex Agents.*

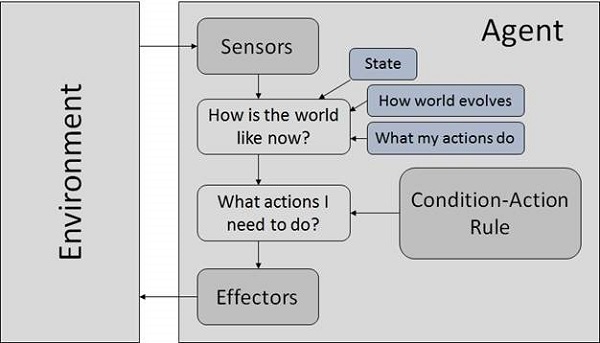
They use a model of the world to choose their actions. They maintain an internal state.

**Model** − knowledge about “how the things happen in the world”.

**Internal State** − It is a representation of unobserved aspects of current state depending on percept history.

**Updating the state requires the information about −**

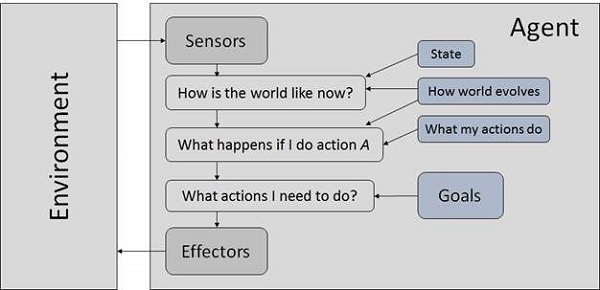
* How the world evolves.
* How the agent’s actions affect the world.



*Fig. - Goal Based Agents.*

They choose their actions in order to achieve goals. Goal-based approach is more flexible than reflex agent since the knowledge supporting a decision is explicitly modeled, thereby allowing for modifications.

**Goal** − It is the description of desirable situations.

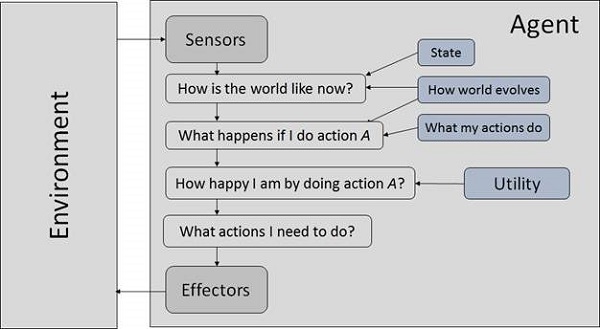


*Fig.-Utility Based Agents.*

They choose actions based on a preference (utility) for each state.

Goals are inadequate when −

* There are conflicting goals, out of which only few can be achieved.
* Goals have some uncertainty of being achieved and you need to weigh likelihood of success against the importance of a goal.



*Fig.-The Nature of Environments.*

Some programs operate in the entirely **artificial environment** confined to keyboard input, database, computer file systems and character output on a screen.

In contrast, some software agents (software robots or softbots) exist in rich, unlimited softbots domains. The simulator has a **very detailed, complex environment**. The software agent needs to choose from a long array of actions in real time. A softbot designed to scan the online preferences of the customer and show interesting items to the customer works in the **real** as well as an **artificial** environment.

The most famous **artificial environment** is the **Turing Test environment**, in which one real and other artificial agents are tested on equal ground. This is a very challenging environment as it is highly difficult for a software agent to perform as well as a human.

**Turing Test**

The success of an intelligent behavior of a system can be measured with Turing Test.

Two persons and a machine to be evaluated participate in the test. Out of the two persons, one plays the role of the tester. Each of them sits in different rooms. The tester is unaware of who is machine and who is a human. He interrogates the questions by typing and sending them to both intelligences, to which he receives typed responses.

This test aims at fooling the tester. If the tester fails to determine machine’s response from the human response, then the machine is said to be intelligent.

**Properties of Environment**

The environment has multifold properties −

* **Discrete / Continuous** − If there are a limited number of distinct, clearly defined, states of the environment, the environment is discrete (For example, chess); otherwise it is continuous (For example, driving).
* **Observable / Partially Observable** − If it is possible to determine the complete state of the environment at each time point from the percepts it is observable; otherwise it is only partially observable.
* **Static / Dynamic** − If the environment does not change while an agent is acting, then it is static; otherwise it is dynamic.
* **Single agent / Multiple agents** − The environment may contain other agents which may be of the same or different kind as that of the agent.
* **Accessible / Inaccessible** − If the agent’s sensory apparatus can have access to the complete state of the environment, then the environment is accessible to that agent.
* **Deterministic / Non-deterministic** − If the next state of the environment is completely determined by the current state and the actions of the agent, then the environment is deterministic; otherwise it is non-deterministic.
* **Episodic / Non-episodic** − In an episodic environment, each episode consists of the agent perceiving and then acting. The quality of its action depends just on the episode itself. Subsequent episodes do not depend on the actions in the previous episodes. Episodic environments are much simpler because the agent does not need to think ahead.

**Solving agents**

**Problem Solving agents** which is type of **goal based agent**. Since the direct mapping from states to actions of a simple reflex agent is too large to store for a complex environment, we use goal based agents that that can consider future actions and the desirability of outcomes.

# Problem Solving Agents

Intelligent agents are supposed to maximize its **performance measure**. Achieving this can be simplified if the agent can adopt a **goal** and aim to satisfy it.

Setting goals help the agent organize its behavior by limiting the objectives that the agent is trying to achieve and hence the actions it needs to consider. This **Goal formulation** based on the current situation and the agent’s performance measure is the first step in problem solving.

We consider the agent’s goal to be a set of states. The agent’s task is to find out actions in the present and in the future that could reach the goal state from the present state. **Problem formulation** is the process of deciding what actions and states to consider, given a goal.

“ An agent with several immediate options of unknown value can decide what to do by first examining the future actions that eventually lead to states of known value ”

After Goal formulation and problem formulation, the agent has to look for a sequence of actions that reaches the goal. This process is called **Search**. A search algorithm takes a problem as input and returns a sequence of actions as output.

After the search phase, the agent has to carry out the actions that are recommended by the search algorithm. This final phase is called**execution** phase.

**Formulate — Search — Execute**

Thus the agent has a formulate, search and execute design to it.

# Problems and Solutions

Before we get into more about problem formulating phase, we need to first understand what a problem is in terms of problem solving agents.

The problem can be defined formally in five components:

1. Initial State
2. Actions
3. Transition Model
4. Goal Test
5. Path Cost

## Initial State

The first component that describes the problem is the**initial state**that the agent starts in. For example, if a taxi agent needs to get to location(B) but the taxi is currently at location(A) then the initial state of the problem would be location(A).

## Actions

The second component that describes the problem is a description of the possible **actions** available to the agent. Given a state s, **Actions(s)** returns the set of actions that can be executed in s. We say that each of these actions is **applicable** in s.

## Transition Model

The third component is the description of what each action does which is called the **transition model**. It is specified by a function **Result(s , a)**that returns the state that results from doing action a in state s.

The initial state, actions and transition model together define the **state space** of a problem which is a set of all states reachable from the initial state by any sequence of actions. The state space forms a **graph** in which the nodes are states and the links between the nodes are actions.

## Goal Test

The goal test determines whether a given state is a goal state or not. Sometimes there is an explicit set of possible goal states and the test simply checks whether the given state is one of them. Sometimes the goal is specified by an abstract property rather than an explicitly enumerated set of states.

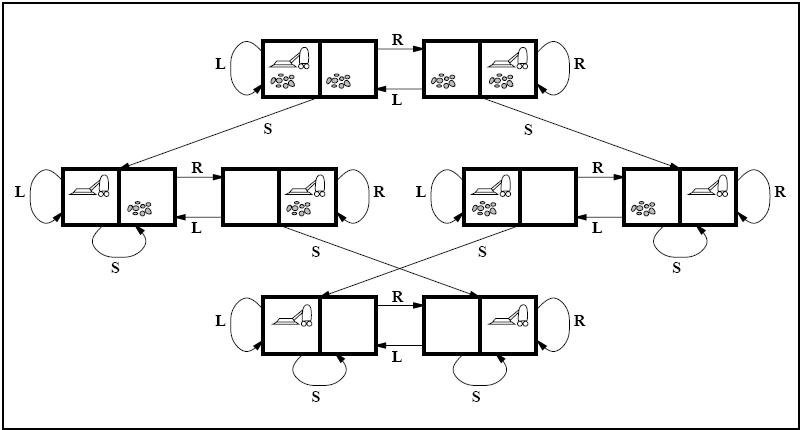
## Path Cost

The last component of the problem is the **path cost** which is a function that assigns a numeric cost to each path. The problem solving agent chooses a cost function that reflects its own performance measure.

The **solution** to the problem is an action sequence that leads from initial state to goal state and the solution quality is measured by the path cost function. An optimal solution has the lowest path cost among all the solutions.

# An Example Problem Formulation

Let us take the example of vacuum world that was introduced in the starting of this series, There is a vacuum cleaner agent and it can move left or right and its jump is to suck up the dirt from the floor.



State space for vacuum world.

The problem for vacuum world can be formulated as follows:

**States:**The state is determined by both the agent location and the dirt location. The agent is in one of two locations, each of which might or might not contain dirt. Therefore, there are 2 x 2**² = 8**possible world states.

A larger environment would have n x 2 to the power of n states.

**Initial State:**Any state can be assigned as the initial state in this case.

**Action:**In this environment there are three actions, Move Left , Move Right , Suck up the dirt.

**Transition Model:**All the actions have expected effects, except for when the agent is in leftmost square and the action is Left, when the agent is in rightmost square and the action is Right and the square is clean when the action is to Suck.

**Goal Test:**Goal test checks whether all the squares are clean.

**Path Cost:**Each step costs 1, so the path cost is the number of steps in the path.

The vacuum world problem is a toy problem and involves only discrete locations, discrete dirt etc. Therefore, this problem is a **Toy Problem**. There are many **Real-World Problems** like the automated taxi world. Try to formulate problems of real world and see what would be the states be and what actions could be chosen etc.