ARTIFICIAL INTELLIGENCE (BCS515B)
Module1

Topics: Introduction: What Is AI? The State of the Art. Intelligent Agents: Agents and environment, Concept of Rationality, The nature of environment, The structure of agents.

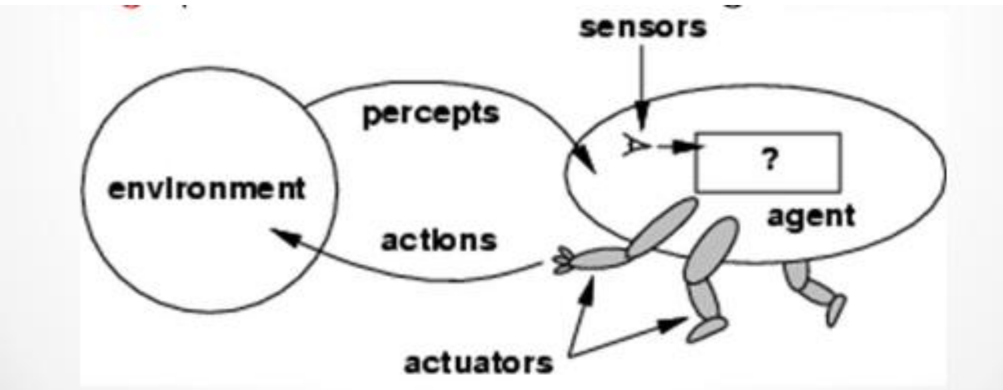
1.1 What is AI?

Artificial Intelligence (AI) is a field of computer science dedicated to develop systems capable of performing tasks that would typically require human intelligence. These tasks include **learning, reasoning, problem-solving, perception, understanding natural language**, and even interacting with the environment. AI aims to create machines or software that can mimic or simulate human cognitive functions.

According to Russell and Norvig, AI can be defined as follows:

" *AI (Artificial Intelligence) is the study of agents that perceive their environment, reason about it, and take actions to achieve goals. Each such agent implements a function that maps percept sequences to actions, and the study of these functions is the subject of AI.*"

Agents: In AI, an agent is any entity that can perceive its environment and take actions to achieve its goals. These agents can be physical robots, software programs, or any system capable of interacting with the world. The agents sense the environment through sensors and act on their environment through actuators. An AI agent can have mental properties such as knowledge, belief, intention, etc.



Source: The Intelligent Agent depicted on books

Percepts: Percepts are the inputs or information an agent receives from the environment. This could include data from sensors, cameras, microphones, or any other sources of information.

Actions: Actions are the responses or behaviours the agent can exhibit to achieve its goals or objectives. These can be physical movements, data processing, decision-making, or any other form of output.

Function: The central idea of AI is to design a function (or algorithm) that maps percept sequences to actions, allowing the agent to make intelligent decisions based on its observations.

Intelligent Agent: This term refers to the function that enables the agent to act in a way that is considered intelligent, i.e., making decisions that are adaptive and can lead to goal achievement.

Russell and Norvig's definition underscore the goal of AI, which is to create systems that can perceive their environment, reason about it, and take appropriate actions to achieve specific objectives. The field of AI encompasses a wide range of techniques and approaches, from symbolic reasoning to machine learning and deep learning, all aimed at building intelligent agents that can perform tasks typically associated with human intelligence

In **Figure,** there are **eight explanations** of **AI shown in two groups**. The top ones talk about thinking and reasoning, while the bottom ones focus on behavior. The left-side definitions judge based on how close to **human-like performance** they are, while the right-side definitions assess based on an ideal measure called **rationality**. A system is considered **rational** if it makes the best decisions based on the information it has.

Thinking Humanly "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)	**Thinking Rationally** "The study of mental faculties through the use of computational models." (Charniak and McDermott, 1985) "The study of the computations that make it possible to perceive, reason, and act." (Winston, 1992)
Acting Humanly "The art of creating machines that 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)	**Acting Rationally** "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)

Source Book: Stuart J. Russell and Peter Norvig, Artificial Intelligence, 3rd Edition, Pearson,2015

Let us look at the four approaches in more detail.

1. Acting humanly: The Turing Test approach

Turing Test : The Turing test is a method proposed by Alan Turing to evaluate a machine's capability to exhibit intelligent behavior that is indistinguishable from that of a human. In this test, a human evaluator engages in a conversation with both a human and a machine through text, without knowing which is which. If the evaluator cannot reliably differentiate between the machine's responses and the human's, the machine is said to have passed the Turing test, demonstrating human-like intelligence.

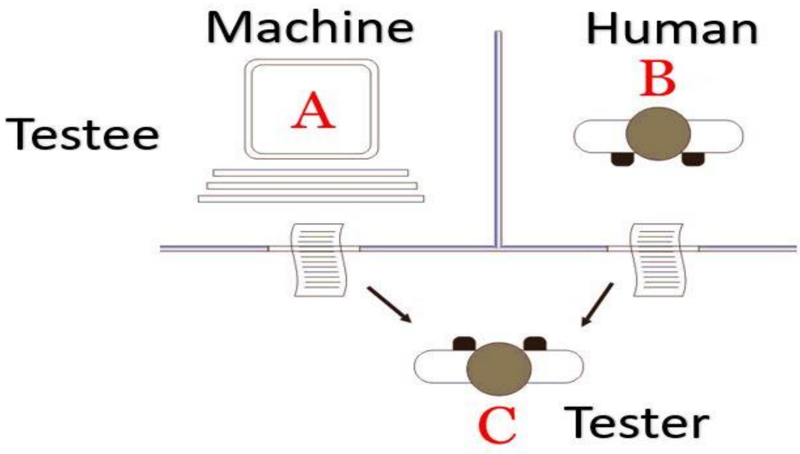


Image Source: https://www.how2shout.com/what-is/what-is-turing-test-and-it-used-for.html

In order to pass **Total Turing Test** the computer must have the following features:

1. **Proficiency in natural language processing** for effective communication in English.
2. **Capability for knowledge representation** to store acquired information.
3. **Automated reasoning to utilize stored data** for answering questions and deriving new conclusions.
4. **Machine learning to adapt to novel situations**, detect patterns, and extrapolate information.
5. **Computer vision** capabilities for object perception.
6. **Robotics proficiency** for manipulating objects and navigating its surroundings.

These **six** disciplines encompass the majority of artificial intelligence (AI), and Turing deserves recognition for designing a test that maintains its relevance six decades later.

2. Thinking humanly: The cognitive modeling approach

To ascertain whether a program exhibits human-like thinking, understanding human thought processes is essential. **This involves exploring the inner workings of the human mind through introspection, psychological experiments, and brain imaging**. Once a precise theory of the mind is established, it can be translated into a computer program. If the program's input-output behavior aligns with human behavior, it suggests shared mechanisms.

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

Aristotle, an early philosopher, sought to formalize "right thinking" through syllogisms, offering patterns for sound reasoning. These principles, foundational to logic, inspired logicians in the 19th century to develop precise notation for various objects and their relations. By 1965, programs capable of theoretically solving any problem in logical notation emerged, paving the way for the logicist tradition in artificial intelligence.

4. Acting rationally: The rational agent approach

Computer agents are expected to autonomously operate, perceive their environment, persist over time, adapt to change, and pursue goals. A rational agent seeks the best outcome by emphasizing correct inferences, aligning with the Turing Test skills. While correct inference is a facet of rationality, it's not exhaustive, as some situations lack provably correct actions. The rational-agent approach, incorporating knowledge representation, reasoning, and learning, offers generality and scientific advantages. Despite challenges acknowledged in the book, it adopts the hypothesis that perfect rationality is a useful starting point, simplifying complex environments for foundational discussions in the field.

Source Book: Stuart J. Russell and Peter Norvig, Artificial Intelligence, 3rd Edition, Pearson,2015

1.2 The State of the Art: What can AI do today?

It's hard to give a short answer because AI is used in so many different areas. Here are a few examples:

- **Robotic Cars:** In 2005, a driverless car named STANLEY drove through the Mojave Desert at 22 mph and won a race by completing a 132-mile course. It used cameras, radar, and lasers to see the environment and had software to control the steering, braking, and acceleration. In 2007, another car from CMU, called BOSS, safely drove through city streets in traffic while following traffic rules and avoiding other cars and pedestrians.
- **Speech Recognition:** You can book a flight with United Airlines by talking to an automated system that understands speech and responds accordingly.
- **Autonomous Planning and Scheduling:** NASA used AI to manage the tasks of a spacecraft far away from Earth. The AI planned and monitored its actions, making sure everything was running smoothly. Newer versions of this AI helped manage operations for the Mars rovers.
- **Game Playing:** IBM's DEEP BLUE was the first computer to beat the world chess champion, Garry Kasparov, in 1997. This was a huge moment for AI, as computers began to show a new kind of intelligence in games.
- **Spam Filtering:** Every day, AI filters out spam emails from your inbox. It learns and adapts to new tactics used by spammers, making it more effective than traditional methods.
- **Logistics Planning:** During the 1991 Persian Gulf crisis, AI was used to plan and schedule transportation for the U.S. military. It managed the movement of thousands of vehicles, cargo, and people in a way that would have taken much longer with older methods.
- **Robotics:** iRobot has sold over two million Roomba robotic vacuum cleaners for home use. Their PackBot robots are used in Iraq and Afghanistan to handle dangerous tasks like clearing explosives.
- **Machine Translation:** AI can translate languages, like turning Arabic into English. For example, AI helped translate an Arabic headline about Turkey and Cyprus without the programmers knowing Arabic—they just used statistics and machine learning.

These are just some examples of what AI can do today. It's not magic or science fiction—just a mix of science, engineering, and math.

1.4 Intelligent Agents

Topics: Agents and environment, Intelligent Agents, Concept of Rationality, The nature of environment, The structure of agents.

1.4.1 Agents and Environments:

An agent is defined as anything capable of perceiving its environment through sensors and acting upon that environment through actuators. This basic concept is depicted in Figure 2.1.

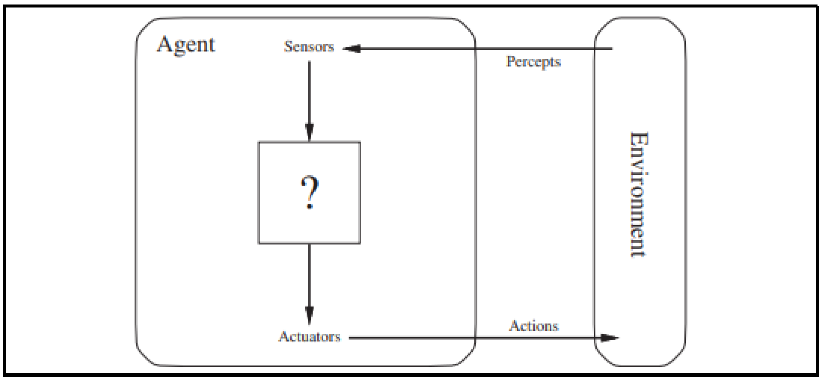


Figure 2.1 Agents interact with environments through sensors and actuators.

Source Book: Stuart J. Russell and Peter Norvig, Artificial Intelligence, 3rd Edition, Pearson,2015

For instance, a **human agent** employs eyes, ears, and other sensory organs while utilizing hands, legs, vocal tract, etc., as actuators.

In contrast, a **robotic agent** might use cameras and infrared range finders as sensors and various motors as actuators.

A **software agent** takes in sensory inputs such as keystrokes, file contents, and network packets and responds by displaying on the screen, writing files, or sending network packets.

Percept: The term **"percept"** refers to an agent's perceptual inputs at any given moment, and a percept sequence encompasses the complete history of everything the agent has perceived. Mathematically, an agent's behavior is described by the agent function, which maps a given percept sequence to an action.

Table: A table serves as an external representation of an agent's behaviour, specifically presented in tabular form that outlines the agent's actions for every possible percept sequences. The table contains entries corresponding to different percept sequences and the corresponding actions the agent would take in response to those sequences

Agent Program: An agent program represents the internal implementation of an agent's behavior. It is a concrete realization of the abstract agent function, running within a physical system. The agent program executes the specified actions determined by the agent function in response to the sensory inputs received by the agent. It involves the use of computational processes, algorithms, or code that dictates how the agent translates perceptual inputs into actions within its environment.

Example: Consider a simple example—the vacuum-cleaner world as depicted in **Figure 2.2**.

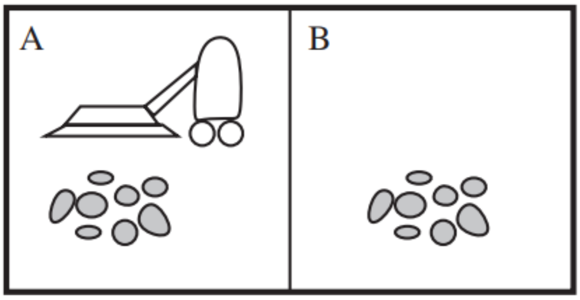


Figure 2.2 A vacuum-cleaner world with just two locations.

Source Book: Stuart J. Russell and Peter Norvig, Artificial Intelligence, 3rd Edition, Pearson,2015

This world consists of **two locations**, squares A and B, where the vacuum agent perceives its location and the presence of dirt. The agent can choose to move left, move right, suck up dirt, or do nothing. An example of a simple agent function is presented: if the current square is dirty, then suck; otherwise, move to the other square. **Figure 2.3** provides a partial tabulation of this agent function and **Figure 2.8** provides agent program.

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.

Source Book: Stuart J. Russell and Peter Norvig, Artificial Intelligence, 3rd Edition, Pearson,2015

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.

Source Book: Stuart J. Russell and Peter Norvig, Artificial Intelligence, 3rd Edition, Pearson,2015

1.4. 2: Intelligent Agents

In artificial intelligence (AI), intelligent agents are entities that perceive their environment and take actions to achieve specific goals. These agents are designed to operate autonomously, making decisions based on their perception of the world and their programming or learning mechanisms. The concept of intelligent agents is fundamental in AI as it provides a framework for building systems that can exhibit intelligent behaviour.

Key characteristics of intelligent agents include:

Perception: Agents have the ability to perceive or sense their environment. This involves gathering information from the surroundings through sensors or other means.
Action: Intelligent agents can take actions in response to their perceptions. These actions are chosen to influence the state of the environment or achieve specific goals.
Autonomy: Agents operate autonomously, making decisions and taking actions without direct human intervention. Autonomy allows them to adapt to changing conditions in their environment.
Goal-Directed Behaviour: Intelligent agents are typically designed to achieve specific goals. These goals can be predefined by a programmer or learned by the agent through experience.
Learning and Adaptation: Many intelligent agents have the capability to learn from experience and adapt their behaviour over time. This learning process may involve acquiring new knowledge or improving decision-making strategies.
Source Book: Stuart J. Russell and Peter Norvig, Artificial Intelligence, 3rd Edition, Pearson,2015

Communication: In some cases, intelligent agents can communicate with other agents or systems. Communication enables collaboration and coordination between multiple agents to achieve common goals.

1.4.3 The Concept of Rationality (Good behaviour)

A rational agent is defined as one that consistently does right thing and makes decisions leading to favourable outcomes. This is reflected in the **correct completion of every entry in the agent function table**. However, determining what constitutes the "**right thing**" involves a detailed consideration of consequences.

- **Assessing Consequences:** To evaluate the rationality of an agent's actions, one must examine the consequences of its behaviour. The agent's sequence of actions influences the environment, causing it to transition through different states. Desirability of this sequence is gauged by a performance measure that evaluates resultant environment states.
- **Performance Measure**: Performance measures play a pivotal role in evaluating an agent's behavior and defining success. These measures are instrumental in shaping the assessment of successful behavior, focusing specifically on the resulting environment states rather than the agent's subjective perception of its performance.

    - **Challenges and Considerations in Designing Performance Measures**: Designing a suitable performance measure is a multifaceted task that involves careful consideration of desired outcomes in the environment. This complex process requires avoiding potential pitfalls, such as the introduction of biased measures influenced by the agent's personal opinion.

        - **Example:** Taking the vacuum-cleaner agent as an example, the design of a performance measure becomes critical. Simply measuring dirt cleaned up in an eight-hour shift may lead to undesired behaviours, such as repetitive cleaning and dumping. A more effective measure would reward the agent for maintaining a clean floor.

- **Alignment with Environmental Goals**: Emphasizing the importance of aligning performance measures with actual environmental goals is crucial. This underscores the need to base measures on tangible objectives within the environment, steering clear of preconceived notions about how the agent should behave.

- **Philosophical Considerations**: Even with careful performance measure design, philosophical questions arise. Comparisons between agents with different approaches (e.g., a consistent but mediocre performer vs. an energetic but intermittently idle cleaner) prompt deeper reflections on preferences in various contexts.

1.4.4 What is Rationality?

Rationality encompasses the state of being reasonable, sensible, and possessing a sound judgment. **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.

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).

Rationality at Any Given Time depends on the following four things:

1. **Performance Measure**: The criterion defining success.
2. **Agent's Prior Knowledge**: Understanding of the environment.
3. **Available Actions**: The actions the agent can perform.
4. **Percept Sequence**: The agent's historical sensory input.

1.4.5 Omniscience, Learning, and Autonomy in Rational Agents

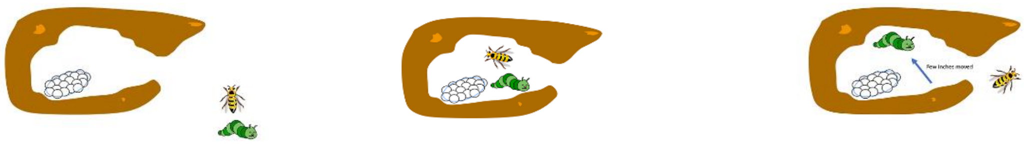
Omniscience and Rationality: Distinguishing between rationality and omniscience is crucial. An omniscient agent knows the actual outcome of its actions and can act accordingly; but omniscience is impossible in reality. While rationality maximizes expected performance.

Information Gathering : Rational agents should perform actions to modify future percepts, known as information gathering. This concept is vital for maximizing expected performance.

Learning in Rational Agents: Rational agents are not only expected to gather information but also to learn from their perceptions. An agent's initial configuration may reflect prior knowledge, but as it gains experience, learning becomes imperative.

*Think about the **dung beetle**, a small insect that carefully constructs its nest, lays eggs, and uses a ball of dung to seal the entrance. If the dung ball is taken away while the beetle is carrying it, the beetle continues its job as if the dung ball is still there, not realizing its missing. Evolution has shaped the beetle's behaviour to expect the dung ball, and any change from this expectation results in unsuccessful actions.*

*The **sphex wasp** is a bit smarter. It digs a hole, stings a caterpillar, drags it into the hole, checks everything is fine, and then lays eggs. The caterpillar becomes food for the hatching eggs. However, if someone moves the caterpillar a little while the wasp is checking, it goes back to dragging the caterpillar, not realizing the change. Even after many attempts to move the caterpillar, the wasp doesn't learn that its plan isn't working and keeps doing it the same way.*



Autonomy in Rational Agents: Autonomy is emphasized as a crucial aspect of rationality. An autonomous agent learns to compensate for **partial or incorrect prior knowledge**, ensuring effective adaptation to the environment. While acknowledging the practical need for some initial knowledge, over time, a rational agent's behavior becomes effectively independent of its initial knowledge through the incorporation of learning.

In essence, the combination of **rationality, information gathering, and learning enables the design of autonomous agents** capable of success across diverse environments.

1.4.4 The Nature of Environments

The task environment, in the context of rational agents, refers to the **external system or surroundings** within which an **agent** operates and performs tasks. It encompasses the conditions, challenges, and dynamics that influence the agent's behavior and decision-making processes. The task environment plays a pivotal role in shaping the **rationality, learning, and autonomy of an agent**.

Specifying the Task Environment

In the design of any agent, the initial crucial step is to comprehensively specify the "**task environment**," referred to as **PEAS** (Performance, Environment, Actuators, Sensors).

To illustrate the concept, consider the example of a more intricate problem: an **automated taxi driver**.

The PEAS description for the taxi's task environment is summarized in Figure 2.4, encompassing aspects such as performance measures, **driving environment, actuators, and sensors**.

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.

Source Book: Stuart J. Russell and Peter Norvig, Artificial Intelligence, 3rd Edition, Pearson,2015

Performance Measure: Determining the desired qualities for the automated driver, such as reaching the correct destination, minimizing fuel consumption, trip time, or cost, adhering to traffic laws, and maximizing safety and passenger comfort, involves addressing conflicting goals and necessitates trade-offs.

Driving Environment: The taxi must navigate a diverse range of roads, contend with various traffic conditions, interact with passengers, and make optional choices like operating in different regions with distinct weather conditions or driving on different sides of the road. The complexity of the environment directly impacts the design challenge.

Actuators and Sensors: Actuators for the automated taxi include control over the engine, steering, and braking, along with interfaces for communication with passengers and other vehicles. Sensors comprise controllable video cameras, infrared or sonar sensors, speedometer, accelerometer, and various vehicle state sensors.

Figure 2.5 outlines the basic PEAS elements for different agent types, showcasing the variety of considerations in specifying task environments.

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.

Source Book: Stuart J. Russell and Peter Norvig, Artificial Intelligence, 3rd Edition, Pearson,2015

1.4.5 Properties of Task Environments

In AI, there are various task environments, and we can classify them based on a few key dimensions as follows:

1. **Fully Observable vs. Partially Observable**:
   Fully Observable: The agent can see the complete environment all the time.
   Partially Observable: The agent's sensors may not capture all relevant aspects due to noise or missing information.

2. **Single Agent vs. Multiagent:**
   Single Agent: An agent operates independently, like solving a crossword puzzle.
   Multiagent: Agents interact with each other, introducing complexities in decision-making.

3. **Deterministic vs. Stochastic**:
   Deterministic: The environment's next state is entirely determined by the current state and the agent's action.
   Stochastic: Some uncertainty exists, making predictions challenging. For example, traffic behavior in taxi driving is stochastic.

4. **Episodic vs. Sequential:**
   Episodic: Each decision is independent, not influenced by past decisions.
   Sequential: Current decisions affect future ones, as seen in chess or taxi driving.

5. **Static vs. Dynamic**:
   Static: The environment remains unchanged during decision-making.
   Dynamic: The environment evolves continuously, demanding constant decision-making.

6. **Discrete vs. Continuous**:
   Discrete: Finite states, actions, or percepts.
   Continuous: Values or actions vary smoothly.

7. **Known vs. Unknown:**
   Known: The agent knows outcomes or probabilities.
   Unknown: The agent needs to learn about the environment.

Source Book: Stuart J. Russell and Peter Norvig, Artificial Intelligence, 3rd Edition, Pearson,2015

Understanding these dimensions helps in determining the complexity of an environment. For instance, *taxi driving is challenging due to being partially observable, multiagent, stochastic, sequential, dynamic, continuous, and unknown in many cases*.

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.

Source Book: Stuart J. Russell and Peter Norvig, Artificial Intelligence, 3rd Edition, Pearson,2015

Figure 2.6 lists the properties of a number of familiar environments.

1.4.6 The structure of agents

The structure of the agent comprises two essential components: the **architecture** and the **program**. The architecture refers to the combination of a computing device equipped with physical sensors and actuators. The program, created by AI, embodies the **agent function**, determining how the agent translates percepts into actions. The overall structure is denoted by the equation:

agent = architecture + program.

Agent Programs: All agent programs share a common structure: they take the current percept as input from the **sensors** and produce an action for the **actuators**.

It's crucial to note the distinction between the **agent program**, which operates based on the current percept, and the **agent function**, which considers the entire percept history. The agent programs are presented in a simple pseudocode language.

An example in **Fig 2.7** illustrates a basic agent program that tracks the percept sequence and uses it to determine the appropriate action from a table.

```
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.

Source Book: Stuart J. Russell and Peter Norvig, Artificial Intelligence, 3rd Edition, Pearson,2015

The table, exemplified for the vacuum world in **Figure 2.3**, explicitly represents the **agent function embodied** by the agent program.

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.

Source Book: Stuart J. Russell and Peter Norvig, Artificial Intelligence, 3rd Edition, Pearson,2015

Constructing a rational agent using this approach requires designers to create a table containing actions for every possible percept sequence. However, the table-driven approach faces insurmountable challenges. The size of the lookup table grows exponentially with the set of possible percepts and the agent's lifetime, making it impractical for complex environments. For instance, a lookup table for an automated taxi's visual input would be astronomically large. Despite these limitations, TABLE-DRIVEN-AGENT fulfils the objective of implementing the desired agent function.

Types of Agent Programs: There are, five basic types of agent programs as given below:

1. Simple reflex agents
2. Model-based reflex agents
3. Goal-based agents
4. Utility-based agents
5. Learning Agents

Each kind of agent program combines particular components in particular ways to generate actions.

1. **Simple Reflex Agents:**

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.

Example: For example, the vacuum agent whose agent function is tabulated in Figure 2.3 is a simple reflex agent, because its decision is *based only on the current location* and on whether that location contains dirt. An **agent program** for this agent is shown in Figure 2.8.

```
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.

Source Book: Stuart J. Russell and Peter Norvig, Artificial Intelligence, 3rd Edition, Pearson,2015

The program illustrated in Figure 2.8 is designed for a specific vacuum environment.

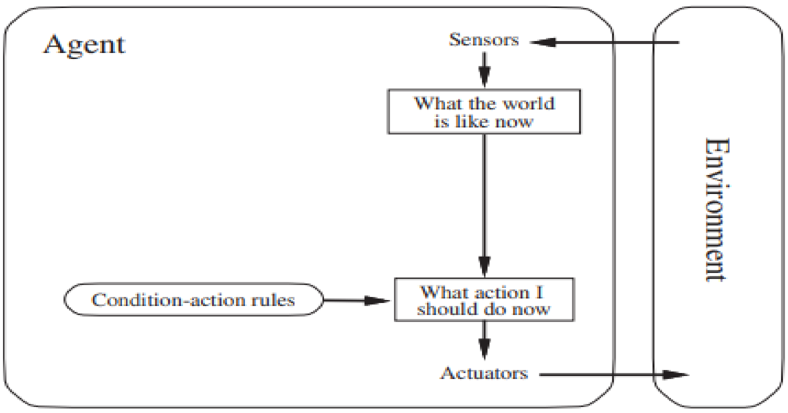


Figure 2.9 Schematic diagram of a simple reflex agent.

Source Book: Stuart J. Russell and Peter Norvig, Artificial Intelligence, 3rd Edition, Pearson,2015

Figure 2.9 provides a schematic representation of this generalized program, demonstrating how condition-action rules enable the agent to establish connections from perception to action. **Rectangles** represent the current internal state of the agent's decision process, while **ovals** depict background information used in the process.

```
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.

Source Book: Stuart J. Russell and Peter Norvig, Artificial Intelligence, 3rd Edition, Pearson,2015

The agent program, depicted in Figure 2.10, is also **straightforward**. The INTERPRET-INPUT function generates an abstracted description of the current state from the percept, and the RULE-MATCH function identifies the first matching rule in the rule set based on the given state description. Note that the conceptual terms **"rules" and "matching**" are used, but actual implementations can be as simple as a collection of logic gates constituting a Boolean circuit.

2. **Model based Reflex Agents:**

In order to address the challenges posed by partial observability, it is most effective for an agent to maintain an internal state that reflects the unobserved aspects of the current state. Updating this internal state requires encoding two types of knowledge in the **agent program**. **First**, knowledge about how the world evolves independently of the agent's actions, and **second**, information about how the agent's actions affect the world. This knowledge is referred to as a **"model of the world,"** and an agent using such a model is termed a model-based agent.

Figure 2.11 illustrates the structure of a model-based reflex agent with an internal state, where the current percept combines with the **old internal state** to generate an updated description of the current state based on the agent's model of the world.

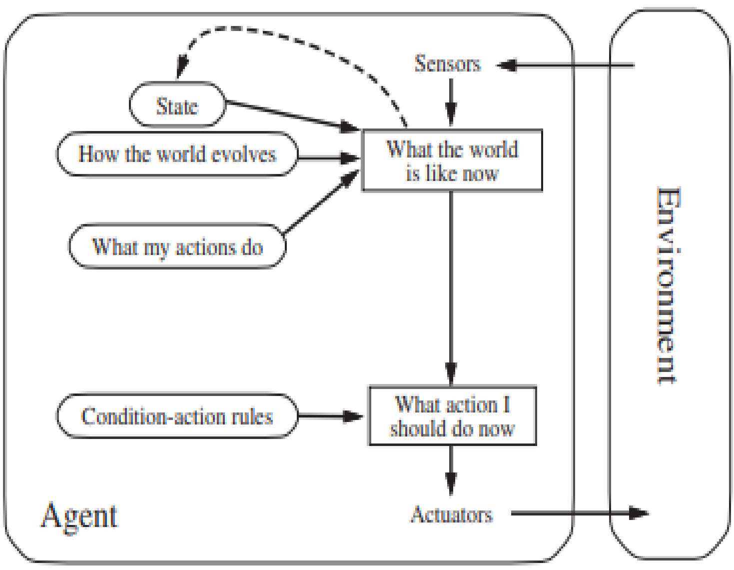


Figure 2.11 A model-based reflex agent.

Source Book: Stuart J. Russell and Peter Norvig, Artificial Intelligence, 3rd Edition, Pearson,2015

The agent program is shown in Figure 2.12. The UPDATE-STATE function is crucial in this process, responsible for creating the new internal state description.

```
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.

Source Book: Stuart J. Russell and Peter Norvig, Artificial Intelligence, 3rd Edition, Pearson,2015

3. **Goal-based agents :**

Deciding on actions based solely on the current state of the environment may not always be sufficient. The agent's program combines **the goal information with its model**, similar to the model-based reflex agent, to select actions that align with the goal. Figure 2.13 illustrates the structure of a goal-based agent.

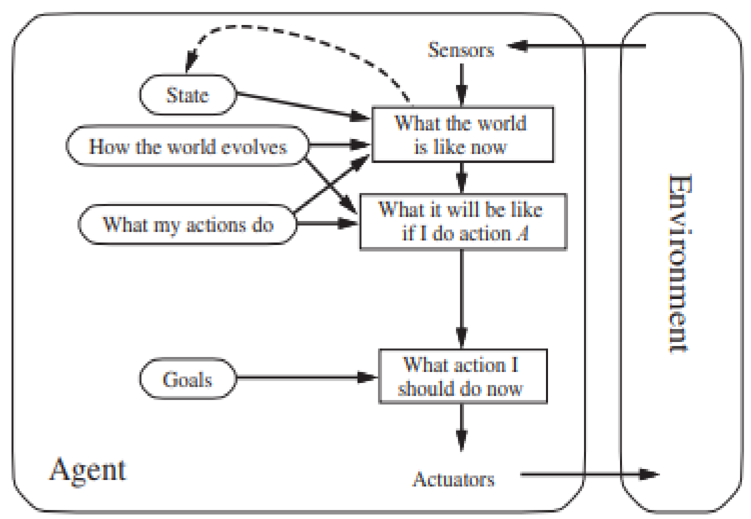


Figure 2.13 A model-based, goal-based agent. It keeps track of the world state as well as a set of goals it is trying to achieve, and chooses an action that will (eventually) lead to the achievement of its goals.

Source Book: Stuart J. Russell and Peter Norvig, Artificial Intelligence, 3rd Edition, Pearson,2015

4. **Utility based Agents**:

Achieving high-quality behaviour in various environments requires more than just setting goals. While goals indicate success or failure, they don't distinguish between different ways of achieving them in terms of efficiency, safety, reliability, or cost. A more detailed evaluation is needed, and this is where the concept of utility comes into play.

Utility, a term used by economists and computer scientists, represents a more refined measure of desirability than the binary distinction provided by goals. A utility function internalizes the performance measure, **assigning a score to different sequences of environmental state**s. If the internal utility function aligns with the external performance measure, an agent maximizing utility is considered rational. The utility-based agent structure is presented in Figure 2.14.

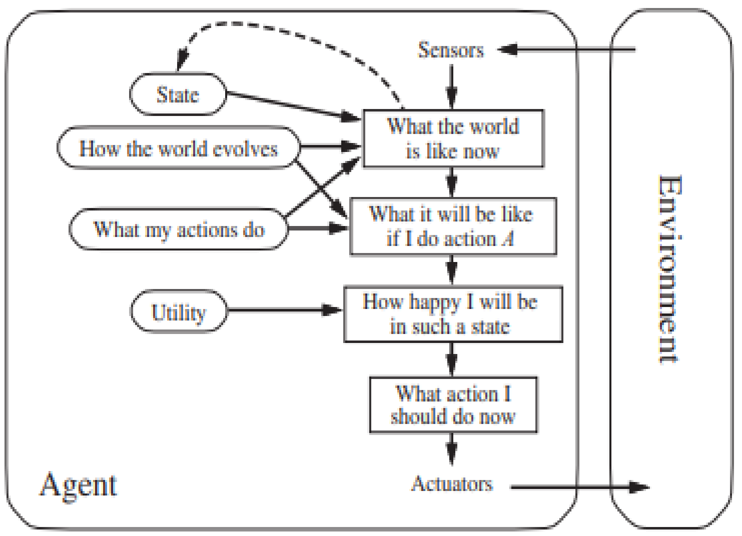


Figure 2.14 A model-based, utility-based agent. It uses a model of the world, along with a utility function that measures its preferences among states of the world. Then it chooses the action that leads to the best expected utility, where expected utility is computed by averaging over all possible outcome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

Source Book: Stuart J. Russell and Peter Norvig, Artificial Intelligence, 3rd Edition, Pearson,2015

5. Learning Agents

Learning offers the advantage of enabling agents to operate effectively in unfamiliar environments, surpassing their initial knowledge. A learning agent comprises **four conceptual components**, as illustrated in Figure 2.15. The crucial distinction lies between the **learning element**, responsible for improvements, and the **performance element**, tasked with selecting external actions. The learning element utilizes feedback from the **critic** to enhance the **performance element** for future improvement.

The critic informs the learning element of the agent's performance relative to a fixed standard. This standard is crucial, as percepts alone do not indicate the agent's success.

The problem generator, the last component, suggests actions for new and informative experiences. While the performance element tends to favor optimal actions based on current knowledge, the problem generator encourages exploration for potentially superior long-term actions.

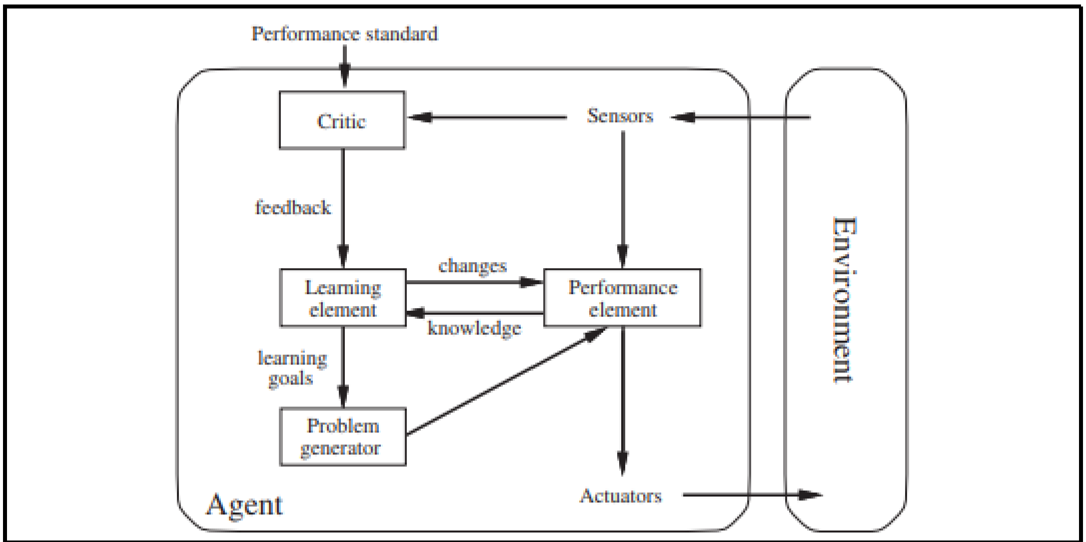


Figure 2.15 A general learning agent.

The automated taxi example: The performance element involves the knowledge and procedures guiding the taxi's driving actions, while the critic observes the world and provides feedback to the learning element. The learning element, in turn, formulates rules based on feedback, modifying the performance element accordingly. The problem generator suggests experiments to improve certain behaviors.

How the components of agent programs work

The representations of components of agents can be categorized along an axis of increasing complexity and expressive power: **atomic, factored, and structured**.

Atomic representations view each state of the world as indivisible, treating them as black boxes with no internal structure. This simplification is exemplified in algorithms like search, game-playing, Hidden Markov models, and Markov decision processes.

Factored representations, on the other hand, break down each state into a fixed set of variables or attributes, allowing shared attributes between different states. This facilitates dealing with uncertainty, as attributes can be left blank. Factored representations underlie various AI areas, including constraint satisfaction algorithms, propositional logic, planning, Bayesian networks, and machine learning algorithms.

Structured representations go beyond variables and values, explicitly describing objects and their relationships. This level of representation is essential for capturing complex scenarios, such as the interaction between a truck and a loose cow blocking its path. Structured representations are foundational to relational databases, first-order logic, first-order probability models, knowledge-based learning, and natural language understanding.

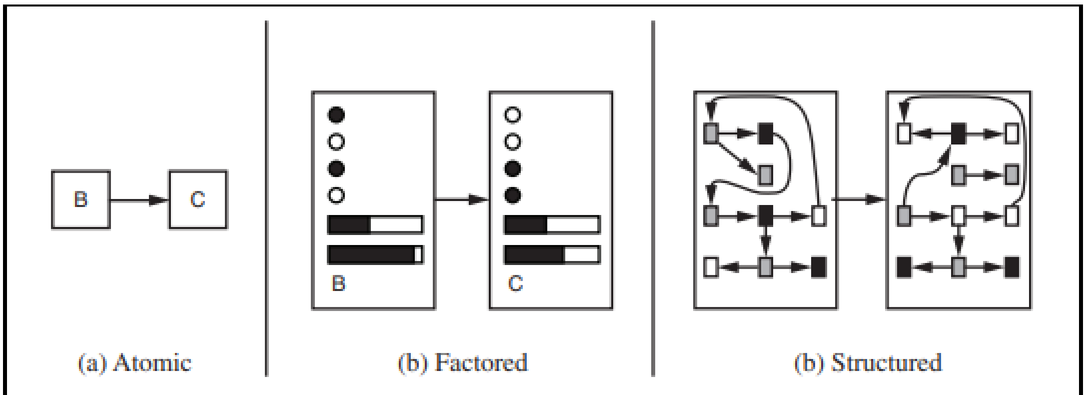


Figure 2.16 Three ways to represent states and the transitions between them. (a) Atomic representation: a state (such as B or C) is a black box with no internal structure; (b) Factored representation: a state consists of a vector of attribute values; values can be Boolean, real-valued, or one of a fixed 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.

Expressiveness is the key axis along which these representations lie, with increasing expressiveness allowing more concise representation of information. However, as expressiveness grows, reasoning and learning become more complex. Balancing the benefits and drawbacks of expressive representations, real-world intelligent systems may need to operate at various points along this axis simultaneously to effectively handle diverse scenarios.

Source Book: Stuart J. Russell and Peter Norvig, Artificial Intelligence, 3rd Edition, Pearson,2015