Artificial Intelligence

Shiraz Khurana

Introduction

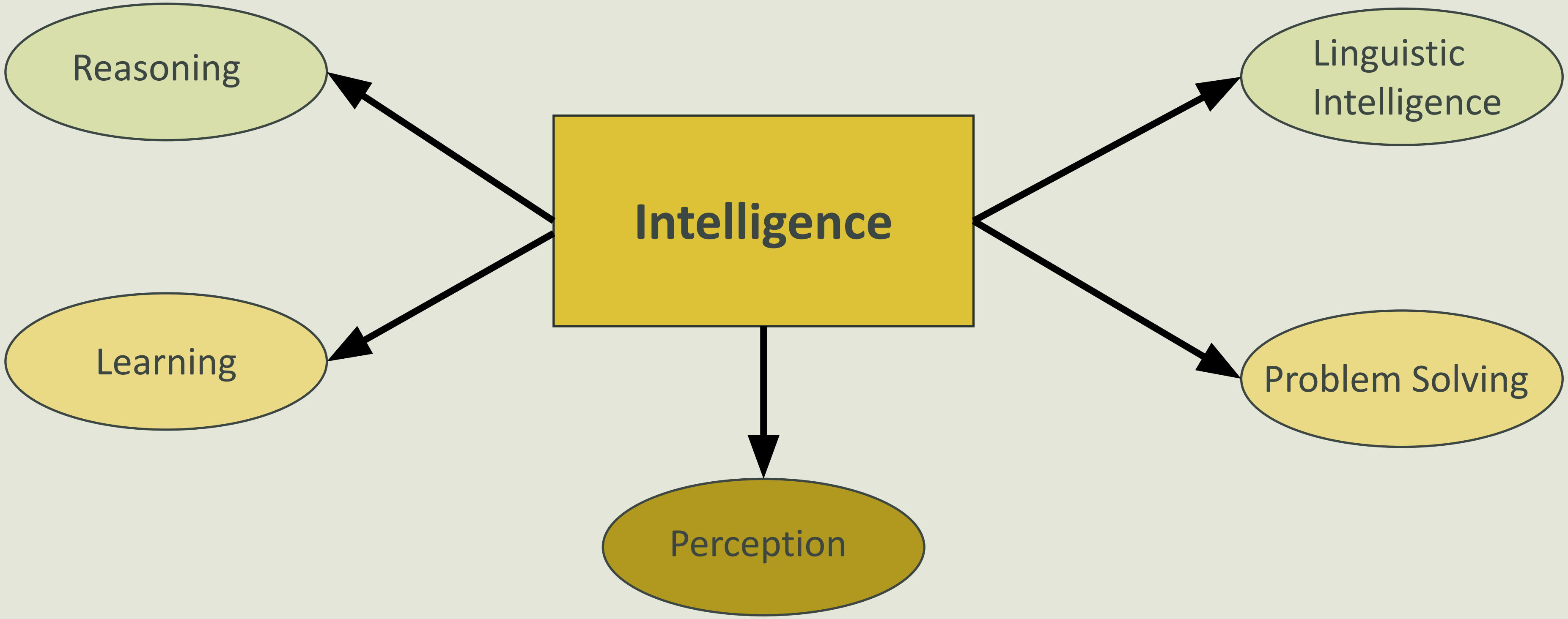
Artificial Intelligence

Man Made

Think, Understand & Learn

What is Intelligence: Intelligence is often defined as the general mental ability to learn and apply knowledge to manipulate your environment, as well as the ability to reason and have abstract thought.

Composition of Intelligence



What is Artificial Intelligence

- It is the branch of Computer Science by which we can create an intelligent machine which can behave like a human, think like human & able to make decision.
- the theory and development of computer systems able to perform tasks normally requiring human intelligence, such as visual perception, speech recognition, decision-making, and translation between languages.
- Artificial Intelligence is a way of making a computer, a computer-controlled robot, or a software think intelligently, in the similar manner the intelligent humans think.

Why we need Artificial Intelligence

- With the help AI:
  - Create Software / Devices which can solve real world problems very easily and with accuracy.
  - You can create virtual assistance
  - You can build machine which can work in an environment where survival of human can be at risk.

Goals of AI

- Replicate human intelligence in machines
- Solve knowledge intensive task
- An intelligent connection of perception and action
- Building a machine which can perform tasks that requires human intelligent such as:
  - Providing a theorem
  - Plan some surgical operation
- To create expert systems
- Create a System which can exhibit intelligent behaviour, learn new things by itself, demonstrate, explain and can advise to its user.

Applications of AI

- Problem Solving, Game Playing, Theorem Proving
- Natural Language Processing, Speech recognition, Pattern recognition
- Image Processing & Machine Translation
- Expert System medical science (disease detection, artificial muscles)
- Finance industry (international stock trade),
- Intelligent Transportation (driverless trains in Japan and UAE),
- Space exploration (NASA vehicle on Mars),

Applications of AI

- Data mining,
- Computer vision,
- pattern recognition (optical character recognition), handwriting recognition
- intelligent agents (smart home security systems),
- Robotics Vehicles
- super computing (Japanese K computer, American IBM Sequoia).
- Spam Fighting
- Logistic Planning

Future of AI

- AI based software will be used in the future to dramatically improve weather predictions for the benefit of forestry and agriculture;
- AI based devices and software will soon be used to clean up the environment and help identify patterns in our ecosystem and protect it;
- AI-enabled robots will be heavily used as substitutes in repetitive tasks in factories, in environments too dangerous for humans like radioactive materials, in land mines, deep sea exploration, and fire fighting;

Books

- 1. Stuart Russel, Peter Norvig “AI – A Modern Approach”, 2nd Edition, Pearson Education 2007.
- 2. Elaine Rich, “Artificial Intelligence”, 2nd Edition, McGraw Hill, 2005
- 3. Dan W.Patterson, “ Introduction to AI and ES”, Pearson Education, 2007
- 4. Peter Jackson,” Introduction to Expert Systems”, 3rd Edition, Pearson Education, 2007

Artificial Intelligence – Intelligent Agent

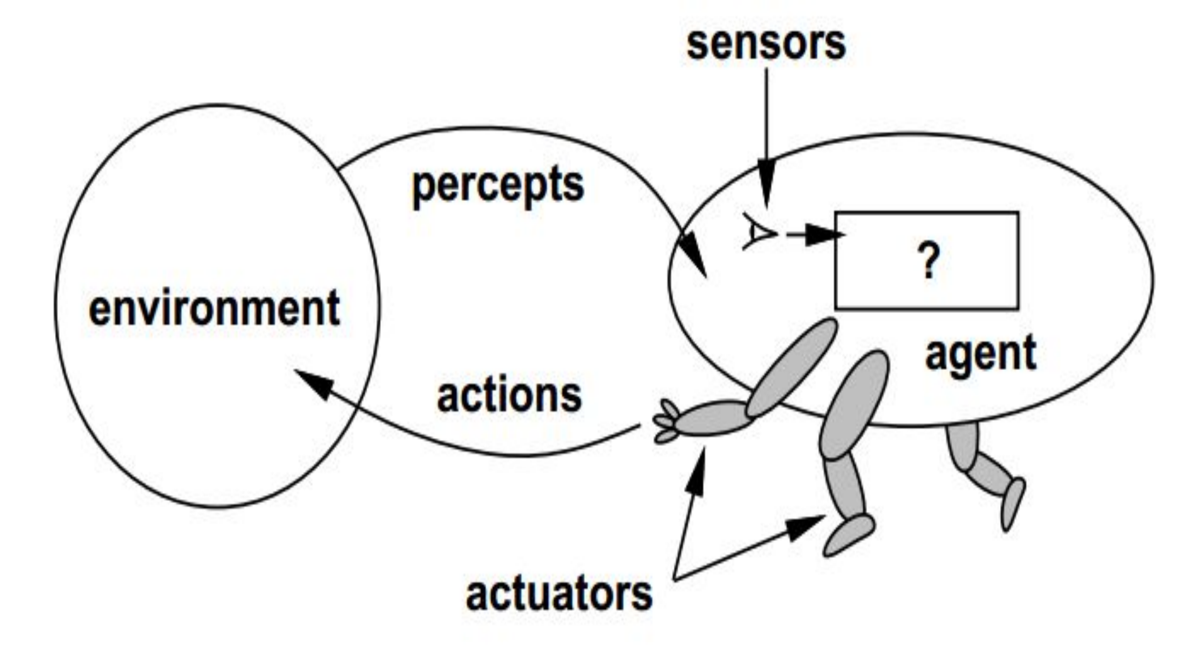
Shiraz Khurana

How AI must be Used

- Different people approach AI with different goals in mind. Two Important questions to ask are:
  - Are you concerned with thinking or behavior
  - Do you want to model humans or work from an ideal standard.
- Intelligence is concerned mainly with rational action. Ideally, an intelligent agent takes the best possible action in a situation.
- Philosopher, Mathematician, Economist, Neuroscientist, Psychologists, Computer Engineer.
- AI has advance more rapidly in the past decade because of greater use of scientific method in experimenting with and comparing approaches.

What is an Agent

- An agent is just something that acts. But computer agents are expected to have other attributes that distinguish them from ordinary programs.
- an agent is anything that can be viewed as perceiving its environment through sensor and acting upon that environment through actuators.

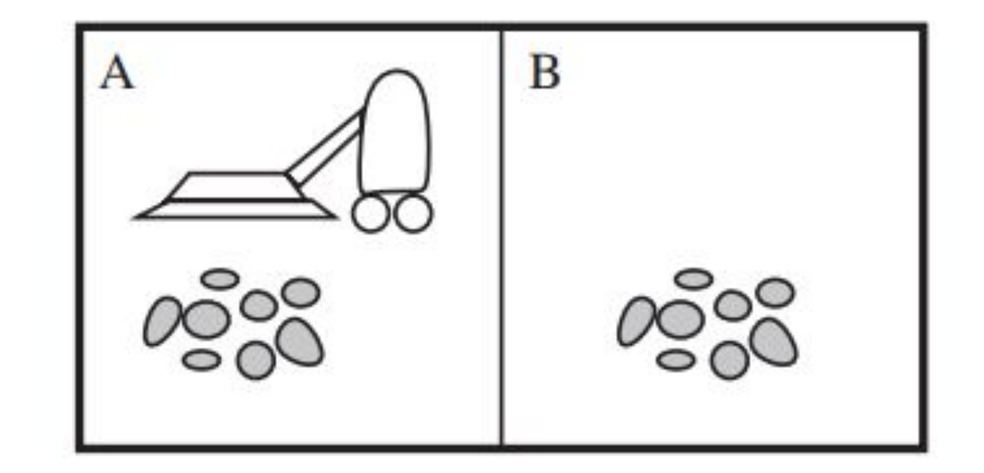


Agent & Environment

- **Percept:** it refers to the agent's perceptual inputs at any given instant.
- **Percept Sequence:** is the complete history of everything the agent has ever perceived.
  - 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.
- **Agent Function/Agent Program:** The agent function is an abstract mathematical description that maps any given percept sequence to an action; the agent program is a concrete implementation, running within some physical system.
- The agent function maps from percept histories to actions:
  $f : P* \longrightarrow A$
  The agent program runs on the physical architecture to produce f.

Vacuum Cleaner Agent Example

- 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.



Agent Function for Vacuum Cleaner

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*
$\vdots$	$\vdots$

The Concept of Rationality

- A **rational agent** is one that does the right thing.
- When an agent is placed in an environment, it generates a sequence of actions according to the percepts it receives. This sequence of actions causes the environment to go through a sequence of states. If the sequence is desirable, then the agent has performed well.
- The Performance can be Measured by some Performance Measures.
- 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.

The Concept of Rationality

- What is rational at any given time depends on four things:
  - The performance measure that defines the criterion of success.
  - The agent’s prior knowledge of the environment.
  - The actions that the agent can perform.
  - The agent’s percept sequence to date.
- 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.

Vacuum Cleaner: Let have Some Assumptions

- 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.

The Concept of Rationality Continues…

- There is Difference Between Rationality & Omniscience(Perfection).
- Rationality maximizes expected performance, while perfection maximizes actual performance.
- Information Gathering & Exploration: Doing actions in order to modify future percepts.
- Learning: a rational agent not only to gather information but also to learn as much as possible from what it perceives.
- Autonomy: To the extent that an agent relies on the prior knowledge of its designer rather than on its own percepts, we say that the agent lacks **autonomy**. A rational agent should be autonomous it should learn what it can to compensate for partial or incorrect prior knowledge.

THE NATURE OF ENVIRONMENTS

- Task Environments are the “problems” to which rational agents are the “solutions.”
- In designing an agent, the first step must always be to specify the task environment as fully as possible.

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

THE NATURE OF 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

Properties of Task Environment

- **Fully observable vs. partially observable:**
- If an agent's sensors give it access to the complete state of the environment at each point in time, then we say that the task environment is fully observable.
- 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.
- Unobservable: If the agent has no sensors at all then the environment is unobservable.

Properties of Task Environment

- **Single agent vs. Multi-agent**
- **Competitive multi-agent environment or Cooperative multi-agent 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.
- Uncertain**:** if it is not fully observable or not deterministic.

Properties of Task Environment

- **Episodic vs. Sequential**
- In an episodic task environment, the agent’s experience is divided into atomic episodes.
- In sequential environment the current decision could affect all future decisions.
- **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.
- 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 semi-dynamic.
- **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.

Properties of Task Environment

- **This is not a Strict Property of Environment**
- **Known** vs. **unknown**: In a known environment, the outcomes for all actions are given. 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
- An unknown environment can be fully observable

Let’s Revise through an Example

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

Structure of Agent

- *agent = architecture + program*
- The job of AI is to design an **agent program** that implements the agent function the mapping from precepts to actions.
- this program will run on some sort of computing device with physical sensors and actuators—**architecture**.
- The difference between the agent program, which takes the current percept as input, and the agent function, which takes the entire percept history.

Table Driven Agent

```
function TABLE-DRIVEN-AGENT(percept) returns an action
  persistent: percepts, a sequence, initially empty
              table, a table of actions, indexed by percept sequences, initially fully specified

  append percept to the end of percepts
  action ← LOOKUP(percepts, table)
  return action
```

Issues with table driven approach

- No physical agent in this universe will have the space to store the table,
- The designer would not have time to create the table,
- No agent could ever learn all the right table entries from its experience, and
- even if the environment is simples enough to yield a feasible table size, the designer can not fill in the table entries.

Agent Types

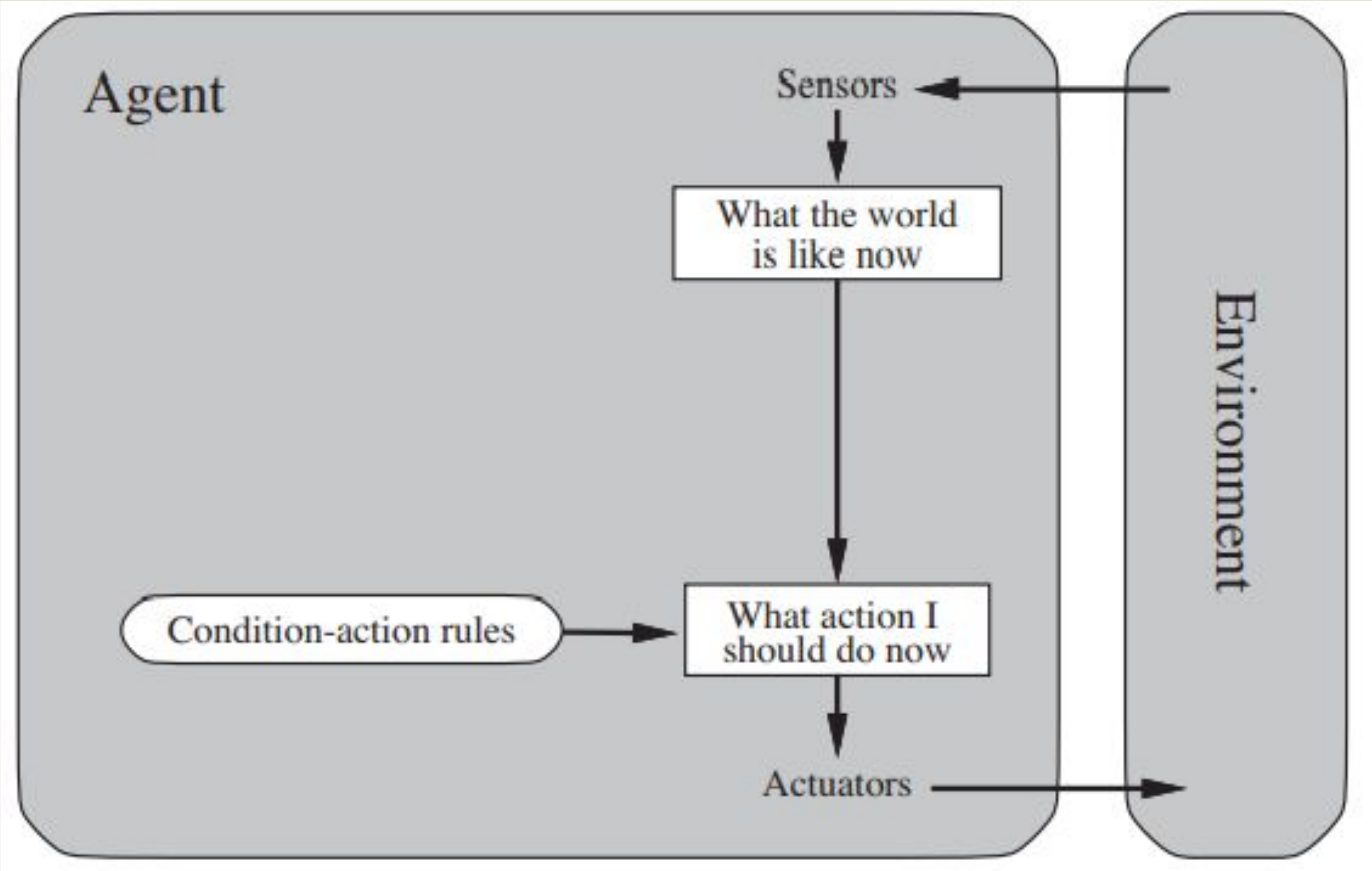
- Simple reflex Agents
- Model Based reflex Agents
- Goal-Based Agents
- Utility-Based Agents

Simple Reflex Agents

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*

Simple Reflex Agents



Simple Reflex Agents

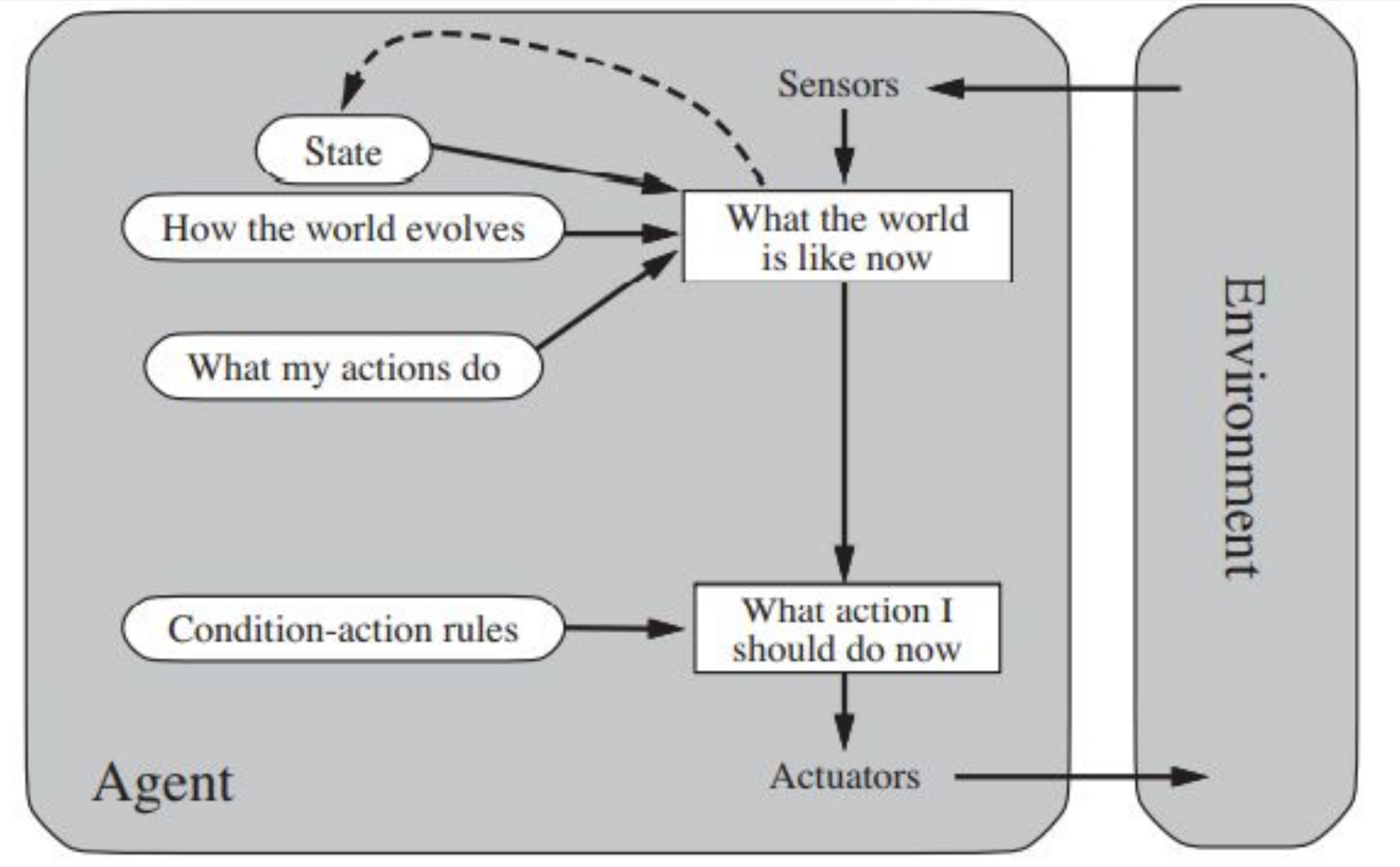
```
function SIMPLE-REFLEX-AGENT(percept) returns an action
  persistent: rules, a set of condition–action rules

  state ← INTERPRET-INPUT(percept)
  rule ← RULE-MATCH(state, rules)
  action ← rule.ACTION
  return action
```

Model Based reflex agents

- The agent should maintain some sort of **internal state** that depends on the percept history and thereby reflects at least some of the unobserved aspects of the current state.
- Updating this internal state requires two kinds of knowledge to be encoded in the agent program.
- how the world evolves independently of the agent.
- how the agent’s own actions affect the world.
- This knowledge about “how the world works”—whether implemented in simple Boolean circuits or in complete scientific theories—is called a **model** of the world. An agent that uses such a model is called a **model-based agent**.

Model Based Agent



Model Based Agent

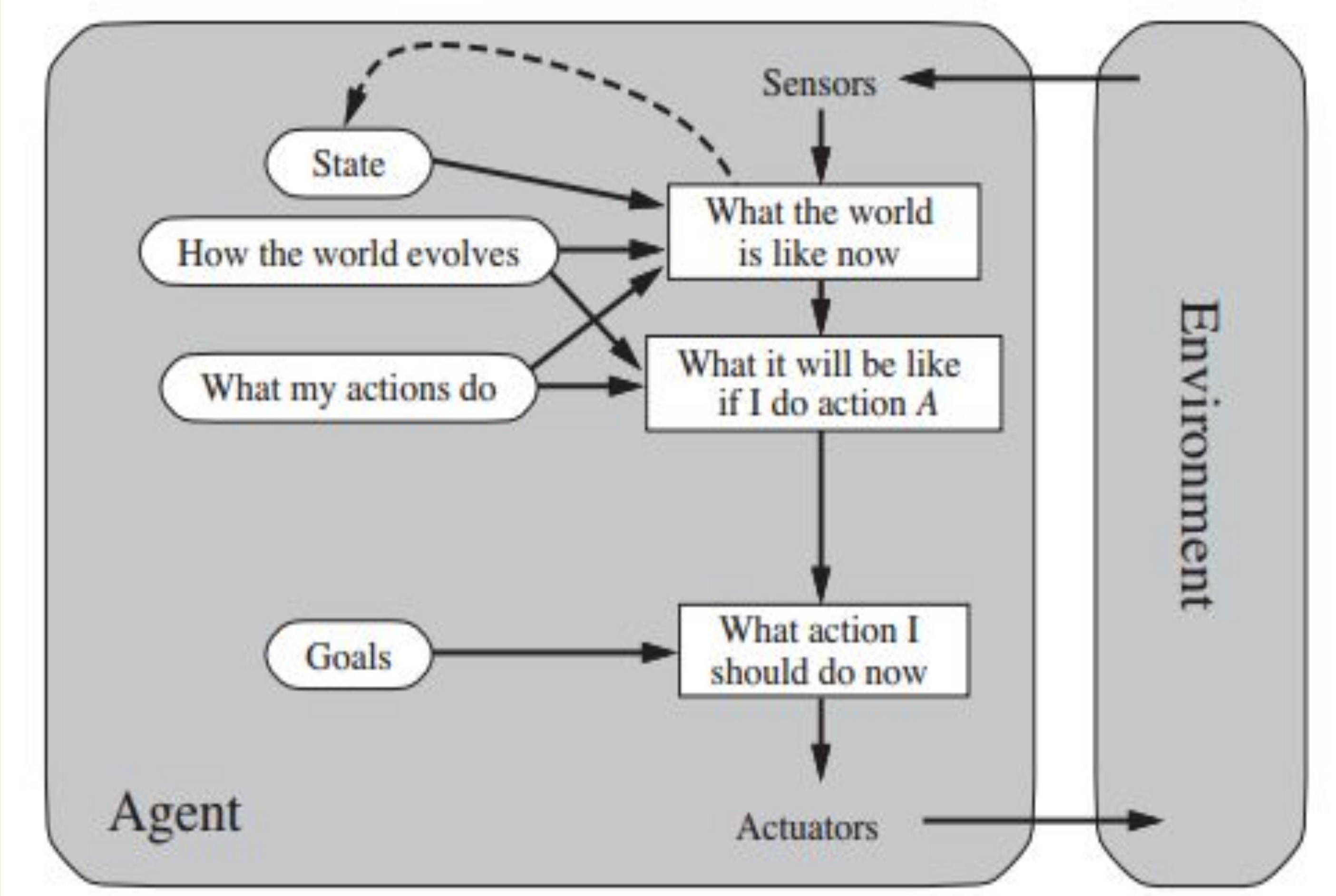
function MODEL-BASED-REFLEX-AGENT(*percept*) **returns** an action
  persistent: *state*, the agent's current conception of the world state
    model, a description of how the next state depends on current state and action
    rules, a set of condition–action rules
    action, the most recent action, initially none

  state ← UPDATE-STATE(*state*, *action*, *percept*, *model*)
  rule ← RULE-MATCH(*state*, *rules*)
  action ← *rule*.ACTION
  return *action*

Goal Based Agent

- Knowing something about the current state of the environment is not always enough to decide what to do.
- The agent program can combine this with the model (the same information as was used in the model based reflex agent) to choose actions that achieve the goal.
- Sometimes goal-based action selection is straightforward—for example, when goal satisfaction results immediately from a single action. Sometimes it will be tricky—for example, when the agent has to consider long sequences of twists and turns in order to find a way to achieve the goal.

Goal Based Agent



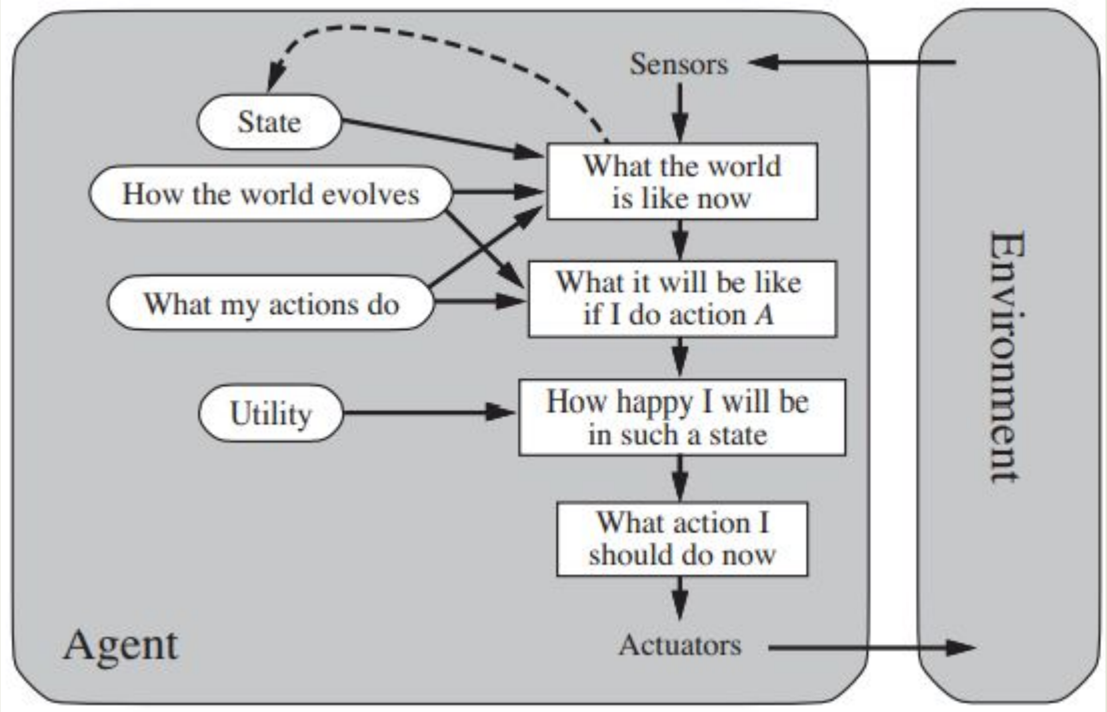
Goal Based Agent

- Although the goal-based agent appears less efficient, it is more flexible because the knowledge that supports its decisions is represented explicitly and can be modified.
- The goal-based agent’s behavior can easily be changed to go to a different destination, simply by specifying that destination as the goal. The reflex agent’s rules for when to turn and when to go straight will work only for a single destination; they must all be replaced to go somewhere new.

Utility-Based Agents

- 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.

Utility Based Agent



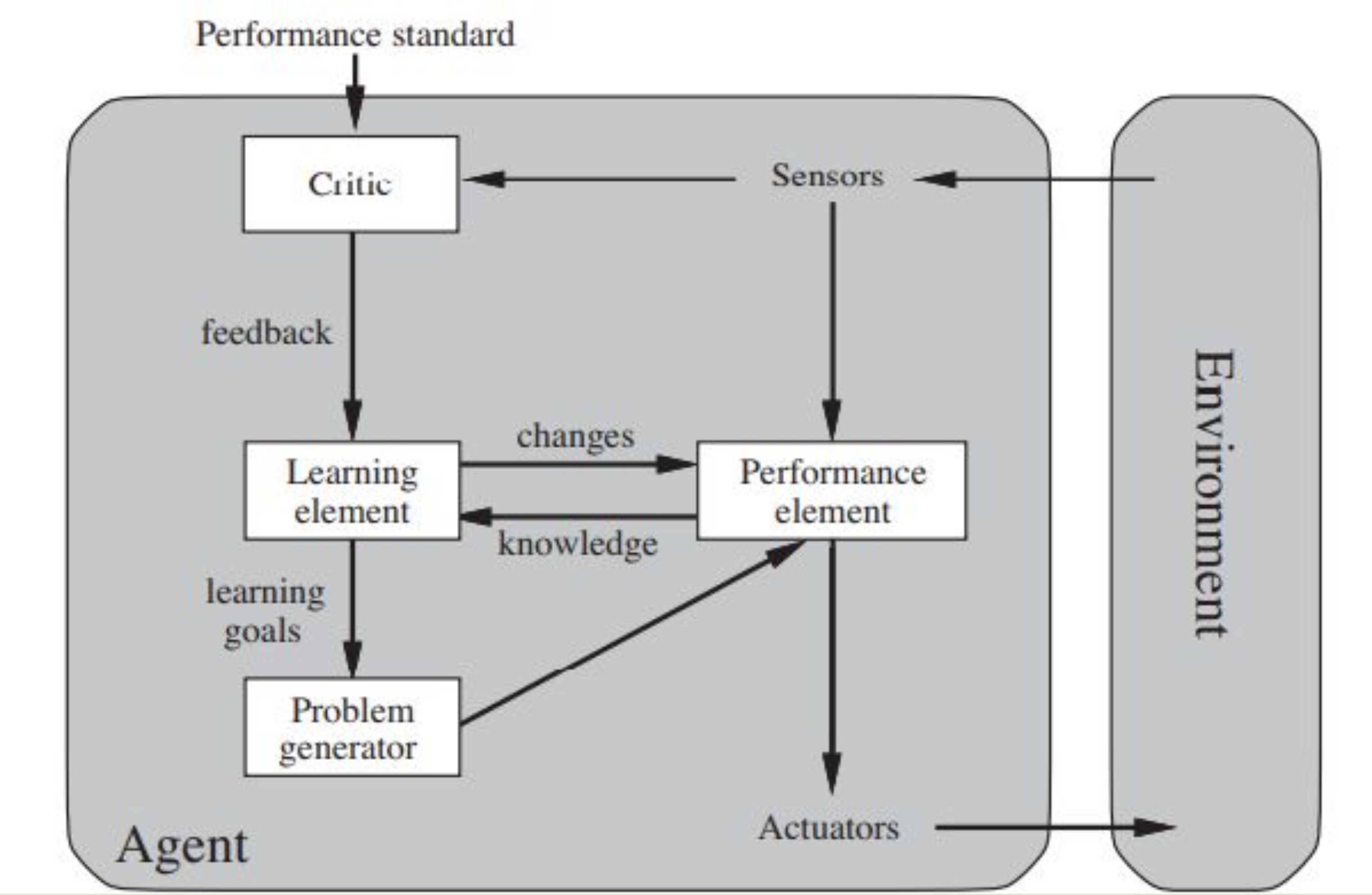
Learning Agent

- A learning agent can be divided into four conceptual components:
  - Learning Element: Responsible for making improvements
  - Performance Element: Responsible for selecting external actions. The performance element is the entire agent: it takes in percepts and decides on actions.
  - Critic: . The learning element uses feedback from the critic on how the agent is doing and determines how the performance element should be modified to do better in the future.

Learning Agent

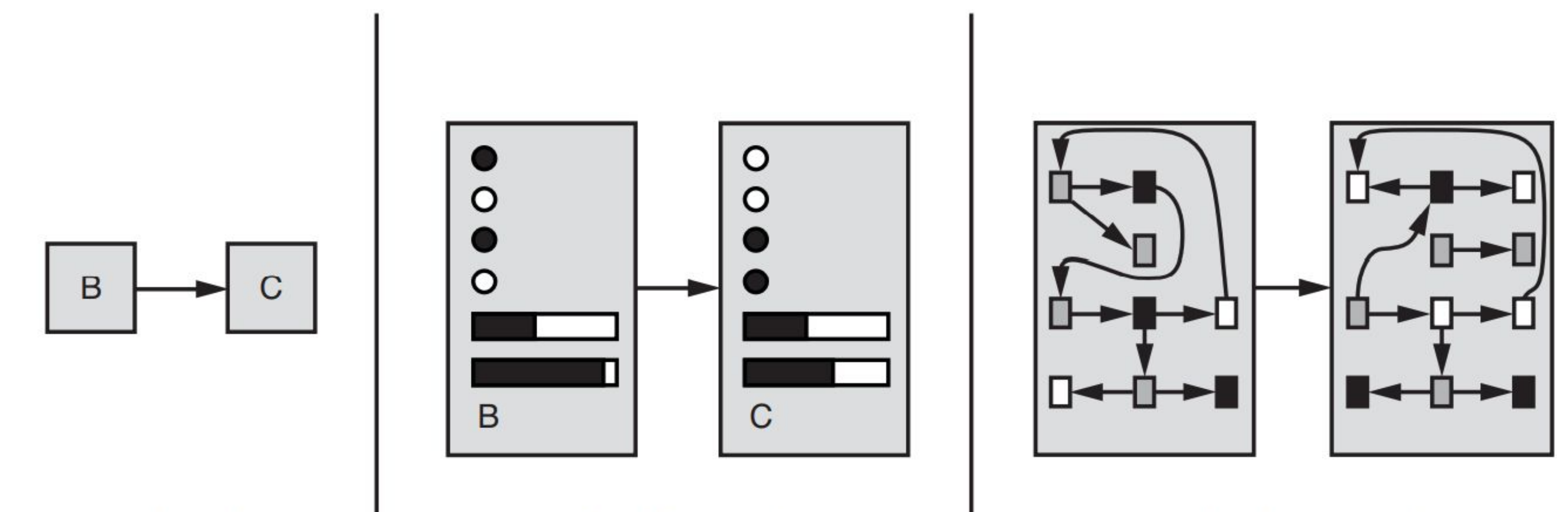
- Problem Generator : It is responsible for suggesting actions that will lead to new and informative experiences.
- The problem generator's job is to suggest exploratory actions.
- Learning in intelligent agents can be summarized as a process of modification of each component of the agent to bring the components into closer agreement with the available feedback information, thereby improving the overall performance of the agent.

Learning Agent



Components of Agent Program Representation

- The Complexity of Details in the Agent Program can be represented using atomic, factored, and structured.



(a) Atomic (b) Factored (b) Structured

Components of Agent Program Representation

- Atomic representation: each state of the world is indivisible—it has no internal structure. Two different atomic states have nothing in common they are just different black boxes
- A factored representation splits up each state into a fixed set of variables or attributes, each of which can have a value. Two different factored states can share some attributes.
- Structured Representation: Relationships between the objects of a state can be explicitly expressed. AI algorithms: first-order logic, knowledge-based learning, natural language understanding.

Summary

An agent is something that perceives and acts in an environment. The agent function for an agent specifies the action taken by the agent in response to any percept sequence. The performance measure evaluates the behavior of the agent in an environment. A rational agent acts to maximize the expected value of the performance measure.

A task environment specification includes the performance measure, the external environment, the actuators, and the sensors. In designing an agent, the first step must always be to specify the task environment as fully as possible. They can be fully or partially observable, single-agent or multiagent, deterministic or stochastic, episodic or sequential, static or dynamic, discrete or continuous, and known or unknown.

Simple reflex agents respond directly to percepts, whereas model-based reflex agents maintain internal state to track aspects of the world that are not evident in the current percept. Goal-based agents act to achieve their goals, and utility-based agents try to maximize their own expected “happiness.”

All agents can improve their performance through learning.