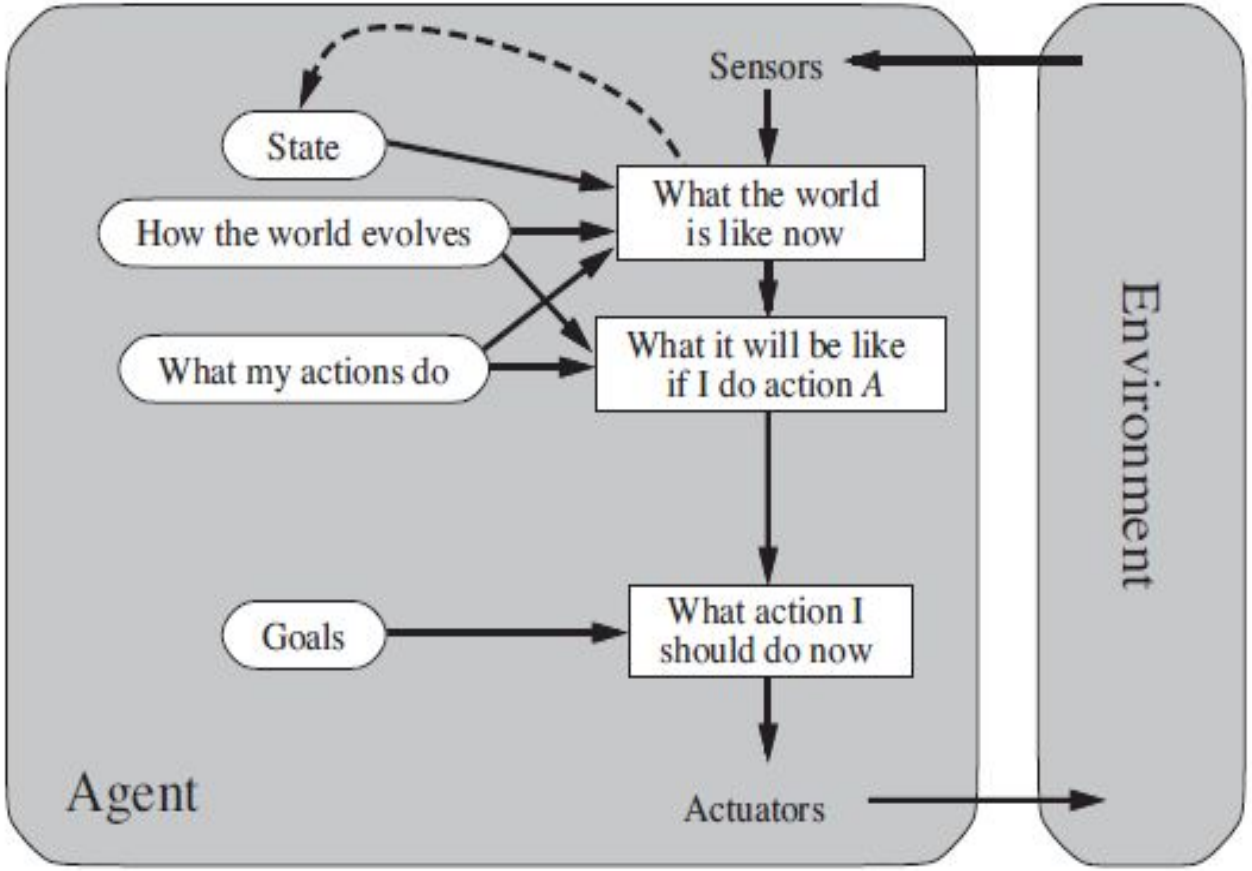
Module 1

Goal-based agents



A model-based, goal-based agent 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.

As a current state description, the agent needs some sort of **goal** information that describes situations that are desirable. 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.

Goal-based agents

- Along with current state description, the agent needs some sort of goal information that describes situations that are desirable.
- For example, at a road junction, the taxi can turn left, turn right, or go straight on. The correct decision depends on where the taxi is trying to get to.
- The agent program can combine this with the model (the same information as was used in the modelbased reflex agent) to choose actions that achieve the goal.

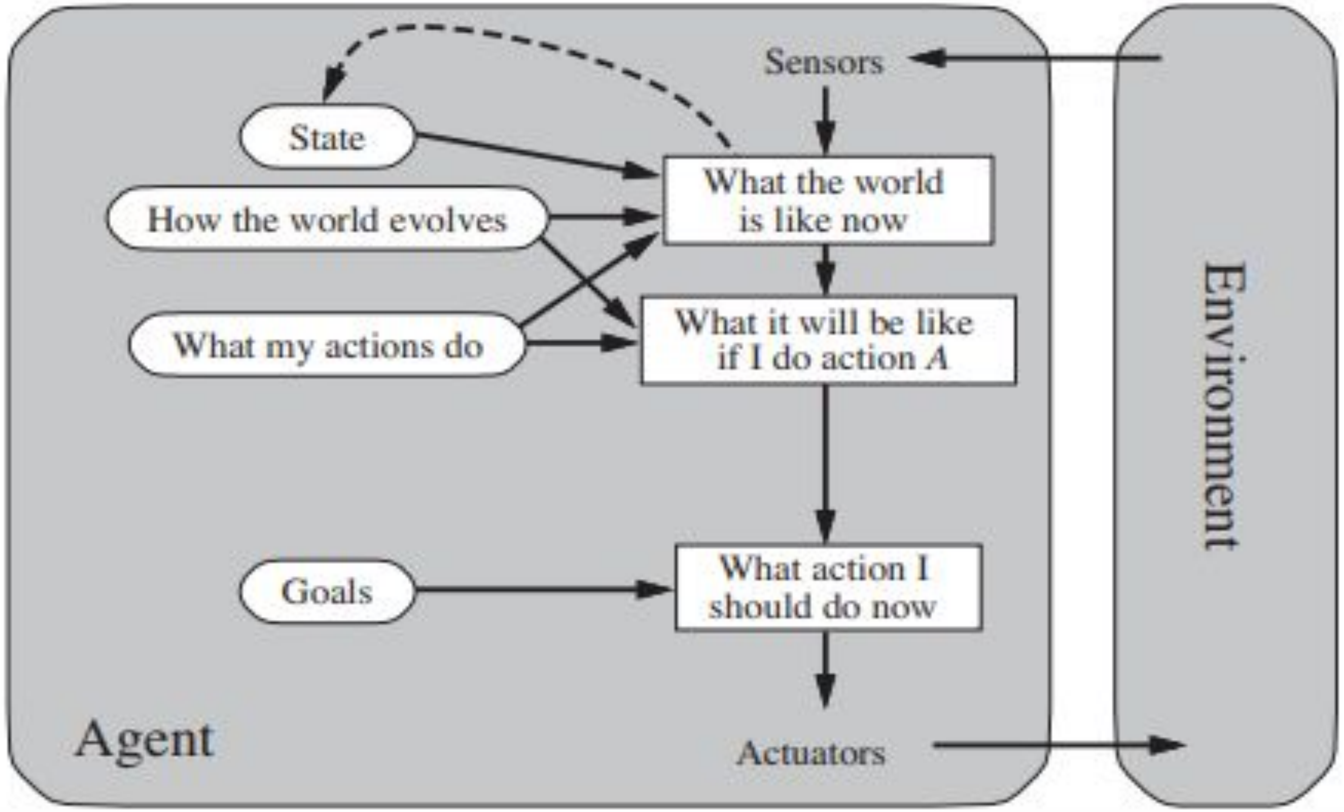


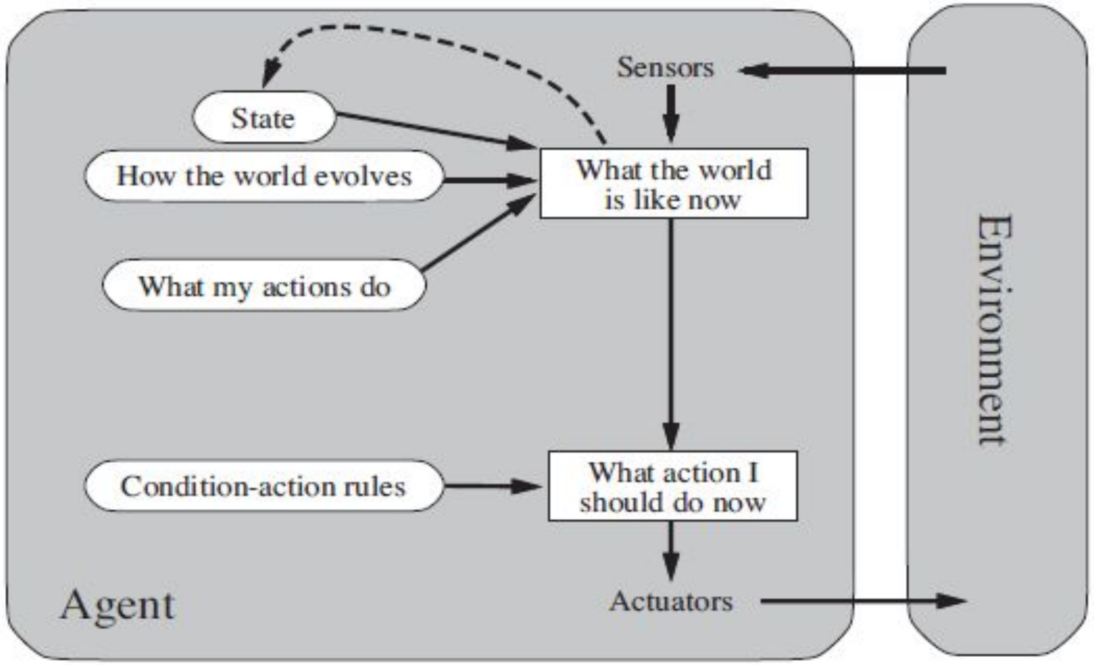
Figure 2.13 A model-based, goal-based agent. It keeps track of the world state as well as a set of goals it is trying to achieve, and chooses an action that will (eventually) lead to the achievement of its goals.

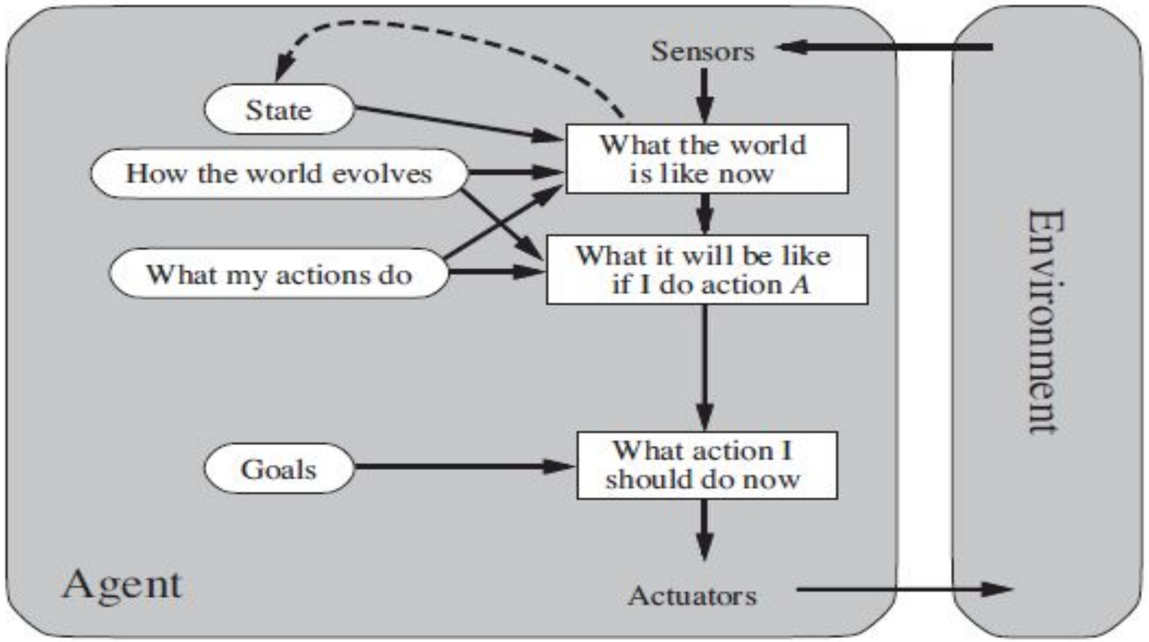
- Sometimes goal-based action selection is straightforward—for example, when goal satisfaction results immediately from a single action.
- Sometimes it will be more tricky—for example, when the agent has to consider long sequences of twists and turns in order to find a way to achieve the goal.
- Search and planning are the subfields of AI devoted to finding action sequences that achieve the agent's goals.

- The reflex agent brakes when it sees brake lights.
- 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.

- If it starts to rain, the agent can update its knowledge of how effectively its brakes will operate; this will automatically cause all of the relevant behaviors to be altered to suit the new conditions.
- For the reflex agent, on the other hand, we would have to rewrite many condition–action rules.
- The goal-based agent’s behavior can easily be changed to go to a different destination, simply by specifying that destination as the goal.

Goal-based Agents





A model-based reflex agent 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.

A model-based, goal-based agent 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 them achievement of its goals.

Utility-based agents

- Goals alone are not enough to generate high-quality behavior in most environments.
- For example, many action sequences will get the taxi to its destination (thereby achieving the goal) but some are quicker, safer, more reliable, or cheaper than others.
- Goals just provide a crude binary distinction between “happy” and “unhappy” states.
- A more general performance measure should allow a comparison of different world states according to exactly how happy they would make the agent.
- Because “happy” does not sound very scientific, economists and computer scientists use the term utility instead.

- 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.

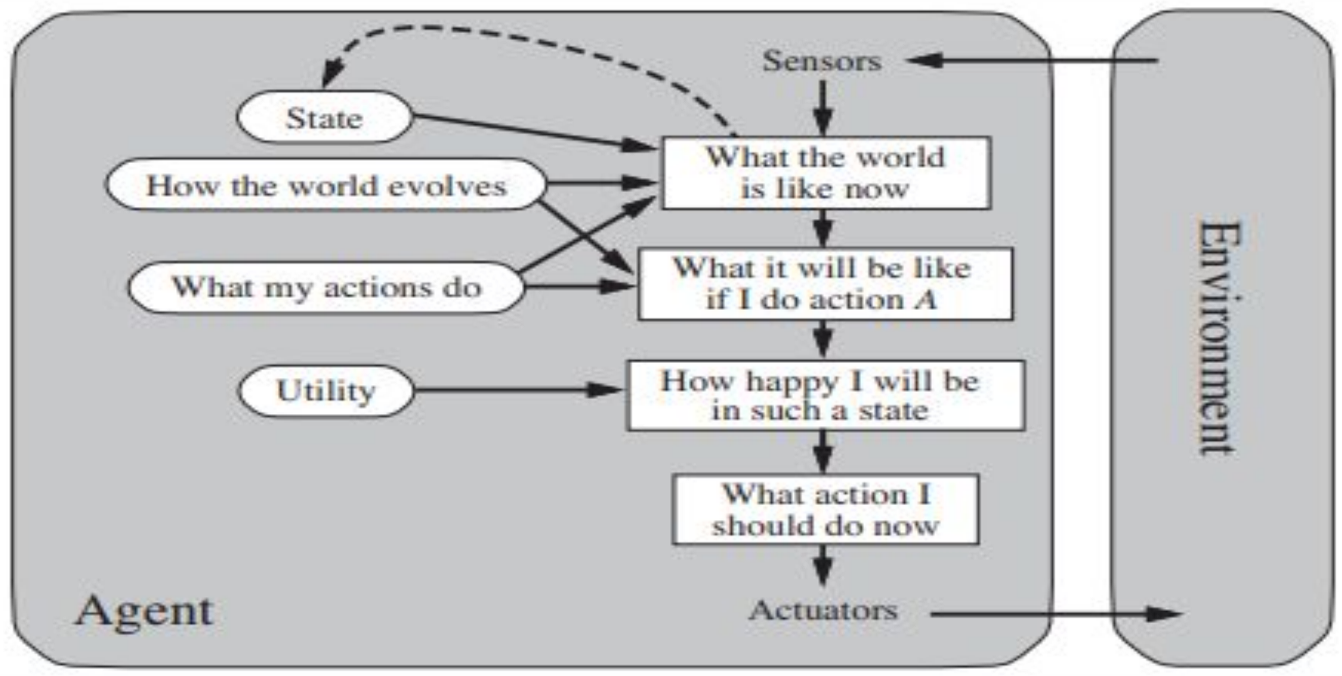


Figure 2.14 A model-based, utility-based agent. It uses a model of the world, along with a utility function that measures its preferences among states of the world. Then it chooses the action that leads to the best expected utility, where expected utility is computed by averaging over all possible outcome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

- A utility-based agent has to model and keep track of its environment, tasks that have involved a great deal of research on perception, representation, reasoning, and learning.

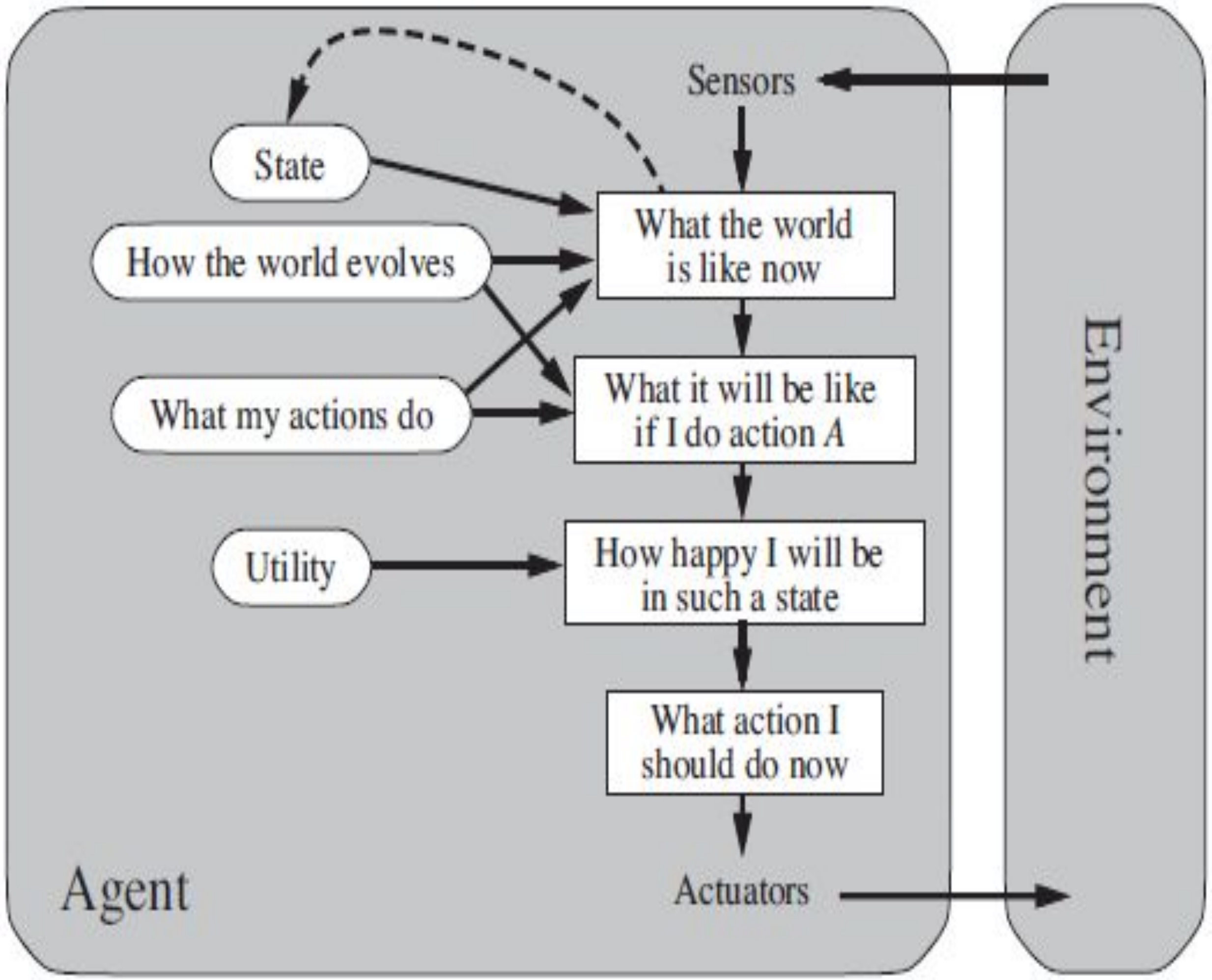
- A utility-based agent has many advantages in terms of flexibility and learning.
- In two kinds of cases, goals are inadequate but a utility-based agent can still make rational decisions.

1. When there are conflicting goals, only some of which can be achieved (for example, speed and safety), the utility function specifies the appropriate tradeoff.
2. 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.

- Partial observability and stochasticity are ubiquitous in the real world, and so, is decision making under uncertainty.
- **Expected utility** – the utility the agent expects to derive, on average, given the probabilities and utilities of each outcome.

- A rational utility-based agent chooses the action that maximizes the expected utility of the action outcomes.
- Any rational agent must behave *as if* it possesses a utility function whose expected value it tries to maximize.
- An agent that possesses an *explicit* utility function can make rational decisions with a general-purpose algorithm that does not depend on the specific utility function being maximized.
- Rational agent functions are those that have the highest performance.

Model-based, Utility-based Agents



A model-based, utility-based agent 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 overall possible outcome states, weighted by the probability of the outcome.

- Utility-based agent programs are designed as decision-making agents that must handle the uncertainty inherent in stochastic or partially observable environments.
- A Utility-based agent – builds agents that maximize expected utility. Such agents would be intelligent, but it’s not simple.
- A utility-based agent has to model and keep track of its environment – this involves tasks that need a great deal of research on perception, representation, reasoning, and learning.
- Choosing the utility-maximizing course of action is also a difficult task, requiring ingenious algorithms.
- Even with these algorithms, perfect rationality is usually unachievable in practice because of computational complexity.

Learning agents

- Turing (1950) considers the idea of actually programming his intelligent machines by hand.
- The method he proposes is to build learning machines and then to teach them.
- In many areas of AI, this is now the preferred method for creating state-of-the-art systems.
- Learning has another advantage: it allows the agent to operate in initially unknown environments and to become more competent than its initial knowledge alone might allow.

Learning agents

- A learning agent can be divided into four conceptual components.
- The most important distinction is between the learning element, which is responsible for making improvements, and the performance element, which is responsible for selecting external actions.
- The performance element is what we have previously considered to be the entire agent: it takes in percepts and decides on actions.
- 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.

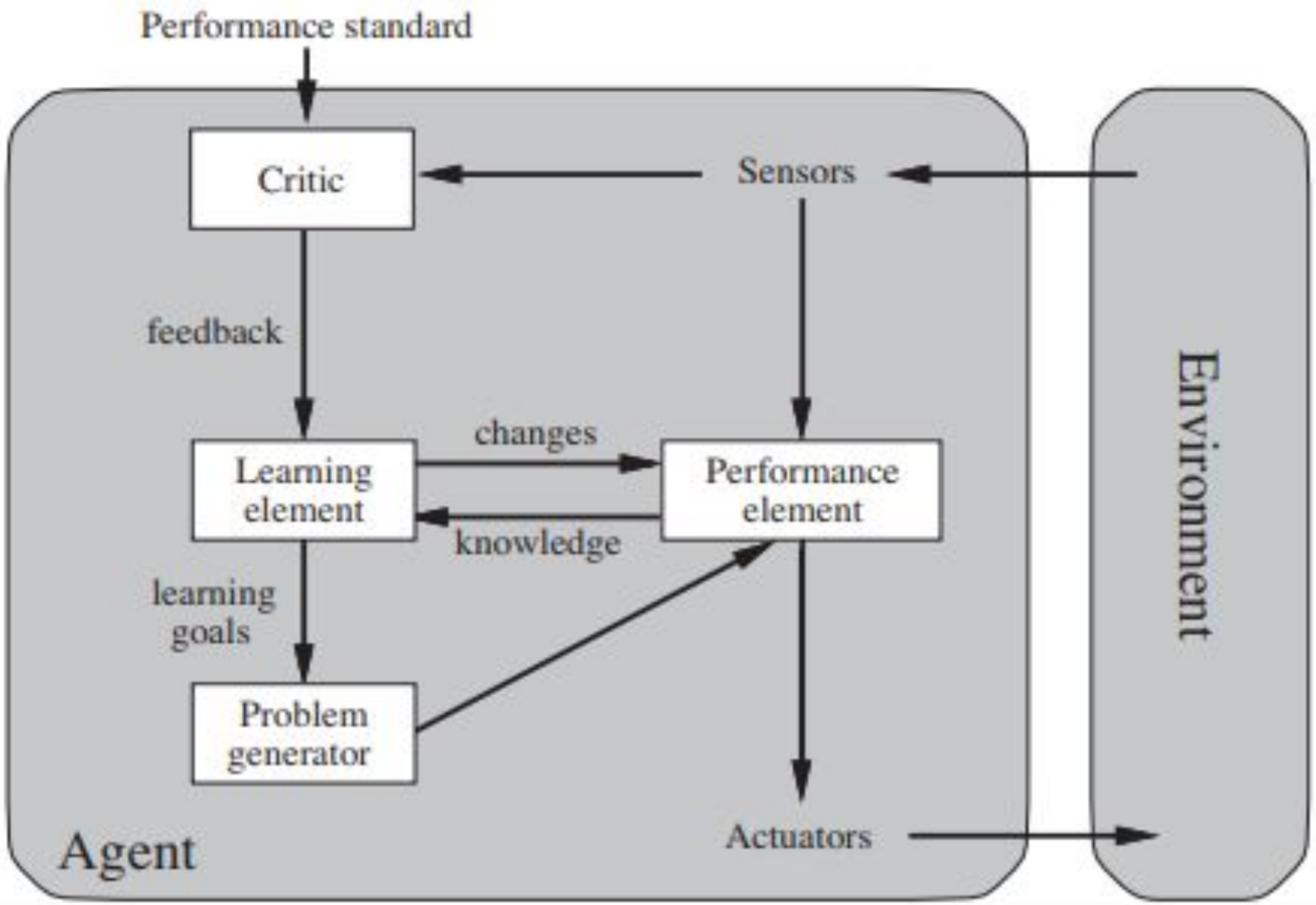


Figure 2.15 A general learning agent.

- The critic tells the learning element how well the agent is doing with respect to a fixed performance standard.
- The critic is necessary because the percepts themselves provide no indication of the agent's success.
- The last component of the learning agent is the problem generator. It is responsible for suggesting actions that will lead to new and informative experiences.
- The point is that if the performance element had its way, it would keep doing the actions that are best, given what it knows. But if the agent is willing to explore a little and do some perhaps suboptimal actions in the short run, it might discover much better actions for the long run. The problem generator's job is to suggest these exploratory actions.

- The performance standard distinguishes part of the incoming percept as a reward (or penalty) that provides direct feedback on the quality of the agent's behavior.
- How the components of agent programs work
- We can place the representations along an axis of increasing complexity and expressive power—atomic, factored, and structured.

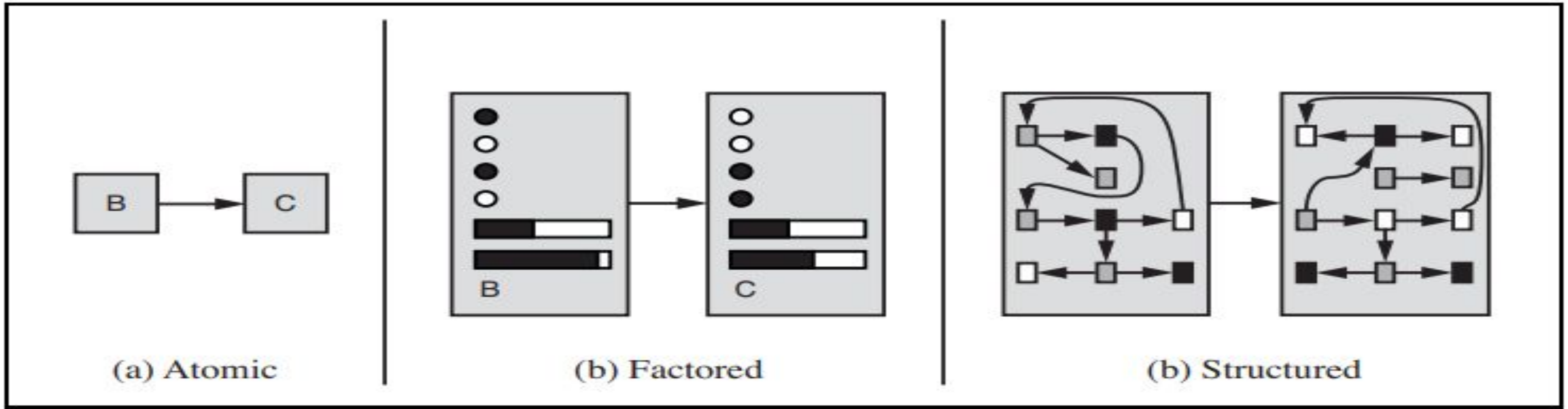


Figure 2.16 Three ways to represent states and the transitions between them. (a) Atomic representation: a state (such as B or C) is a black box with no internal structure; (b) Factored representation: a state consists of a vector of attribute values; values can be Boolean, real-valued, or one of a fixed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

- In an atomic representation each state of the world is indivisible—it has no internal structure.
- Consider the problem of finding a driving route from one end of a country to the other via some sequence of cities.
- For the purposes of solving this problem, it may suffice to reduce the state of world to just the name of the city we are in—a single atom of knowledge; a “black box” whose only discernible property is that of being identical to or different from another black box.
- The algorithms underlying search and game-playing , Hidden Markov models, and Markov decision processes all work with atomic representations.

- A factored representation splits up each state into a fixed set of variables or attributes, each of which can have a value.
- While two different atomic states ATTRIBUTE VALUE have nothing in common—they are just different black boxes—two different factored states can share some attributes; this makes it much easier to work out how to turn one state into another.
- Many important areas of AI are based on factored representations, including constraint satisfaction algorithms, propositional logic, Bayesian networks etc.

- A factored representation is unlikely to be pre-equipped with the attribute TruckAheadBackingIntoDairyFarmDrivewayBlockedByLooseCow

  with value true or false.
- Instead, we would need a structured representation, in which objects such as cows and trucks and their various and varying relationships can be described explicitly.
- Structured representations underlie relational databases and first-order logic, first-order probability models, knowledge-based learning and much of natural language understanding. In fact, almost everything that humans express in natural language concerns objects and their relationships.