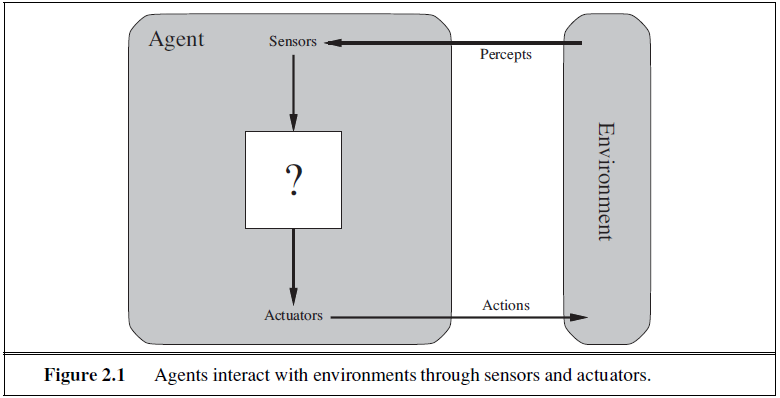
**Session-02**

**Intelligent Agents**

**Agent:**  An **agent** is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators. We use the term percept to refer to the agent's perceptual inputs at any given instant.



**Eg.** **1**: A human agent has eyes, ears, and other organs for sensors and hands, legs, vocal tract, and so on for actuators.

**Eg. 2**: A robotic agent might have cameras and infrared range finders for sensors and various motors for actuators**.**

**Eg. 3**: A software agent receives keystrokes, file contents, and network packets as sensory inputs and acts on the environment by displaying on the screen, writing files, and sending network packets.

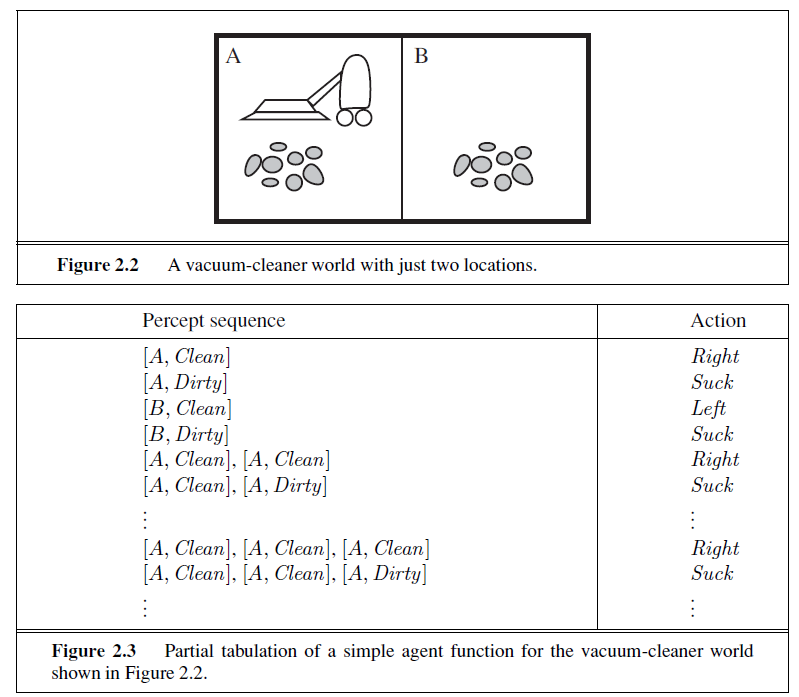
An agent's percept sequence is the complete history of everything the agent has ever perceived. In general, an agent's choice of action at any given instant can depend on the entire **percept sequence observed** to date, but not on anything it hasn't perceived.

Mathematically speaking, we say that an agent's behavior is **AGENT FUNCTION** described by the agent function that maps any given percept sequence to an action. We can imagine tabulating the agent function that describes any given agent; for most agents, this would be a very large table—infinite, in fact, unless we place a bound on the length of percept sequences we want to consider.

The table is, of course, an external characterization of the agent. Internally, the agent function for an artificial agent will be implemented by an **agent program**.

The difference between an agent function and program is that an agent function is an abstract mathematical description; the agent program is a concrete implementation, running within some physical system.

**Eg.** 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



**Good Behaviour:**

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

A 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.

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

First, we need to say what the performance measure is, what is known about the environment, and what sensors and actuators the agent has.

**Eg.** Vacuum Cleaner

* 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, 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.

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

**Autonomy** A rational agent should be autonomous—it should learn what it can to compensate for partial or incorrect prior knowledge.

After sufficient experience of its environment, the behavior of a rational agent can become effectively independent of its prior knowledge and that will succeed in a vast variety of environments.

**The Nature of Environments:**

**Task environments**, which are essentially the “problems” to which rational agents are the “solutions.” we had to specify the performance measure, the environment, and the agent’s actuators and sensors. We group all these under the heading of the **task environment**.

For the acronymically minded, we call this as the **PEAS** (**Performance, Environment, Actuators, Sensors**) description. In designing an agent, the first step must always be to specify the task environment as fully as possible.

The vacuum world was a simple example; let us consider a more complex problem: **an automated taxi driver**.

The full driving task is extremely open-ended. There is no limit to the novel combinations of circumstances that can arise.

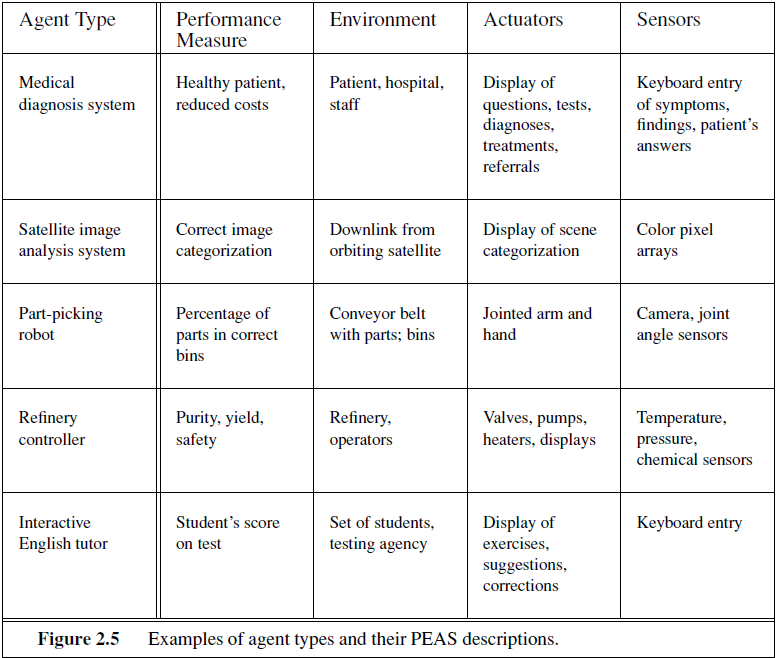
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Type Agent Type** | **Performance Measure**  **Measure** | **Environment** | **Actuators** | **Sensors** |
| Taxi driver | Safe, fast, legal,  comfortable trip,  maximize profits | Roads, other  traffic,  pedestrians,  customers | Steering,  accelerator,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

**Performance measure:** 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.

**Environment:** 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.

**Actuators:** include control over the engine through the accelerator and control over steering and braking. In addition, it will need output to a display screen or voice synthesizer to talk back to the passengers, and perhaps some way to communicate with other vehicles, politely.

**Sensors:** include one or more controllable video cameras so that it can see the road; it might augment these with infrared or sonar sensors to detect distances to other cars and obstacles. To avoid speeding tickets, the taxi should have a speedometer, and to control the vehicle properly, especially on curves, it should have an accelerometer.



In contrast, some **software agents**  (or software robots or **softbots**) exist in rich, unlim ited domains. Imagine a softbot Web site operator designed to scan Internet news sources and show the interesting items to its users, while selling advertising space to generate revenue.

To do well, that operator will need some natural language processing abilities, it will need to learn what each user and advertiser is interested in, and it will need to change its plans dynamically—for example, when the connection for one news source goes down or when a new one comes online. The Internet is an environment whose complexity rivals that of the physical world and whose inhabitants include many artificial and human agents.

**Different types of Enviornment:**

**Fully observable vs. partially Fully observable:**

If an agent’s sensors give it access to the complete state of the environment at each point in time, time, then we say that the task environment is **fully observable**.

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.

**Eg**. a vacuum agent with only a local dirt sensor cannot tell whether there is dirt in other squares.

**Single agent Vs. Multi agent:**

An environment in which only single agent exists is called a **Single Agent Environment.**

**Eg.** Crossword puzzle.

An environment in which more than one single agent exists is called a Multi-agent Environment.

**Eg**. Chess is a **competitive** multiagent environment because both the agents try to maximize their performance (win). In the taxi-driving environment, on the other hand, avoiding collisions maximizes the performance measure of all agents, so it is a **partially cooperative** multiagent environment.

**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; otherwise, it is stochastic. If the environment is partially observable, however, then it could appear to be stochastic.

**Eg.** The **vacuum world** as we described it is **deterministic**, but variations can include stochastic elements such as randomly appearing dirt and an unreliable suction mechanism.

**Taxi driving** is clearly **stochastic** in this sense, because one can never predict the behavior of traffic exactly. We say an environment is **uncertain** if it is not fully observable or not deterministic.

**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. Crucially, the next episode does not depend on the actions taken in previous episodes. Many classification tasks are episodic. Episodic environments are much simpler than sequential environments because the agent does not need to think ahead.

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

In **sequential** environments the current decision could affect all future decisions.

**Eg.** Chess and taxi driving.

**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**.

**Eg.** Taxi Driving is dynamic

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.

**Eg**. Crossword puzzle is static

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**.

**Eg.** Chess, when played with a clock, is semidynamic.

**Discrete vs. Continuous:**

The discrete/continuous distinction applies to the state of the environment, to the way time is handled, and to the percepts and actions of the agent.

**Eg.** Input from digital cameras is discrete; Taxi driving is a continuous-state

**Known vs. Unknown:**

This distinction refers not to the environment itself but to the agent’s state of knowledge about the environment. In a known environment the outcomes (or outcome probabilities if the

environment is stochastic) for all actions are given. Obviously, if the environment is unknown, the agent will have to learn how it works in order to make good decisions.

It is quite possible for a **known** environment to be **partially** **observable**—for example, in solitaire card games, we know the rules but still unable to see the cards that have not yet been turned over. Conversely, an **unknown** environment can be **fully observable**—in a new video game, the screen may show the entire game state but I still don’t know what the buttons do until we try them.

