Artificial Intelligence

Intelligent Agents

Lecture -2

Agents and Environments

- An **agent** is a software or system that is designed to act autonomously and perform specific tasks
- It perceives its environment through *sensors* and **acts upon that environment** through *actuators*
- Human: sensors are eyes, ears; actuators (effectors) are hands, legs, mouth

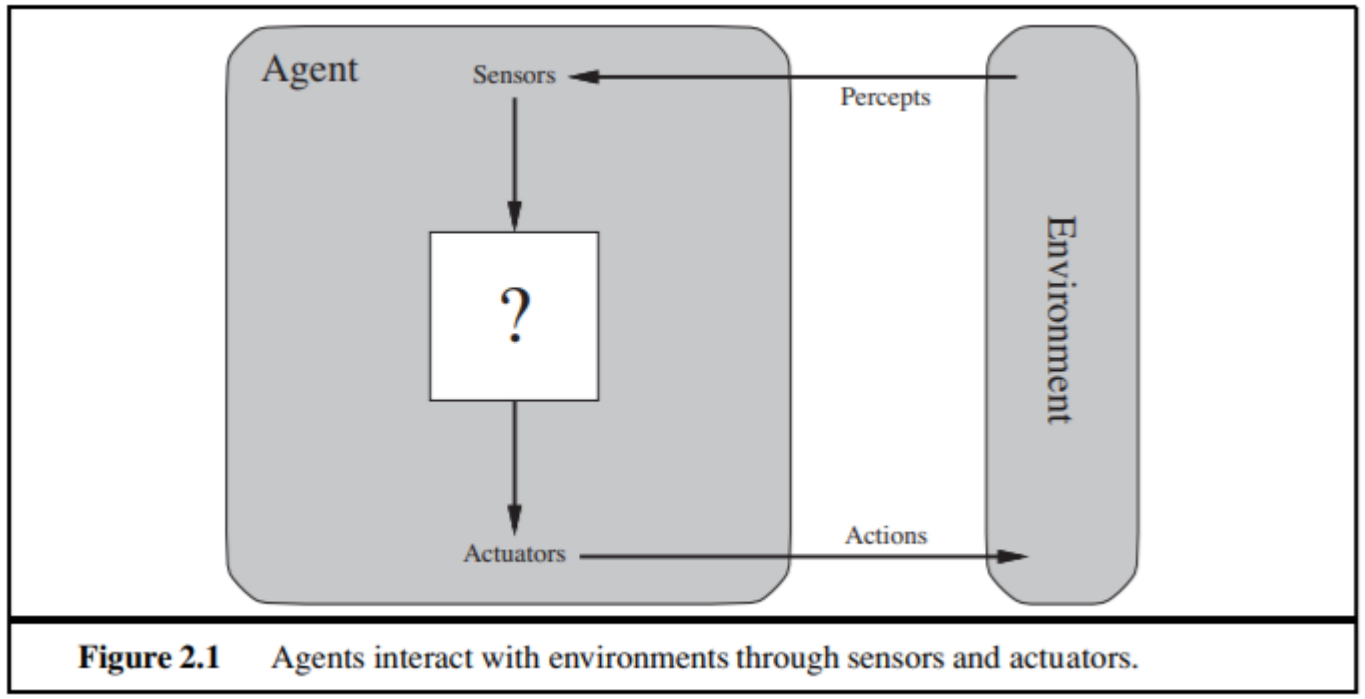


Figure 2.1 Agents interact with environments through sensors and actuators.

Agents and Environments

- Software Agent
  - **sensory input**: keystrokes, file contents, and network packets
  - **Actuator**: acts on the environment by displaying on the screen, writing files, and sending network packets

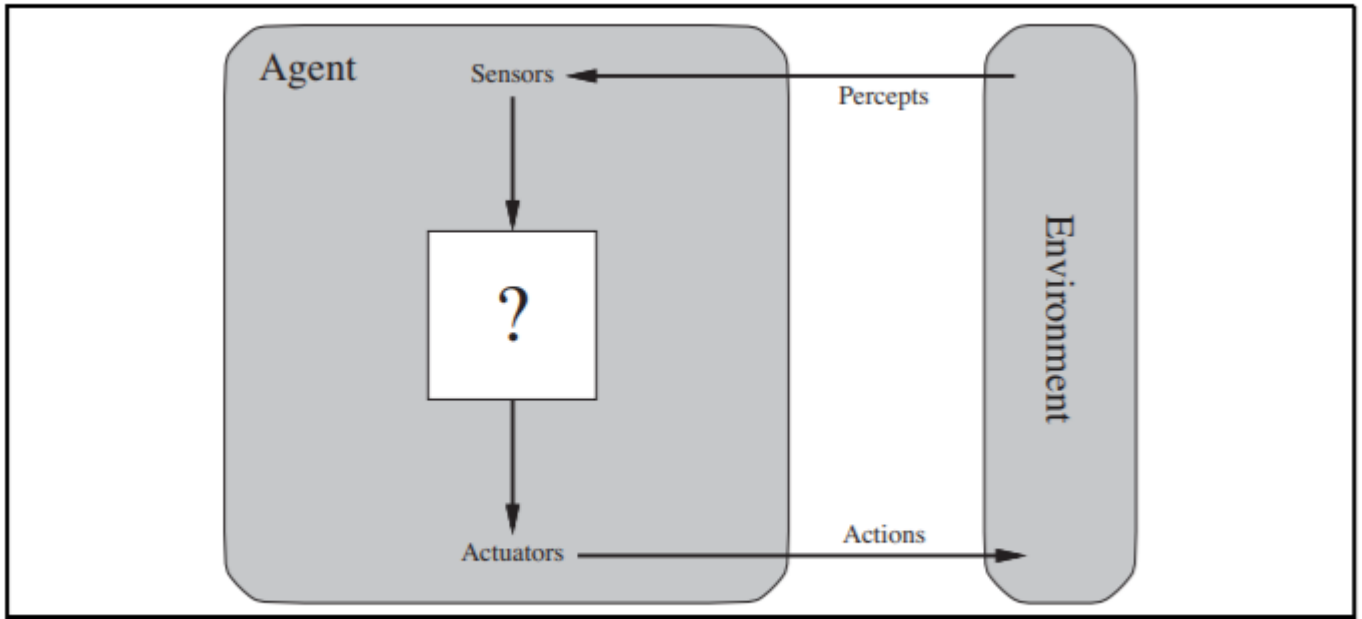


Figure 2.1 Agents interact with environments through sensors and actuators.

Rational Agent

- A rational agent is one that does the right thing
- Rationality:
  - The performance measure that defines criterion of success
  - The agent prior knowledge of the environment
  - The actions that agent can perform

DEFINITION OF A RATIONAL AGENT

This leads to a **definition of a 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.

PEAS

- Performance measure
- Environment
- Actuators
- 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.

Properties of Task Environments

- Fully observable vs. partially observable **or Unobservable**
- Deterministic vs. stochastic / strategic
  - 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
  - Stochastic systems, involve a certain degree of randomness or probability.
  - Strategic systems refers to decision making process where the decision maker take into account the potential actions of other decision makers and their potential reactions to the decision maker's own actions.

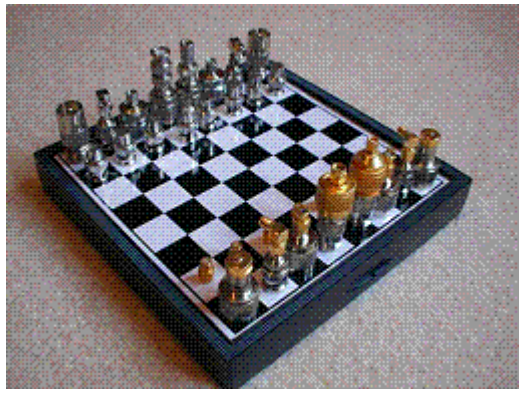
Properties of Task Environments

- Taxi driving: stochastic (behavior of traffic cannot be determined)
- Vacuum cleaning: deterministic
  - Variations may include stochastic elements such as randomly appearing dirt and an unreliable suction
- Single agent vs. multiagent (Solving crossword puzzle vs chess playing)
  - For Multi-Agents: Competitive (in Chess)
  - cooperative multi-agent env (in driving to avoid collision)
  - Partially competitive: only one car can occupy a parking space

Properties of Task Environments

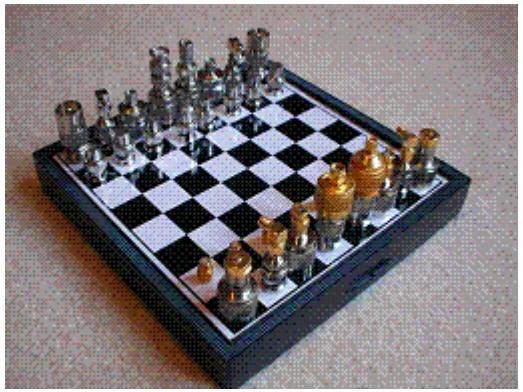
- Episodic vs. sequential
  - In each episode, the agent receives a percept and then performs a single action that doesn't affect the action in previous episodes.
  - In sequential environments, the current decision could affect all future decisions
- Static vs. dynamic
  - If the environment can change while an agent is deliberating, then we say the environment is dynamic for that agent; otherwise, it is static
  - **semi dynamic:** 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.
- Discrete vs. continuous
  - Discrete has finite countable states
  - Transitions between states occur in discrete steps, and the agent's actions lead to distinct state changes, in continuous env transitions occur smoothly and gradually in a continuum
  - Taxi-driving is continuous-state, chess is discrete
- Known vs Unknown -- Focused on Agent's state of knowledge and not on env

Environment Properties



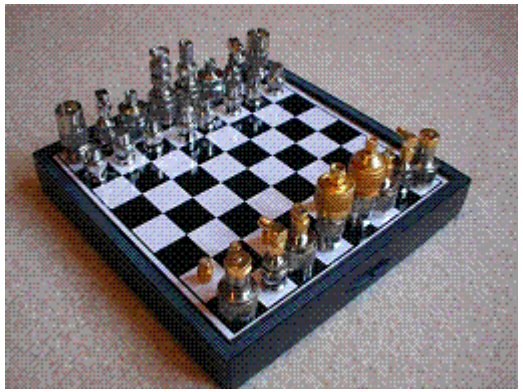
Environment	Observable	Deterministic	Episodic	Static	Discrete	Agents
Chess with a clock						
Chess without a clock						

Environment Properties



Environment	Observable	Deterministic	Episodic	Static	Discrete	Agents
Chess with a clock	Fully	Deterministic Strategic	Sequential	Semi	Discrete	Multi
Chess without a clock						

Environment Properties



Environment	Observable	Deterministic	Episodic	Static	Discrete	Agents
Chess with a clock	Fully	Deterministic Strategic	Sequential	Semi	Discrete	Multi
Chess without a clock	Fully	Deterministic Strategic	Sequential	Static	Discrete	Multi

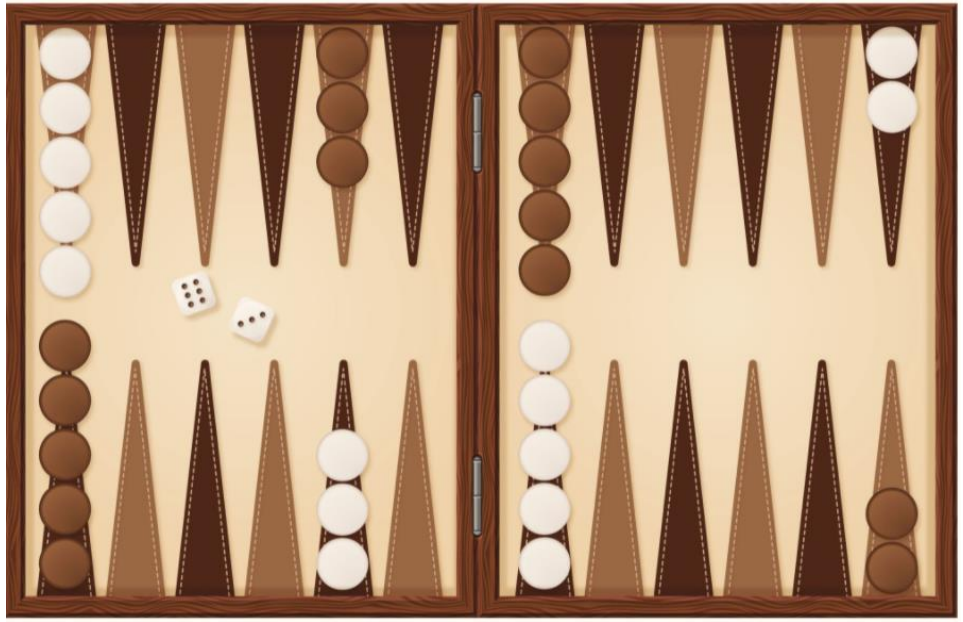
Environment Properties

Environment	Observable	Deterministic	Episodic	Static	Discrete	Agents
Chess with a clock	Fully	Strategic	Sequential	Semi	Discrete	Multi
Chess without a clock	Fully	Strategic	Sequential	Static	Discrete	Multi
Poker						

Environment Properties

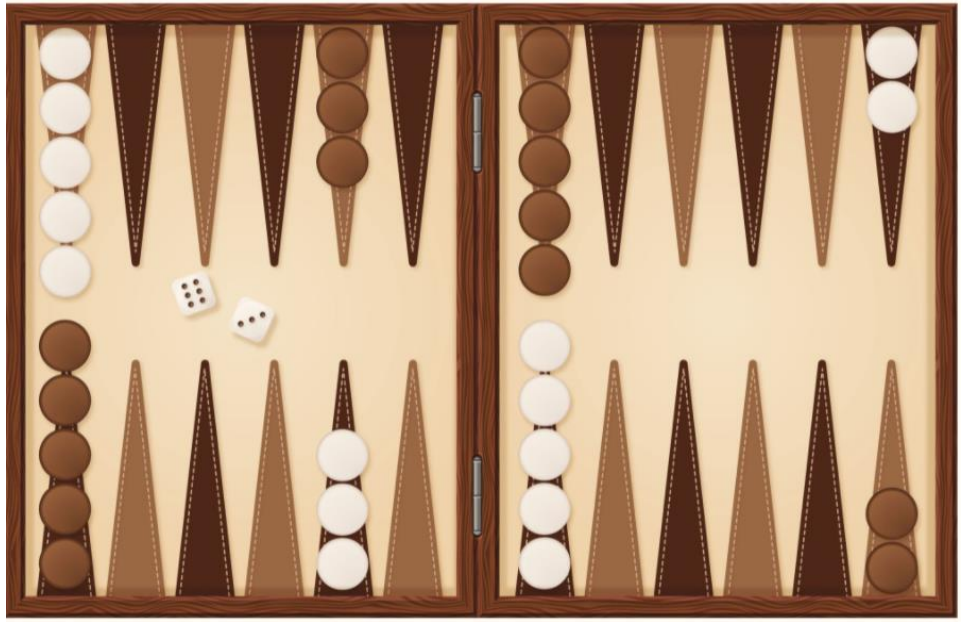
Environment	Observable	Deterministic	Episodic	Static	Discrete	Agents
Chess with a clock	Fully	Strategic	Sequential	Semi	Discrete	Multi
Chess without a clock	Fully	Strategic	Sequential	Static	Discrete	Multi
Poker	Partial	Strategic	Sequential	Static	Discrete	Multi

Environment Properties



Environment	Observable	Deterministic	Episodic	Static	Discrete	Agents
Chess with a clock	Fully	Strategic	Sequential	Semi	Discrete	Multi
Chess without a clock	Fully	Strategic	Sequential	Static	Discrete	Multi
Poker	Partial	Strategic	Sequential	Static	Discrete	Multi
Backgammon						

Environment Properties



Environment	Observable	Deterministic	Episodic	Static	Discrete	Agents
Chess with a clock	Fully	Strategic	Sequential	Semi	Discrete	Multi
Chess without a clock	Fully	Strategic	Sequential	Static	Discrete	Multi
Poker	Partial	Strategic	Sequential	Static	Discrete	Multi
Backgammon	Fully	Stochastic	Sequential	Static	Discrete	Multi

Environment Properties

Environment	Observable	Deterministic	Episodic	Static	Discrete	Agents
Chess with a clock	Fully	Strategic	Sequential	Semi	Discrete	Multi
Chess without a clock	Fully	Strategic	Sequential	Static	Discrete	Multi
Poker	Partial	Strategic	Sequential	Static	Discrete	Multi
Backgammon	Fully	Stochastic	Sequential	Static	Discrete	Multi
Taxi driving						
Medical diagnosis						

Environment Properties

Environment	Observable	Deterministic	Episodic	Static	Discrete	Agents
Chess with a clock	Fully	Strategic	Sequential	Semi	Discrete	Multi
Chess without a clock	Fully	Strategic	Sequential	Static	Discrete	Multi
Poker	Partial	Strategic	Sequential	Static	Discrete	Multi
Backgammon	Fully	Stochastic	Sequential	Static	Discrete	Multi
Taxi driving	Partial	Stochastic	Sequential	Dynamic	Continuous	Multi
Medical diagnosis	Partial	Stochastic	Episodic	Static	Discrete	Single

Structure of Agents

- Kinds of agents
  - Simple reflex agents
  - Model based reflex agents
  - Goal-based agents
  - Utility-based agents
  - Learning agents

Simple Reflex Agent

- Use simple “if then” rules also called condition-action rule
- Can be short sighted

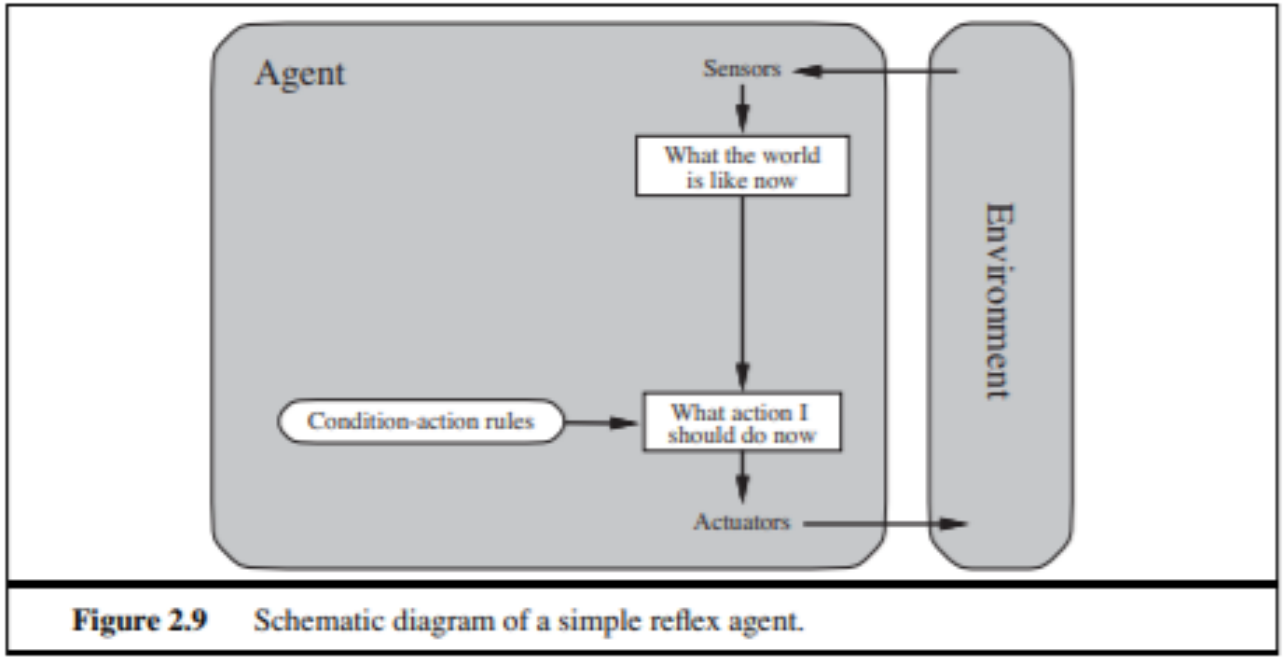


Figure 2.9 Schematic diagram of a simple reflex agent.

Model based reflex Agent

- Store previously-observed information
- Can reason about unobserved aspects of current state

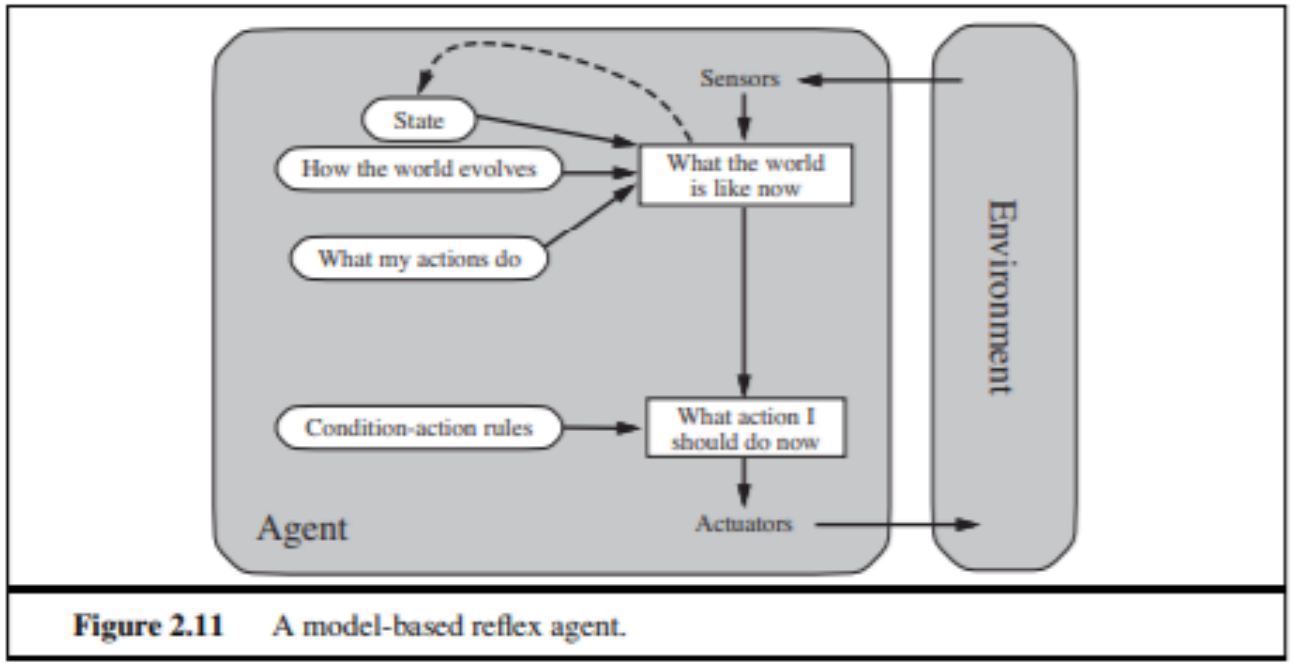


Figure 2.11 A model-based reflex agent.

Goal-based Agent

- Goal reflects desires of agents
- May project actions to see if consistent with goals
- Takes time, world may change during reasoning
- They have specific goals or objectives that they try to achieve, and they take actions based on the current percepts and their internal state to reach those goals

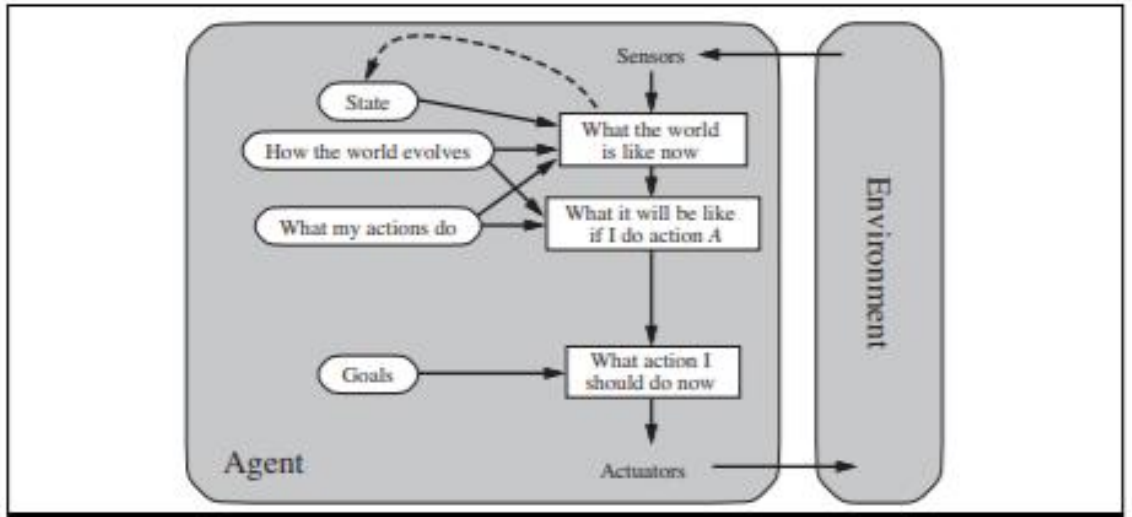


Figure 2.13 A model-based, goal-based agent. It keeps track of the world state as well as a set of goals it is trying to achieve, and chooses an action that will (eventually) lead to the achievement of its goals.

Utility-based Agent

- Goals alone are not enough to generate high-quality behavior in most environments
- Utility is a measure of the value or desirability of a particular state or outcome. The agent uses utility to determine the actions it should take in order to achieve its goals.
- They take into account the long-term consequences of their actions to maximize a specific utility function.

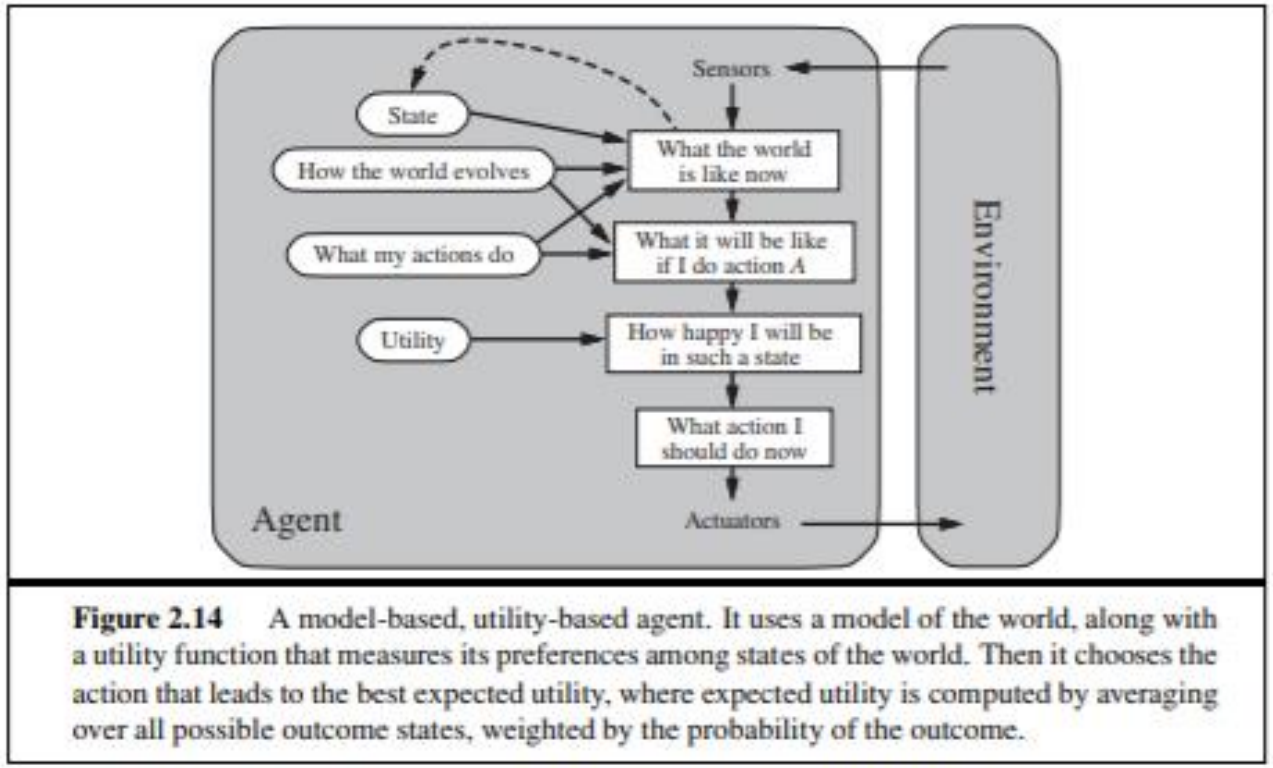


Figure 2.14 A model-based, utility-based agent. It uses a model of the world, along with a utility function that measures its preferences among states of the world. Then it chooses the action that leads to the best expected utility, where expected utility is computed by averaging over all possible outcome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

Learning Agent

- They are able to improve their performance over time by learning from their experiences and adjusting their behavior accordingly
- The agent uses past experiences and feedback to continuously improve its decision-making and problem-solving abilities.

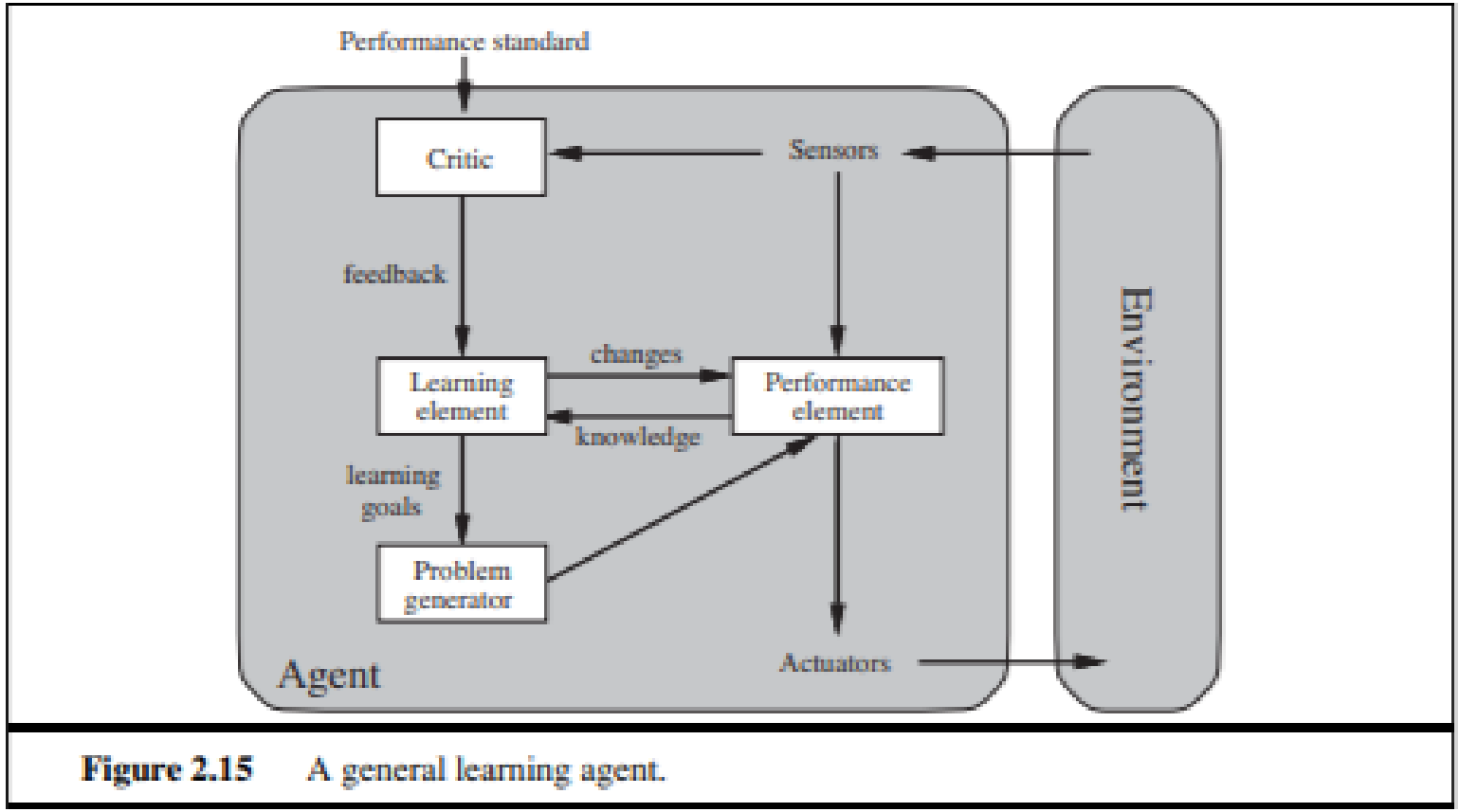


Figure 2.15 A general learning agent.

Homework

- Readings
  - CH 2- Intelligent Agent (Section 2.1 - 2.4)