Artificial Intelligence
DSE 3252

Introduction to AI

ROHINI R RAO & RASHMI LM
DEPT OF DATA SCIENCE & COMPUTER APPLICATIONS

JANUARY 2024

Thinking Humanly	**Thinking Rationally**
"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)	"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	**Acting Rationally**
"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)	"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)

Figure 1.1 Some definitions of artificial intelligence, organized into four categories.

Acting humanly: The Turing Test approach (1950)

The computer would need to possess the following capabilities:

- • natural language processing to enable it to communicate successfully in English
- • knowledge representation to store what it knows or hears
- • automated reasoning to use the stored information to answer questions and to draw new conclusions;
- • machine learning to adapt to new circumstances and to detect and extrapolate patterns

To pass the total Turing Test, the computer will need

- • computer vision to perceive objects, and
- • robotics to manipulate objects and move about.

Thinking humanly: The cognitive modeling approach

Determining how humans think through

- Introspection—trying to catch our own thoughts as they go by
- Psychological experiments—observing a person in action
- Brain imaging—observing the brain in action

From the theory of the mind it is possible to express theory as computer program

If the computer program’s input–output behavior matches corresponding human behavior, that is evidence that some of the program’s mechanisms could also be operating in humans

Cognitive science

- brings together computer models from AI and experimental techniques from psychology to construct precise and testable theories of the human mind

Thinking rationally: The “laws of thought” approach

Greek philosopher **Aristotle** was one of the first to attempt to codify “right thinking,” that

is, irrefutable reasoning processes.

Syllogisms

- patterns for argument structures that always yielded correct conclusions when given correct premises
- example - “Socrates is a man; all men are mortal; therefore, Socrates is mortal.”

These laws of thought were supposed to govern the operation of the mind

their study initiated the field called **LOGIC**

Main obstacles to approach

- it is not easy to take informal knowledge and state it in the formal terms required by logical notation, particularly when the knowledge is less than 100% certain
- there is a big difference between solving a problem “in principle” and solving it in practice

Acting rationally:
The rational agent approach

An **Agent** is just something that acts (agent comes from the Latin agere, to do)

Computer agents
- operate autonomously
- perceive their environment
- persist over a prolonged time period
- adapt to change
- create and pursue goals

As opposed to Laws of Thought Approach
- making correct inferences is part of being a rational agent, need to reason logically to the conclusion that a given action will achieve one's goals and then to act on that conclusion
- However there may be a need to act rationally without careful deliberation

Advantages
- More general than "laws of thought"
- Mathematically well defined, so can be used in applications

Foundations of AI

Philosophy

- *Can formal rules be used to draw valid conclusions?*
- *How does the mind arise from a physical brain?*
- *Where does knowledge come from?*
- *How does knowledge lead to action?*

Dualism- Materialism- Naturalism- Utilitiarinism etc.

Principle of Induction – general rules are acquired by exposure to repeated associations between their elements

Mathematics

- *What are the formal rules to draw valid conclusions?*
- *What can be computed?*
- *How do we reason with uncertain information?*

Formal Logic – George Boole proposed propositional or Boolean logic

Probability – generalizing logic to situations with uncertain information

Ronald Fisher is first modern **statistician**

Euclid’s **algorithm** for greatest common divisors

Alan Turing tried to characterize functions that are **computable**

Tractability – A problem is intractable if the time required to solve problems grows exponentially with the size of instances

NP-completeness – almost always intractable

Foundations of AI

Economics
• *How should we make decisions so as to maximize payoff?*
• *How should we do this when others may not go along?*
• *How should we do this when the payoff may be far in the future?*

Decision Theory combines probability with utility, provides a framework for individual decisions.
Game Theory – rational agent should adopt policies that are randomized
Operations Research
Markov Decision Process

Neuroscience
• *How do brains process information?*
Study of nervous system , particularly the brain
Measurement of intact brain activity began with invention of **Electroencephalograph** (EEG)
Optogenetics - Single cell electrical recording of neuron activity
Brain-Machine Interface – for both sensing and motor control to restore function to disabled individuals

Foundations of AI

Psychology

• *How do humans and animals think and act?*

Cognitive Psychology views the brain as an information-processing device
Perception involved a form of unconscious logical inference

Knowledge based agent has 3 steps

1. The stimulus must be translated to internal representation
2. The representation is manipulated by cognitive process to derive new internal representations
3. These are in turn retranslated back into action

Human Computer Interaction – idea of intelligent augmentation rather than AI.
Computers should augment human abilities rather than automate away human tasks

Foundations of AI

Computer engineering

• *How can we build an efficient computer?*

Moores law- each generation of computer hardware has brought an increase in speed and capacity and a decrease in price

Quantum Computing

Control theory and cybernetics

• *How can artifacts operate under their own control?*

Modern Control Theory

Viewed purposive behavior as arising from a regulatory mechanism trying to minimize error

Goal is to design systems that minimize cost function

Linguistics

• *How does language relate to thought?*

Computations Linguistics, NLP requires

- Understanding the structure of sentences
- understanding of the subject matter and context

Knowledge Representation is tied to language

Rational Agent

A **rational agent** strives to do the right thing, based on what it can perceive and the action it can perform

Performance measure: An objective criterion for success of agent’s behavior

Example : Vacuum Cleaner Robot

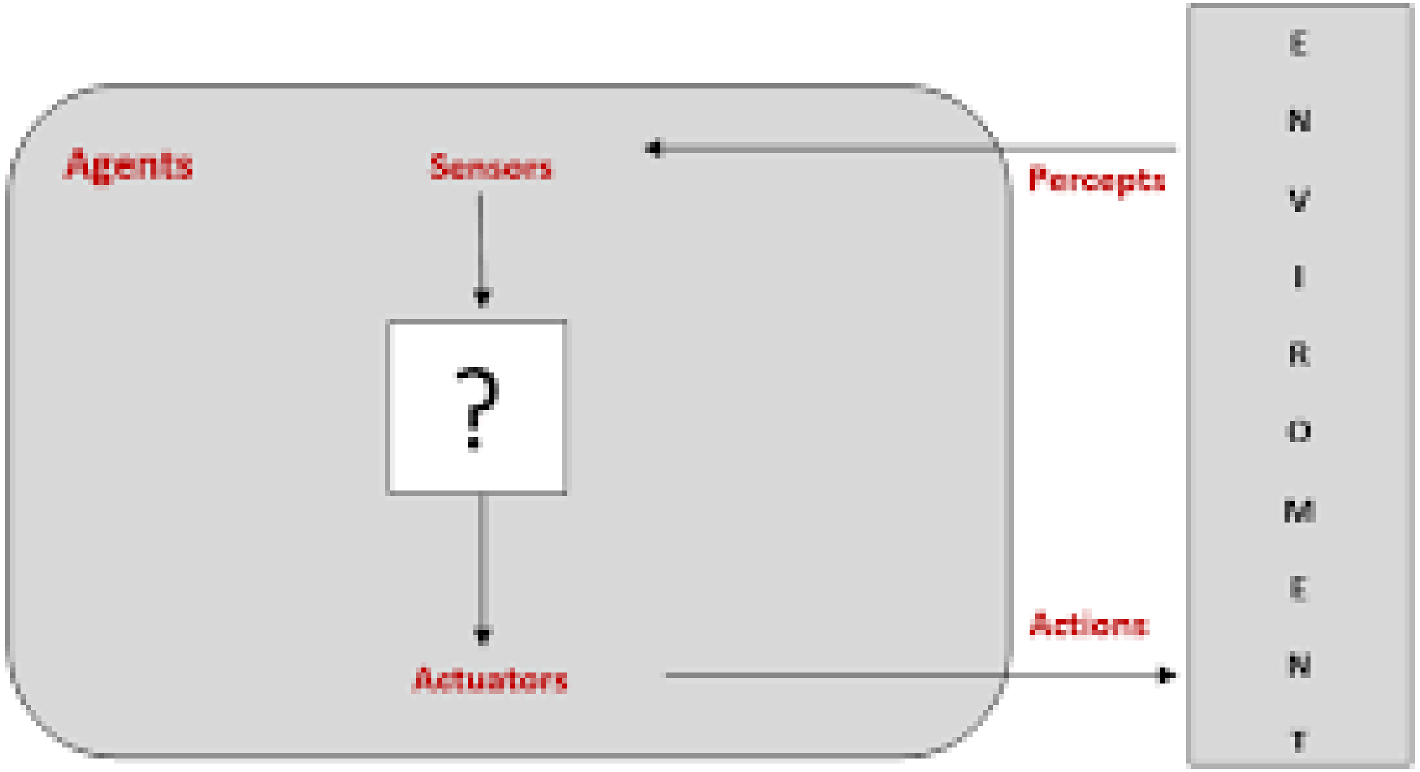
Performance Criterion

- How must dust has to be collected?
- How much electricity can be consumed?
- How much time should it take?
- How much noise can be tolerated?

Ideal Rational Agent

For every possible percept sequence, it does whatever action is expected to maximize its performance measure on the basis of evidence perceived so far and built in knowledge

Agents interact with environments through sensors and actuators



Agent

- **Percept** to refer to the agent's perceptual inputs at any given instant
- An agent's **percept sequence** is the complete history of everything the agent has ever perceived
- In general, 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
- An agent's behavior is described by the **agent function** that maps any given percept sequence to an action
- Internally, the agent function for an artificial agent will be implemented by an **agent program**.

Good Behaviour : The concept of rationality

- **Doing the right thing** by considering the consequences of the agent's behavior
- 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
- This notion of desirability is captured by a performance measure that evaluates any given sequence of environment states

Rationality at any given time depends on :

- • 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

As a general rule, 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

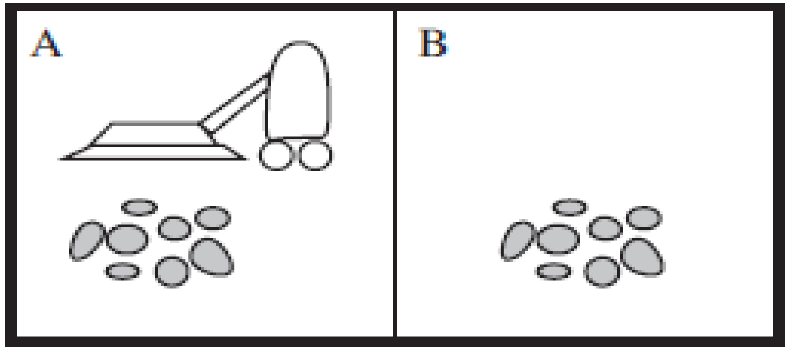


Figure 2.2 A vacuum-cleaner world with just two locations.

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.

Omniscience

- An omniscient agent knows the actual outcome of its actions and can act accordingly
- but **omniscience** is impossible in reality.
- **Rationality** maximizes expected performance, while perfection maximizes actual performance
- rationality does not require omniscience, then, because the rational choice depends only on the percept sequence to date
- Doing actions in order to modify future percepts—sometimes called **information gathering**
- Example - exploration that must be undertaken by a vacuum-cleaning agent in an initially unknown environment

Learning & Autonomy

- Learn from what it perceives
- The agent's initial configuration could reflect some prior knowledge of the environment, but as the agent gains experience this may be modified and augmented.
- There are extreme cases in which the environment is completely known a priori.
- In such cases, the agent need not perceive or learn; it simply acts correctly
- An agent relies on the prior knowledge of its designer rather than on its own percepts, we say that the agent lacks **autonomy**

Specifying the task environment

PEAS

- Consider, e.g., the task of designing an automated taxi driver:

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.

Examples of agent types and their PEAS descriptions

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.

Properties of task environments

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 the task environment is fully observable.
- if the sensors detect all aspects that are relevant to the choice of action
- 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
  - for example, a vacuum agent with only a local dirt sensor cannot tell whether there is dirt in other squares
- If the agent has no sensors at all then the environment is unobservable

Single agent vs. multiagent

- Example – Single agent - solving a crossword puzzle , Two agent – playing chess
- An entity becomes an agent when optimizing one agent's performance affects other agent
- Multiagent environment
  - Communication is part of rational behaviour
  - chess is a **competitive**
  - Taxi driving is **cooperative**

Properties of task environments

Deterministic vs. stochastic

- **Deterministic**- If the next state of the environment is completely determined by the current state and the action executed by the agent
- otherwise, it is **Stochastic -** uncertainty about outcomes is quantified in terms of probabilities
- Environment is uncertain if it is not fully observable or not deterministic
- **Nondeterministic** environment - actions are characterized by their possible outcomes, but no probabilities are attached to them

Episodic vs. sequential:

- In an **Episodic** task environment, the agent's experience is divided into atomic episodes.
- In each episode the agent receives a percept and then performs a single action.
- next episode does not depend on the actions taken in previous episodes
  - Example - an agent that is spotting defective parts on an assembly line
- In **sequential** environments, the current decision could affect all future decisions

Properties of Task environments

Static vs. dynamic:

- **Dynamic** If environment can change while an agent is deliberating otherwise, it is static
- Example – Taxi driving is dynamic, playing a puzzle is static

Discrete vs. continuous:

- applies to the state of the environment, to the way time is handled, and to the percepts and actions of the agent
- Example – Chess has finite number of discrete state
- **Taxi driving**
  - is a continuous-state and continuous-time problem: the speed and location of the taxi are a range of continuous values over time.
  - actions are also continuous (steering angles, etc.).

Known vs. unknown:

- agent's (or designer's) state of knowledge about the "laws of physics" of the environment
- In **Known** environment, the outcomes (or outcome probabilities if the environment is stochastic) for all actions are given

Hardest agent is partially observable, multiagent, stochastic, sequential, dynamic, continuous, and unknown

Examples of task environments and their characteristics

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.

Structure of Agents

- The job of AI is to design an agent program that implements the agent function
- program will run on some sort of computing device with physical sensors and actuators

agent = architecture + program

- Agent Programs take the current percept as input from the sensors and return an action to the actuators

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.

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

Structure of Agents

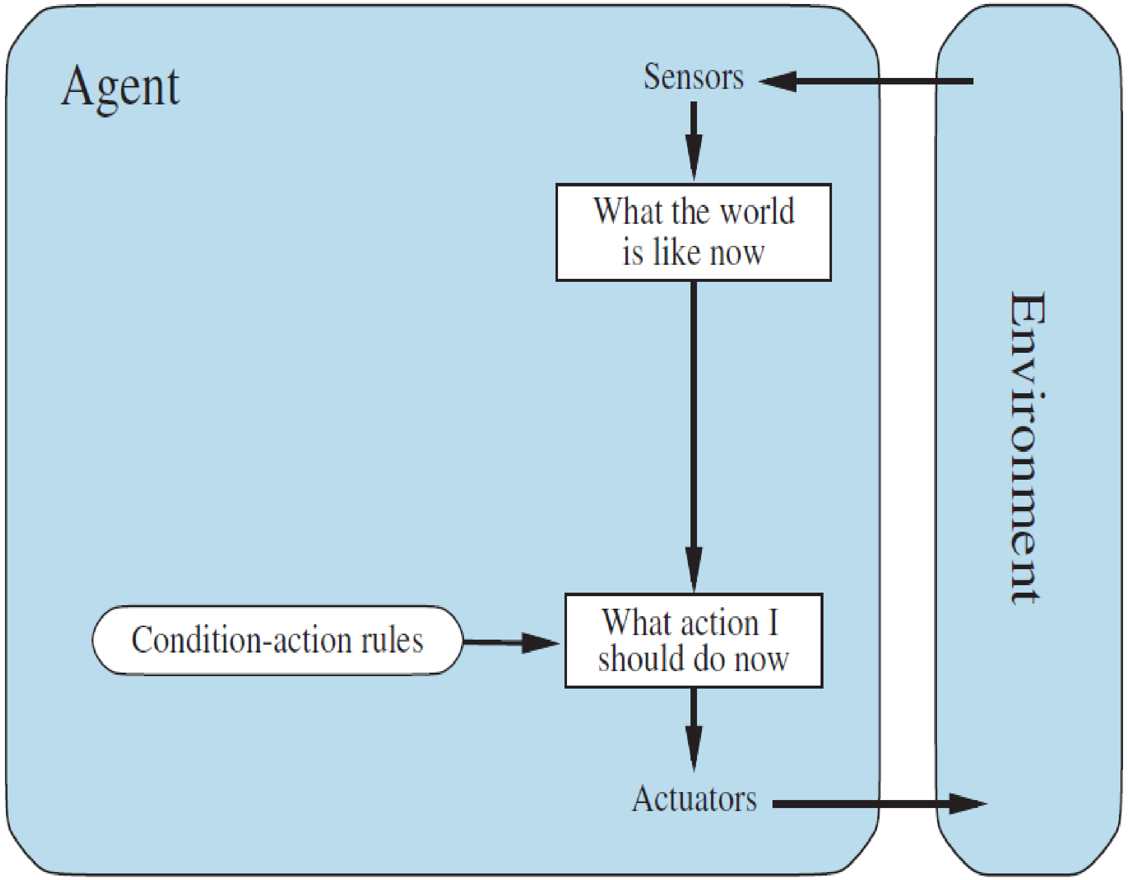
Simple reflex agents

- agents select actions on the basis of the current percept, ignoring the rest of the percept history
- Example – Vacuum Agent has location sensor and dirt sensor

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*

Reflex Agents

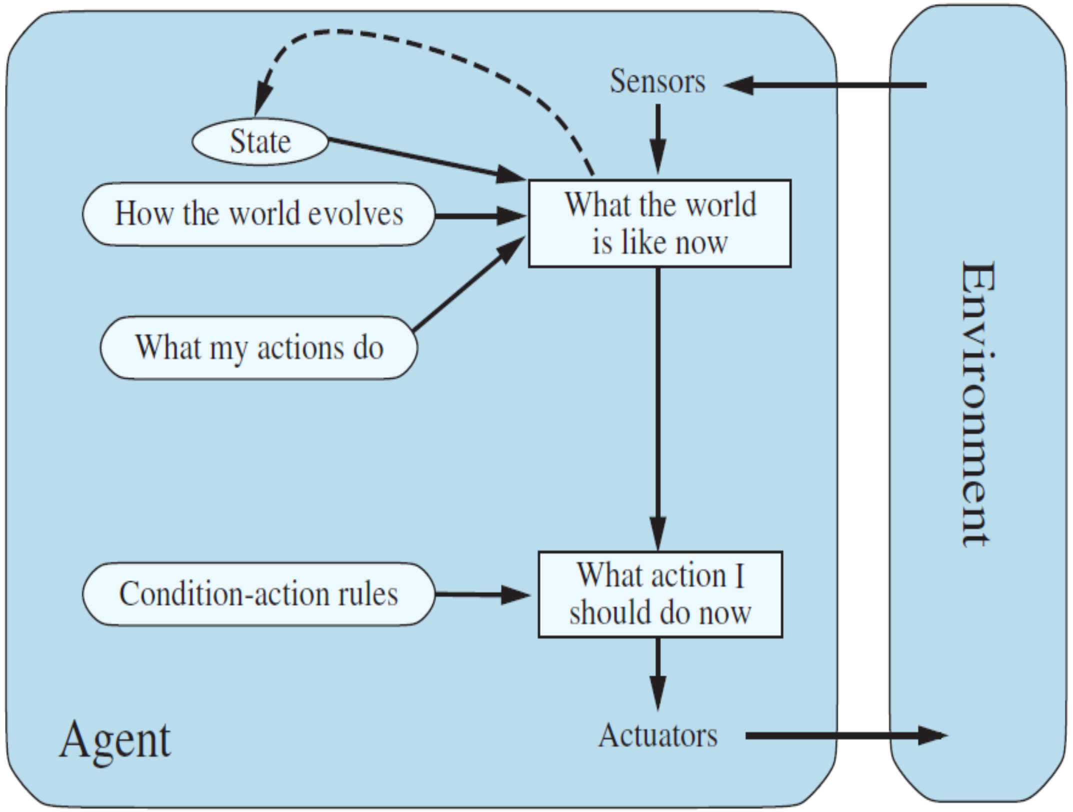


function SIMPLE-REFLEX-AGENT(*percept*) **returns** an action
persistent: *rules*, a set of condition–action rules

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

condition–action rule:
if *car-in-front-is-braking* **then** *initiate-braking*.

Model based Reflex Agent

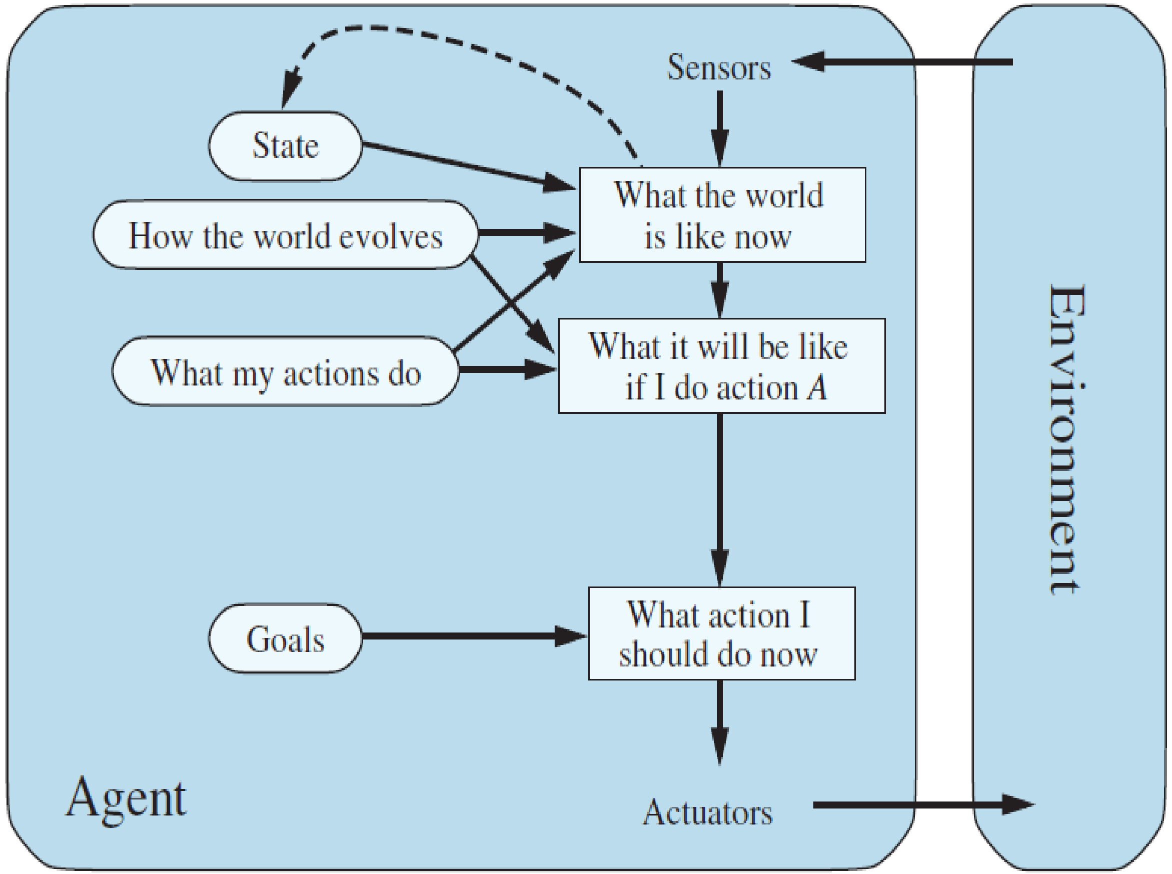


- the agent should maintain some sort of **internal state** that depends on the percept history
- Which reflects at least some of the unobserved aspects of the current state

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

  $state \leftarrow$ UPDATE-STATE($state, action, percept, transition\_model, sensor\_model$)
  $rule \leftarrow$ RULE-MATCH($state, rules$)
  $action \leftarrow rule$.ACTION
  return $action$

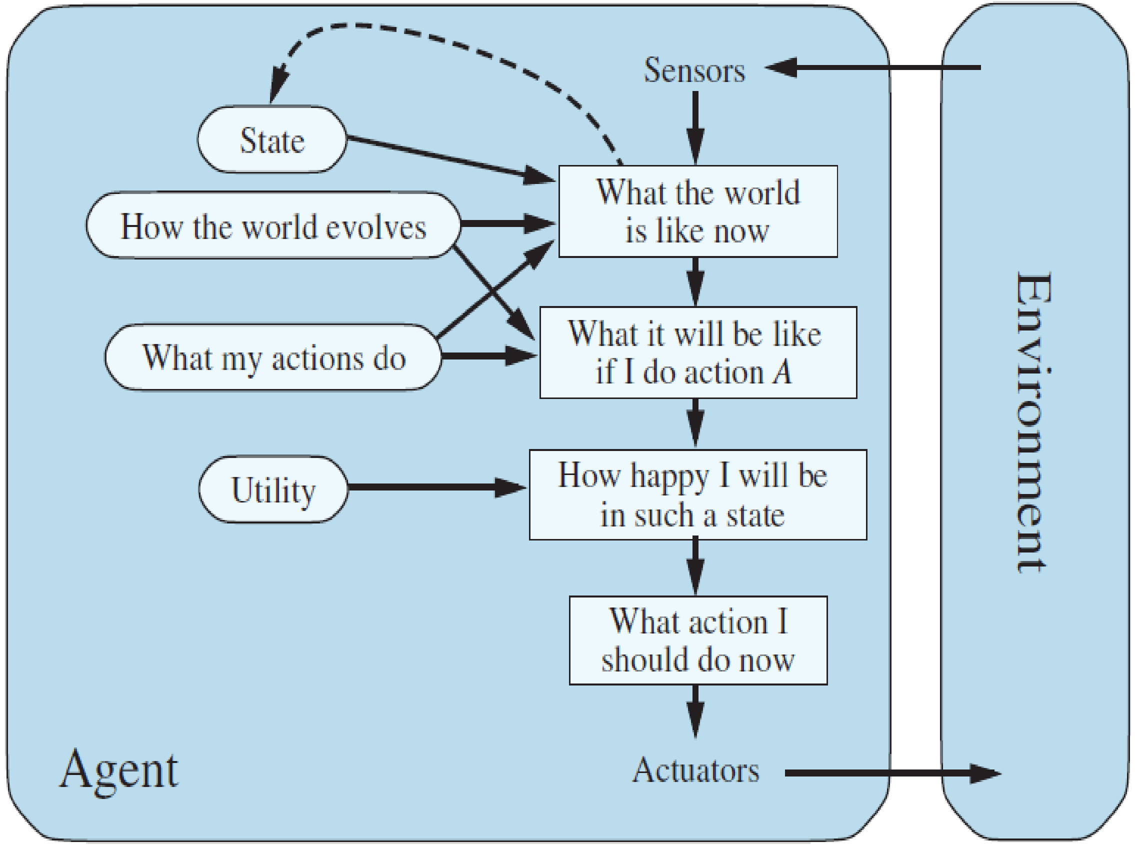
Model based, Goal based agent



The agent program can combine

- The model
- choose actions that achieve the goal
- **Search** and **Planning** are subfields of AI devoted to finding action sequences that achieve the agent's goals

Model based , Utility based agent



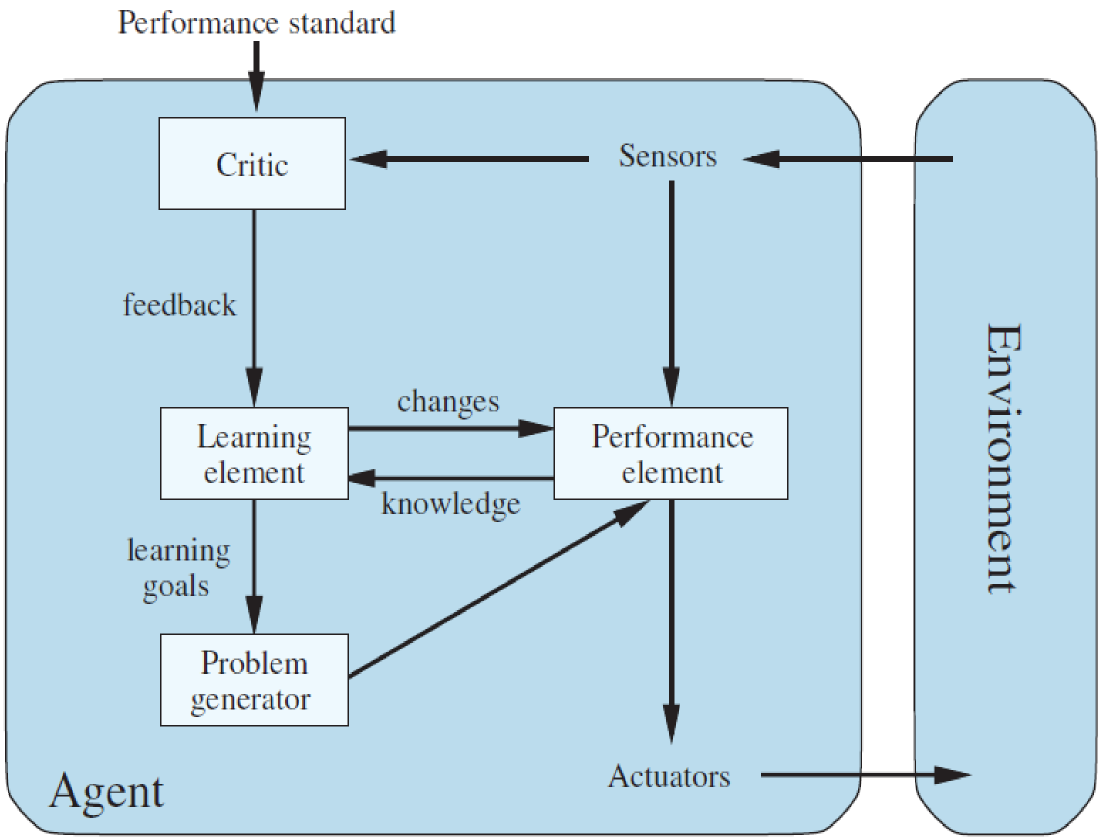
Performance measure

- assigns a score to any given sequence of environment states, so it can easily distinguish between good and bad action

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

- If there are multiple goals, the utility function specifies the appropriate **trade-off**
- 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

General Learning Agent



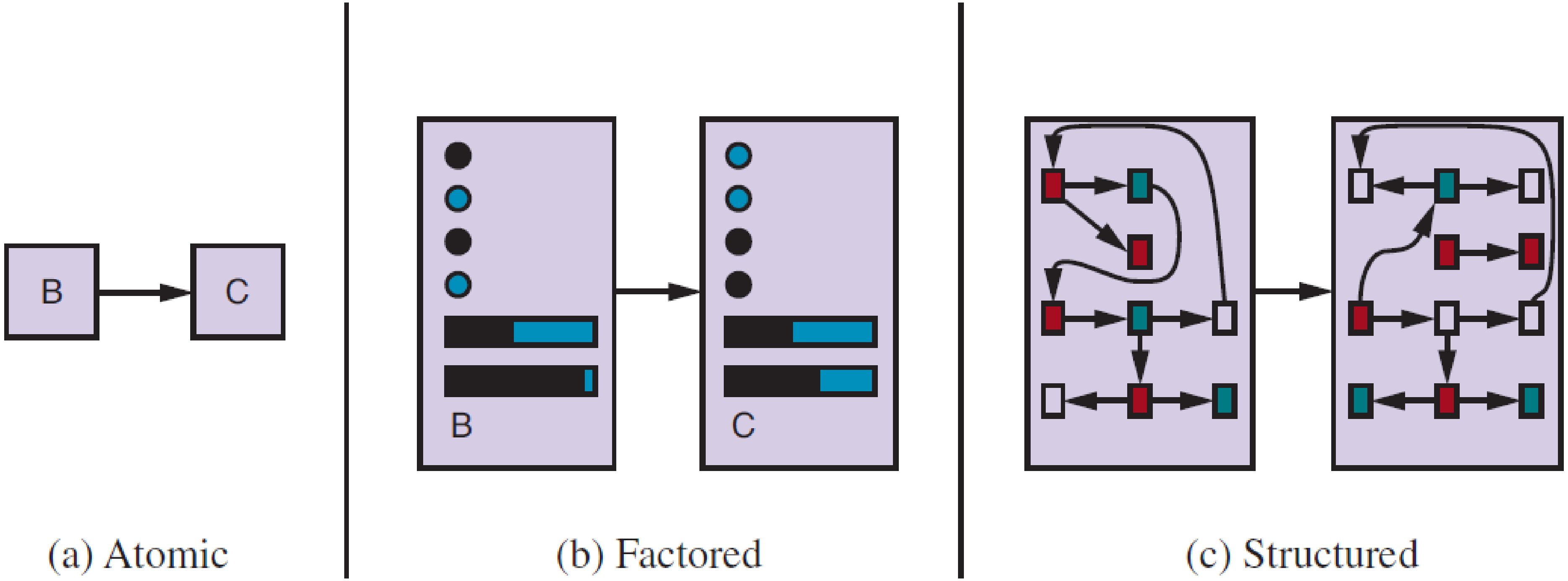
Learning element which is responsible for making improvements
Performance element, which is responsible for selecting external actions
Critic tells the learning element how well the agent is doing with respect to a fixed performance standard
Problem Generator which is responsible for suggesting actions that will lead to new and informative experiences.

Example of Learning Agent

Automated taxi

- **Performance element** consists of whatever collection of knowledge and procedures the taxi has for selecting its driving actions.
- **Critic** observes the world and passes information along to the learning element.
- **Problem generator** might identify certain areas of behavior in need of improvement and suggest experiments
  - Example: Trying out the brakes on different road surfaces under different conditions.

Representing States and Transitions



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

References

1. Russell S., and Norvig P., Artificial Intelligence A Modern Approach (3e), Pearson 2010