Agent and Environment

Outline

- Agents and Environments
- Rationality
- PEAS (Performance Measure, Environment, Actuators, Sensors)
- Environment Types
- Agent types

Intelligent Agents

- The concept of rational agents as central to artificial intelligence.

- How well an agent can behave depends on **the nature of the environment;** some environments are more difficult than others.

- In general, an agent's choice of action at any given instant can depend on the **entire percept sequence observed to date (History)**, but not on anything it hasn't perceived.

- Mathematically an agent's behavior is described by the **agent function that maps from the percept sequence to an action**.

Agents

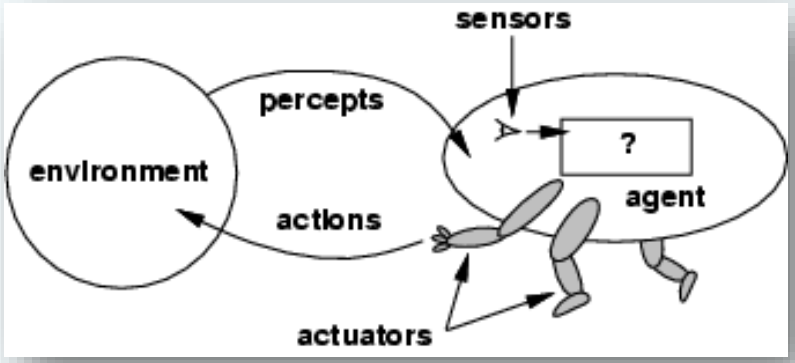
- An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators.
- Human agent:
  - Eyes, ears, and other organs for sensors
  - Hands, legs, mouth, and other body parts for actuators
- Robotic agent:
  - Cameras and infrared range finders for sensors
  - Various motors for actuators.

Agents and Environments

- The agent function maps from percept histories to actions:

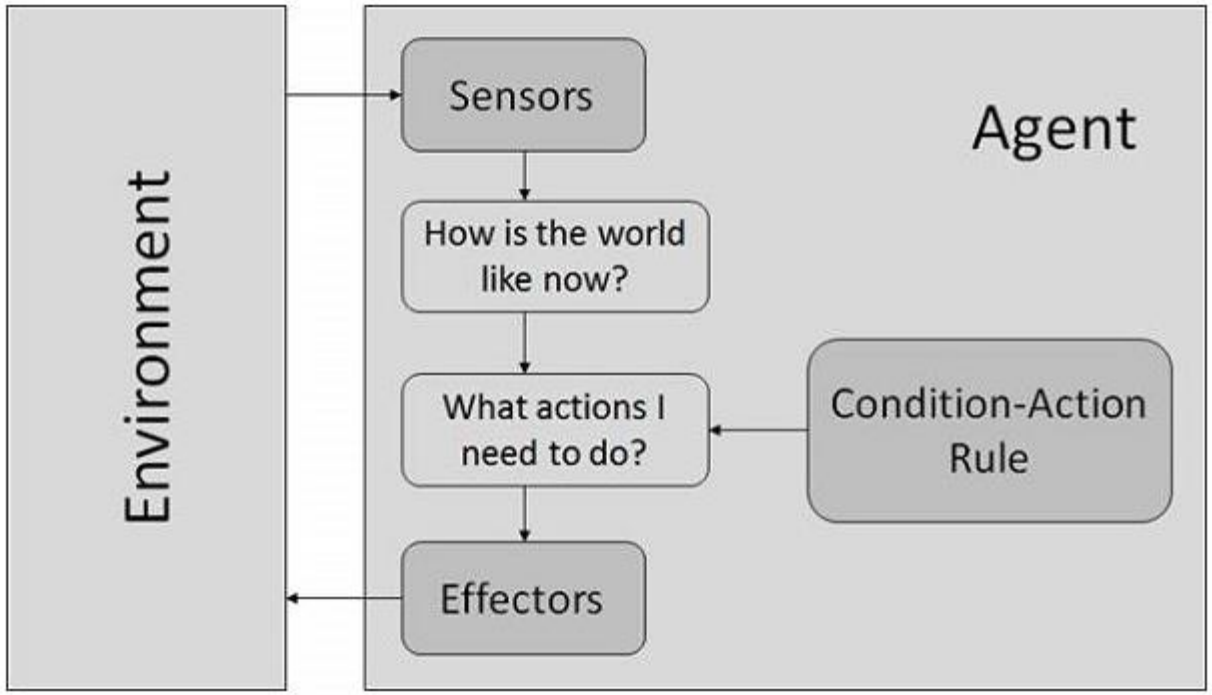
[*f*: P* → A]; The agent function is an abstract mathematical description;

- The agent program runs on the physical architecture to produce *f;* a concrete implementation
  - Agent = architecture + program



• The program we choose must be appropriate for the architecture

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

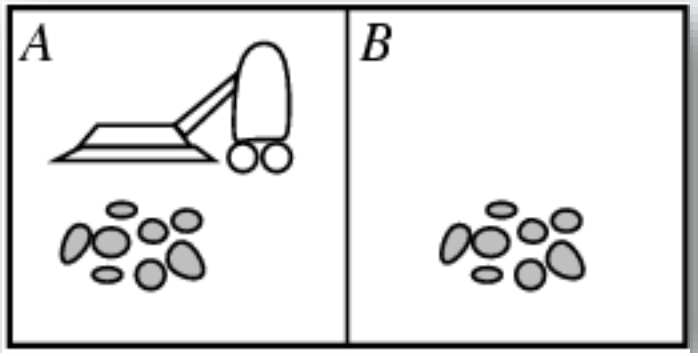


Construct a table by trying out all possible percept sequences and recording which actions the agent does in response.

The table is, an *external characterization* of the agent while *internally, the agent function by an* agent program.

Vacuum-cleaner World

- Percepts: *location* and *contents*, e.g., [A,Dirty]
- Actions: *Left*, *Right*, *Pick_Dirt*, *NoOp*



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
⋮	⋮

Rationality – Performance Measure

- Rationality is nothing but status of being reasonable, sensible, and having good sense of judgment.
- An agent should strive to "do the right thing", based on what it can perceive and the actions it can perform.
- The right action is the one that will cause the agent to be most successful.
- Performance measure: An objective criterion for success of an agent's behavior.
  - e.g., performance measure of a vacuum-cleaner agent could be amount of dirt cleaned up, amount of time taken, amount of electricity consumed, amount of noise generated, etc.

Rational Agent - Role

- Rational Agent: 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.
- Rationality maximizes *expected performance, while perfection maximizes actual performance.*
- 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 and its built-in knowledge base.

PEAS

- PEAS: Performance measure, Environment, Actuators, Sensors
- Must first specify the setting for intelligent agent design
- Consider, e.g., the task of designing an **automated taxi driver**:
  - Performance measure: Safe, fast, legal, comfortable trip, maximize profits
  - Environment: Roads, other traffic, pedestrians, customers
  - Actuators: Steering wheel, accelerator, brake, signal, horn
  - Sensors: Cameras, sonar, speedometer, GPS, odometer, engine sensors, keyboard.

PEAS

- Agent: Medical Diagnosis System
  - Performance measure: Healthy patient, minimize costs, lawsuits
  - Environment: Patient, hospital, staff
  - Actuators: Screen display (questions, tests, diagnoses, treatments, referrals)
  - Sensors: Keyboard (entry of symptoms, findings, patient's answers).

PEAS

- Agent: Part-picking robot
  - Performance measure: Percentage of parts in correct bins
  - Environment: Conveyor belt with parts, bins
  - Actuators: Jointed arm and hand
  - Sensors: Camera, joint angle sensors.

PEAS

- Agent: Interactive English tutor
  - Performance measure: Maximize student's score on test
  - Environment: Set of students
  - Actuators: Screen display (exercises, suggestions, corrections)
  - Sensors: Keyboard.

Environment Types

- Fully Observable : An agent's sensors give it access to the complete state of the environment at each point in time.
- In FO, the agent can be confident that it requires nothing more in order to decide on the optimal action.
- Partially Observable: PO can be due to faulty sensors
  - If we are driving along a stretch of road that we know well, and if dust enters our eyes, we are still able to drive on the route with blinking eyes (partial observation).
  - PO requires the agent to have an internal representation of the state.

Environment Types

- Deterministic (vs. Stochastic): The next state of the environment is completely determined by the current state and the action executed by the agent.
  - Stochastic (Non-Deterministic): There can be more than one next state, for a given state-action combination
    - Taxi Driving is clearly stochastic
- Consider a Multi-agent environment
  - If the environment is deterministic except for the actions of other agents, then the environment is strategic.
  - Strategy Games.

Environment Types

- Episodic (vs. sequential): The agent's experience is divided into atomic "episodes"
  - Each episode consists of the agent perceiving and then performing a single action, and the choice of action in each episode depends only on the episode itself, e.g., a robot whose job is to detect faulty parts on a line in some factory
  - In a sequential setting, the next episode depends on the previous one(s), e.g., learning which chess move to execute at each sequential step, in order to win the game at the end
  - Also called a sequential decision process.

Environment Types

- Static (vs. Dynamic): The environment is unchanged while an agent is deliberating which action to execute
  - Much more simpler to deal with
  - For the dynamic case, the agent needs to keep track of the changes
  - The environment is semi-dynamic if the environment itself does not change with the passage of time but the agent's performance score does, e.g., checkers.

- Discrete (vs. Continuous): The environment is discrete if the number of actions and possible states of the environment is finite otherwise it is continuous.

Environment Types

- Single Agent (vs. Multi-Agent): An agent operating by itself in an environment
  - In the multi-agent case, the performance measure of one agent depends on the performance measures of the other agent(s)
  - Competitive multi-agent: Chess Playing
  - Collaborative multi-agent: Robo Soccer.
  - A quite complicated field which is currently the focus of much research.

Environment Types

	Chess with a clock	Chess without a clock	Taxi driving
Fully observable	Yes	Yes	No
Deterministic	Strategic	Strategic	No
Episodic	No	No	No
Static	Semi	Yes	No
Discrete	Yes	Yes	No
Single agent	No	No	No

- The environment type largely determines the agent design
- The real world is partially observable, stochastic, sequential, dynamic, continuous, multi-agent.

Agent Functions and Programs

- An agent is completely specified by the agent function that maps percept sequences to actions.
  - Can also be labeled as the strategy of the agent
- There could be many possible agent functions.
- Aim: Discover the most rational (optimal) agent function.

Table-lookup Agent

- Simplest possible agent function:
  - All possible states and their optimal actions specified by the designers in advance
- Drawbacks:
  - Huge table (consider continuous states)
  - Could take a long time to build the table
  - No autonomy!
  - Even with learning, agent could need a long time to learn the table entries.

Let P be the set of possible percepts and let T be the lifetime of the agent (the total number of percepts it will receive).

The lookup table will contain $\sum_{t=1..T} |P|^t$ entries.

- Consider the automated taxi: the visual input from a single camera comes in at the rate of roughly 27 megabytes per second (30 frames per second, 640×480 pixels with 24 bits of color information).

- This gives a lookup table with over $10^{250,000,000,000}$ entries for an hour's driving.

No physical agent in this universe will have the space to store the table.

However, it works when decision is based only on the current location .

Agent types

- Four basic types in order of increasing generality:
  - Simple Reflex agents
  - Model-based Reflex agents
  - Goal-based agents
  - Utility-based agents

- And Finally: Learning agents

Agent programs

- **Simple reflex agents** ignore the percept history and act only on the basis of the **current percept**.

- The agent function is based on the **condition-action rule**.

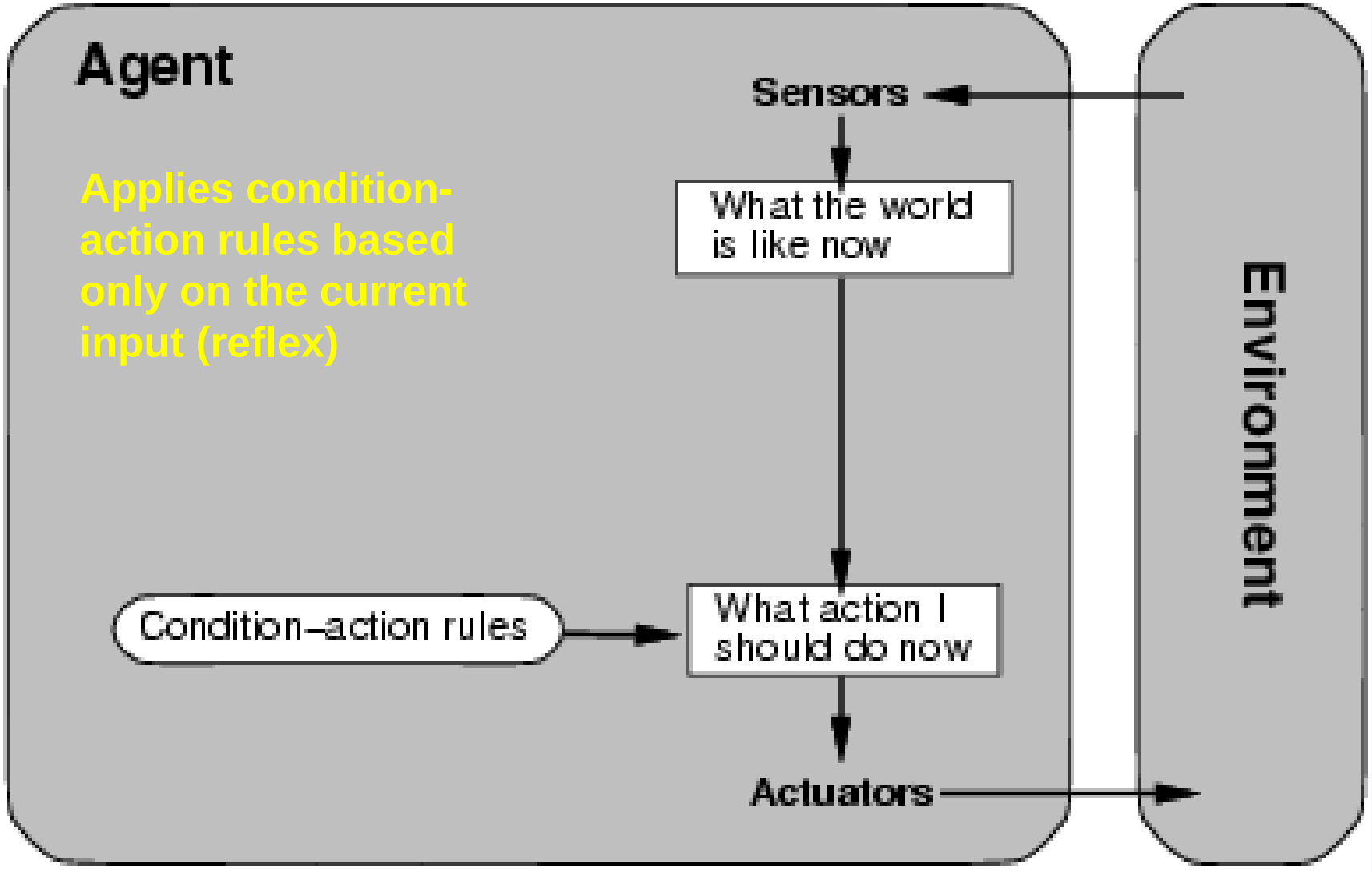
- A condition-action rule is a rule that maps a state i.e., a condition to an action and if the condition is true, then the action is taken, else not.

- This agent function only succeeds when the environment is fully observable.

- For simple reflex agents operating in partially observable environments, infinite loops are often unavoidable.

- It may be possible to escape from infinite loops if the agent can randomize its actions.

Simple Reflex Agents



The vacuum agent is a simple reflex agent whose agent function is to take decision based only on the current location and on whether that location contains dirt.

The agent program for a simple reflex agent in the two-state vacuum environment.

```
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
```

• The most obvious reduction comes from ignoring the percept history, which cuts down the number of possibilities from 4^T to just 4.

• A further, small reduction comes from the fact that when the current square is dirty, the action does not depend on the location.

• Simple reflex behaviors occur even in more complex environments.
condition–action rule, written as

if car-in-front-is-braking then initiate-braking

Simple Reflex agents

- Automated Taxi:
  - Agent observes rain falling on the windshield: Agent powers on the viper.
  - Agent observes a red signal; Agent brakes the taxi until it stops.

Problems with Simple reflex agents :

(i) Very limited intelligence.
(ii) No knowledge of non-perceptual parts of the state.
(iii) If any change occurs in the environment, then the rules needs to be updated.

work *only if the correct decision can be made on the basis of the current percept and if the environment is fully observable*.

Model-based reflex agents

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

- The agent has to keep track of the **internal state,** a representation of unobserved aspects of current state depending on percept history.

- Internal state is adjusted by each percept and that depends on the percept history.

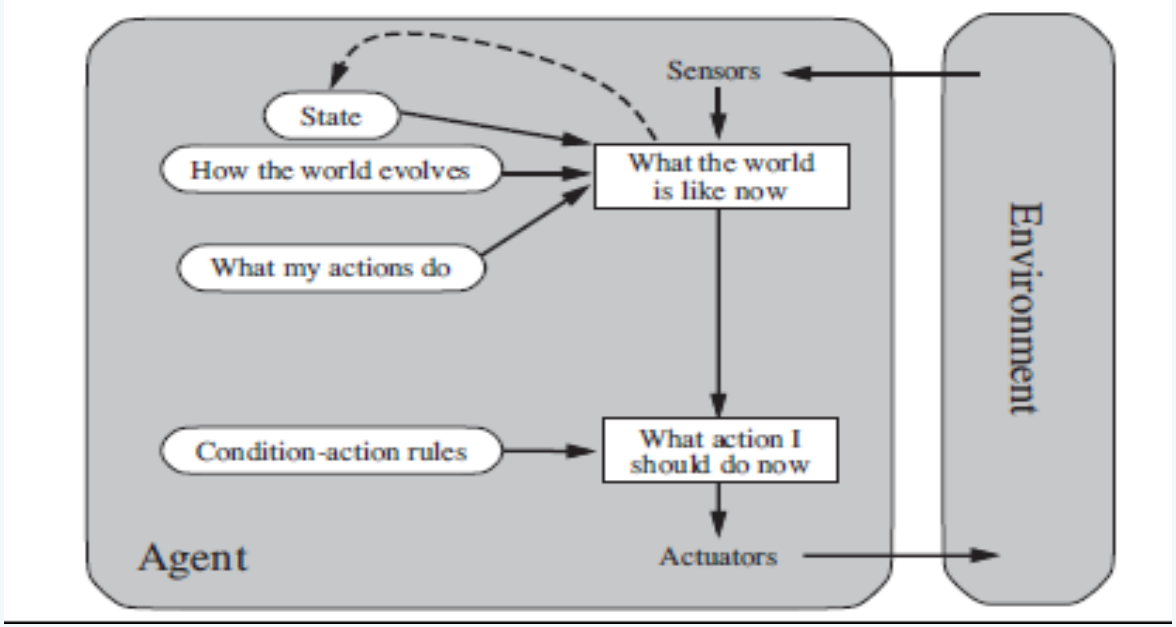
Updating the state requires the information about −

How the world evolves.

How the agent's actions affect the world.

For the braking problem, the internal state is the previous frame from the camera, allowing the agent to detect when two red lights at the edge of the vehicle go on or off simultaneously.

Model-based Reflex agents



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

It is seldom possible for the agent to determine the current state of a partially observable environment *exactly.* Uncertainty about the current state may be unavoidable, but the agent still has to make a decision.

"driving back home" may *seem to an aspect of the world state, the fact of the taxi's destination is actually an aspect of* the agent's *internal state*.

Model-based Reflex agents

- Robo-Soccer Example:
  - Imagine a robotic goalkeeper
  - It can build a model of the dynamics of the game that is played on the field, e.g., when the ball is kicked in its direction, the ball will be nearer to it in the next time step
  - If this robot is not able to acquire its state at some time step, then using the model, it knows that the ball has come nearer
  - It also know what consequences a dive will have
  - So, it can time its dive early and hence, save the goal.

Goal Based Agents

- These kinds of agents take decisions based on how far they are currently from their **goal** (description of desirable situations).

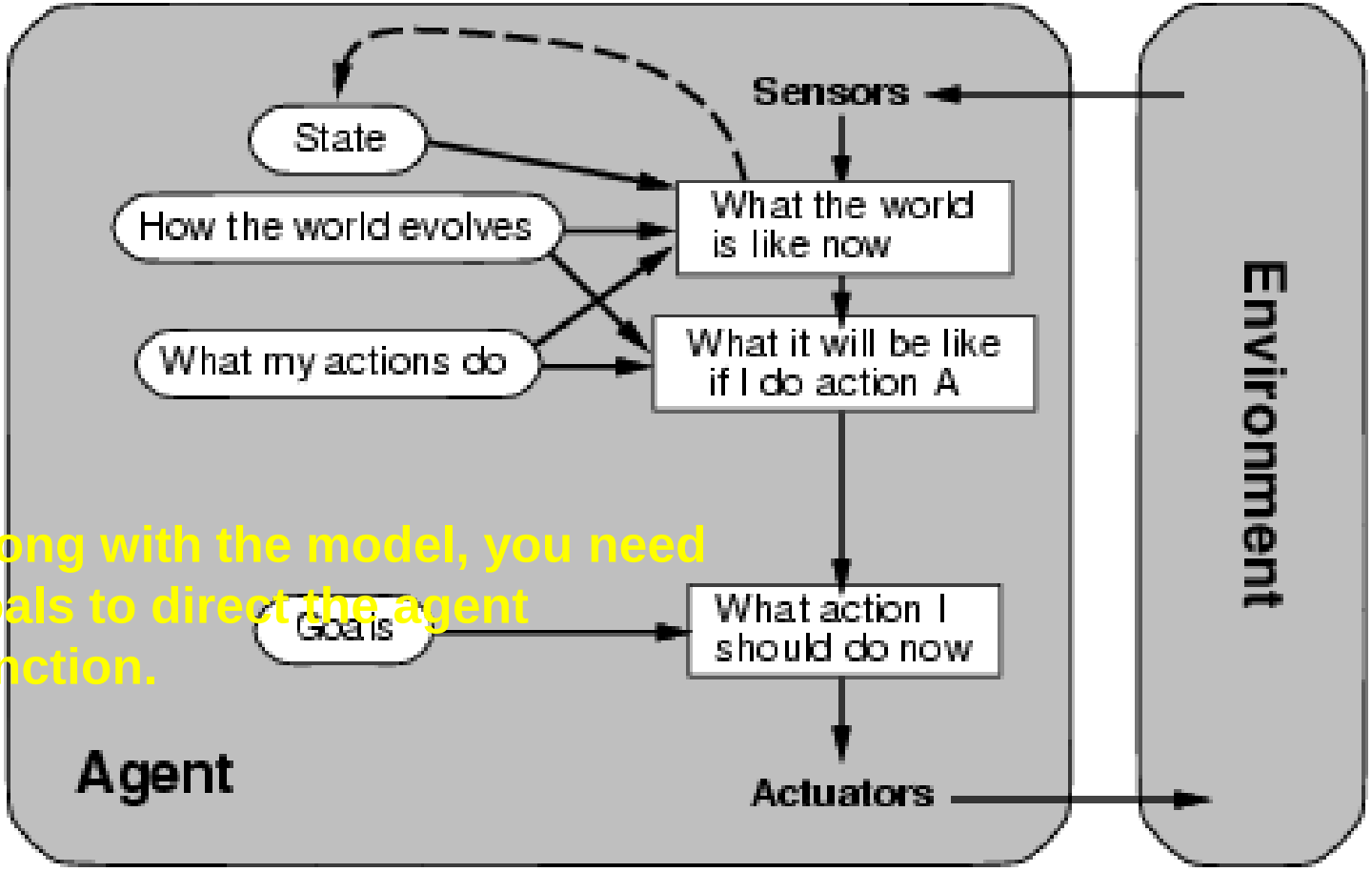
- Their every action is intended to reduce their distance from the goal, which allows the agent a way to choose among multiple possibilities, selecting the one which reaches a goal state.

- The agent program can combine this with the model to choose actions that achieve the goal.

- The knowledge that supports its decisions is represented explicitly and can be modified, which makes these agents more flexible.

- **Search** and **planning** are the subfields of AI devoted to finding action sequences that achieve the agent's goals.

Goal-based agents



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.

• Decision making of this kind is fundamentally different from the condition–action rules .

• It involves consideration of the future—both “What will happen if I do such-and-such?” and “Will that make me happy?”

• In the reflex agent designs, this information is not explicitly represented because the built-in rules map directly from percepts to actions.

• The reflex agent brakes when it sees brake lights while a goal-based agent, in principle, could reason that if the car in front has its brake lights on, *it will slow down.*

• Given the way the world usually evolves, the only action that will achieve *the goal of not hitting other cars is to brake*.

• 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. For the reflex agent, on the other hand, we would have to rewrite many condition–action rules.

Goal-based Agents

- Automated Taxi:
  - Consider the agent at a crossing, where it can turn right, left, or go straight
  - Using the model, the Agent can understand the consequences of turning left, right or going straight ahead.
  - All 3 might seem the optimal actions to take.
  - However, the agent needs to select one of these actions in order to reach the destination of the passenger.
- 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.

Utility Based Agents

- Goals alone are not enough to generate high-quality behavior in most environments.
- When there are multiple possible alternatives, then to decide which one is best, utility-based agents are used.
- They choose actions based on a **preference (utility)** for each state.
- Agent happiness should be taken into consideration.
- Utility describes how **"happy"** the agent is.
- Because of the uncertainty in the world, a utility agent chooses the action that maximizes the expected utility.
- A utility function maps a state onto a real number which describes the associated degree of happiness.

- An agent's **utility function** is essentially an internalization of the performance measure.

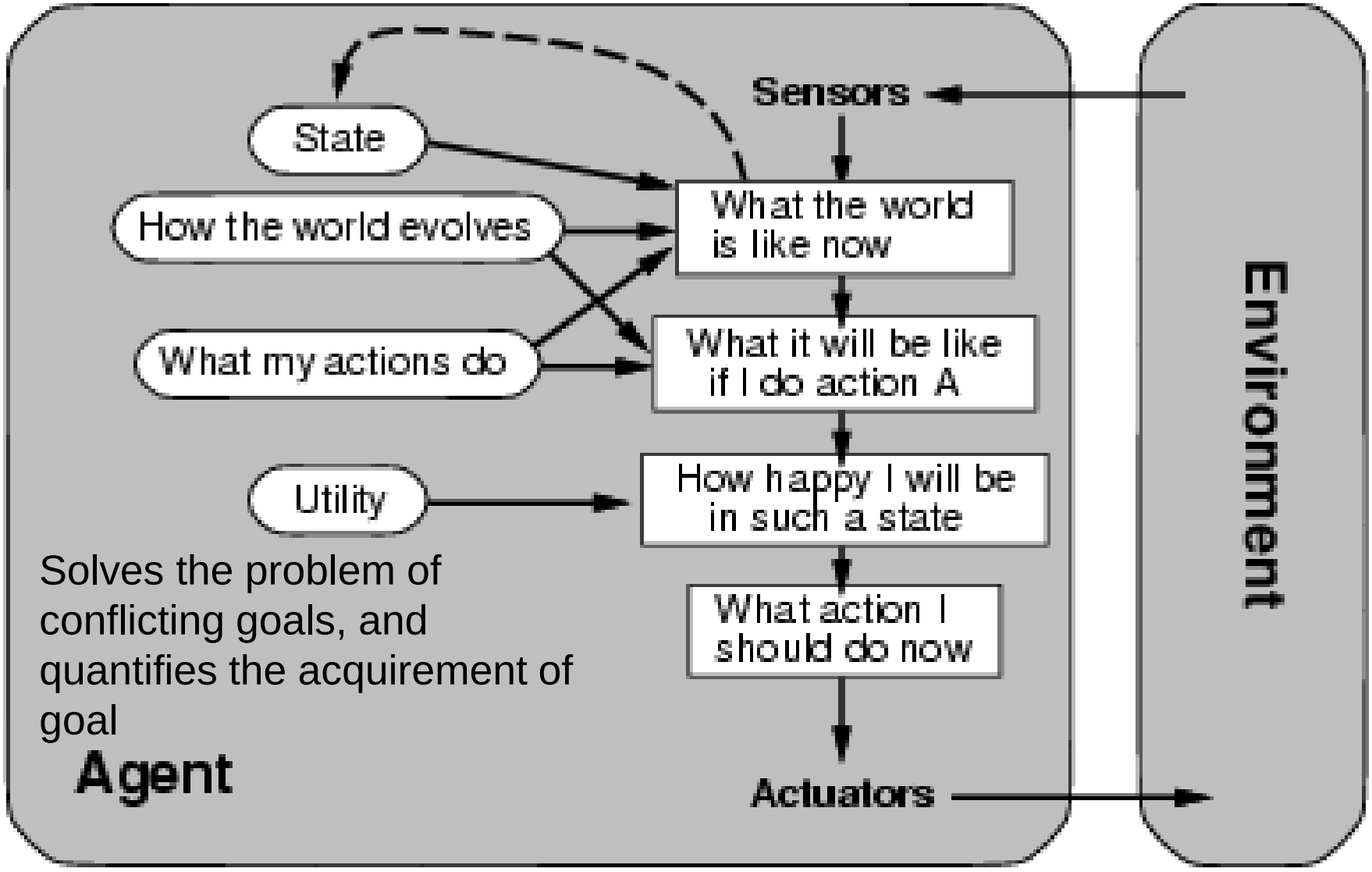
- If the internal utility function and the external performance measure are in agreement, then an agent that chooses actions to maximize its utility will be rational.

- When there are conflicting goals, only some of which can be achieved (for example, speed and safety), the utility function specifies the appropriate tradeoff.

- When there are several goals that the agent can aim for, none of which can be achieved with certainty, utility provides a way in which the likelihood of success can be weighed against the importance of the goals.

- Technically speaking, a rational utility-based agent chooses the action that maximizes the expected utility of the action outcomes—i.e., th**e** utility the agent expects to derive, on average, given the probabilities and utilities of each outcome.

Utility-based agents



Solves the problem of conflicting goals, and quantifies the acquirement of goal

Utility-based Agent

- Automated Taxi:
  - Consider the agent at a crossing, where it can turn right, left, or go straight
  - The agent will calculate the utility of each such action
  - It will select the action which maximizes the utility function, i.e., in most cases, the expected profit that the agent can expect to receive in the long run (when the passenger reaches the destination)
  - E.g., going straight could have highest utility.

Learning Agent

A learning agent in AI is the type of agent that can learn from its past experiences or it has learning capabilities.

It starts to act with basic knowledge and then is able to act and adapt automatically through learning.

A learning agent has mainly four conceptual components, which are:

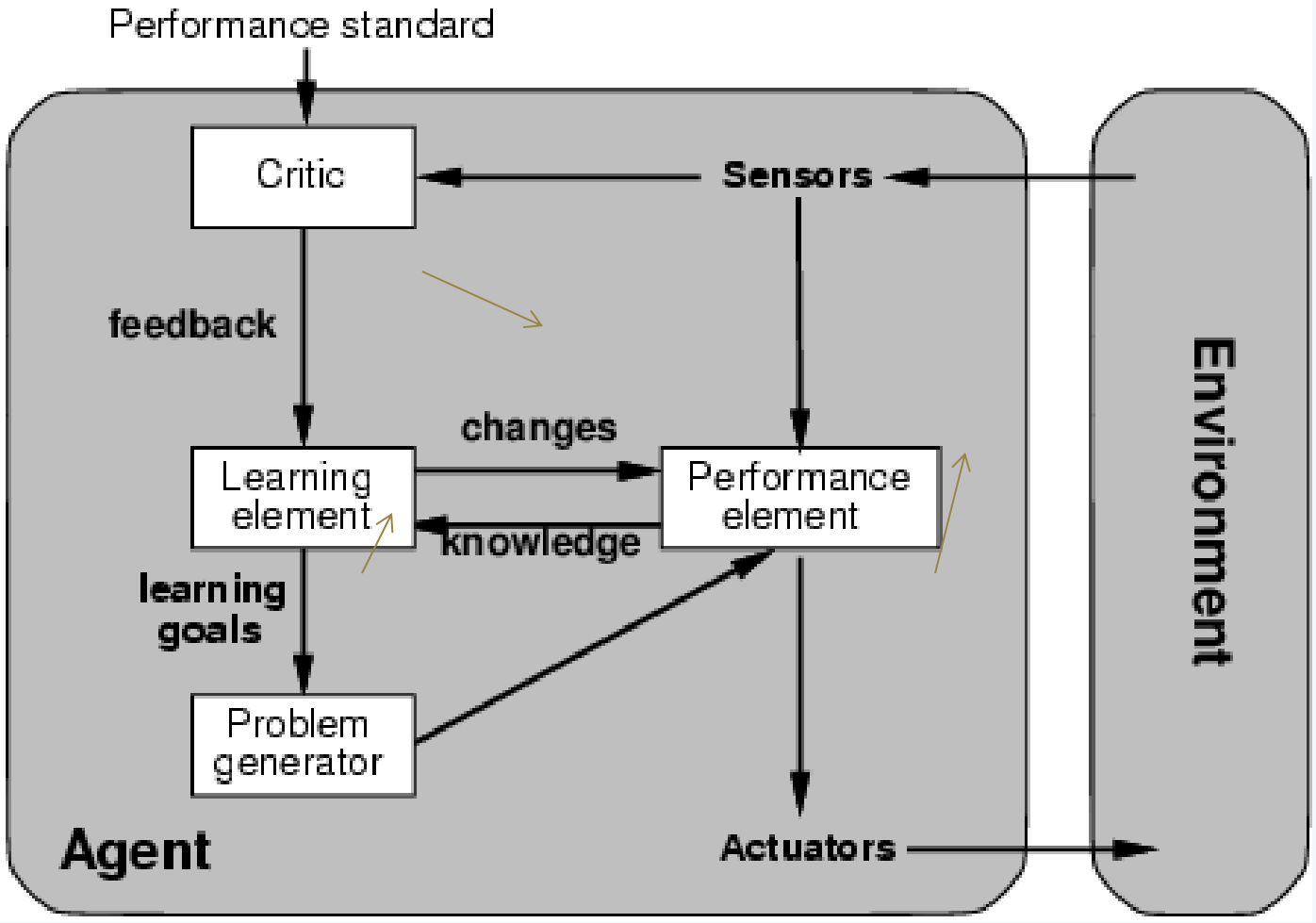
Learning element: It is responsible for making improvements by learning from the environment.

Critic: The learning element takes feedback from critics which describes how well the agent is doing with respect to a fixed performance standard.

Performance element: It is responsible for selecting external action.

Problem Generator: This component is responsible for suggesting actions that will lead to new and informative experiences.

Learning Agents



problem generator is responsible for suggesting actions that will lead to new and informative experiences.

The agent is willing to explore a little and the generator's job is to suggest these exploratory actions.

Multi-Agent Systems

A multi-agent system (MAS) is a system composed of multiple interacting agents that are designed to work together to achieve a common goal.

These agents may be autonomous or semi-autonomous and are capable of perceiving their environment, making decisions, and taking action to achieve the common objective.

In a homogeneous MAS, all the agents have the same capabilities, goals, and behaviors.

In contrast, in a heterogeneous MAS, the agents have different capabilities, goals, and behaviors.

MAS can be implemented using different techniques, such as game theory, machine learning, and agent-based modeling.

Game theory is used to analyze strategic interactions between agents and predict their behavior.

Machine learning is used to train agents to improve their decision-making capabilities over time.

Agent-based modeling is used to simulate complex systems and study the interactions between agents.

Hierarchical Agents

- These agents are organized into a hierarchy, with high-level agents overseeing the behavior of lower-level agents.
- The high-level agents provide goals and constraints, while the low-level agents carry out specific tasks to achieve the goals set by the high-level agent.
- Hierarchical agents are useful in complex environments with many tasks and sub-tasks.
- These tasks may be relatively simple or more complex, depending on the specific application.
- For example, in a transportation system, low-level agents might be responsible for managing traffic flow at specific intersections.

Artificial Environment

- A model of the environment where the simulation model is operating and the model is completely controllable by the modeler.

- Particular environment models are highly relevant when modeling adaptive elements of the system or when using adaptive capabilities of the agents contained within the model.

- The most famous **artificial environment** is the **Turing Test environment**, in which one real and other artificial agents are tested on equal ground.

- The success of an intelligent system can be measured with **Turing Test**.

- 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.