CS3243: INTRO. TO AI

Semester 2, 2014-2015

Outline

- Course Administration
- Introduction to Artificial Intelligence (AI)
- Intelligent Agents

Course Administration

- Teaching Staff
- Teaching Resources
- Objective
- Syllabus
- Assessment Overview

Teaching Staff

- Lecturer: Bryan Low

  Email: lowkh@comp.nus.edu.sg

  Website: http://www.comp.nus.edu.sg/~lowkh

  Office: COM2-02-58

  Consultation hours: By appointment

  Research: Multi-agent systems, machine learning, robotics

- TAs: Zhang Yehong, Son Jaemin

  Email: zyhredleaf@gmail.com, woalsdnd@gmail.com

  Consultation hours: By appointment

Teaching Resources: IVLE

http://ivle.nus.edu.sg/

- Lesson Plan
- Lectures, Tutorials, Supplementary Materials, Homeworks
- Discussion forum
  - Any questions related to the course should be raised on this forum
  - Emails to me will be considered public unless otherwise specified
- Announcements
- Homework submissions
- Webcasts

Objective

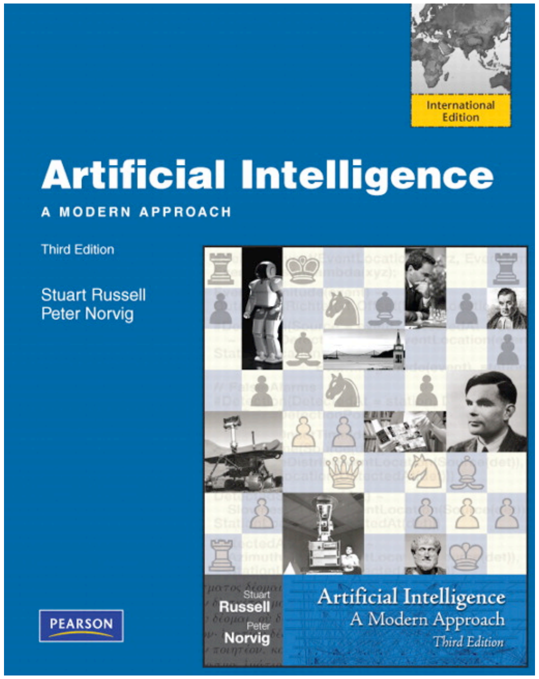
- Understand the essential concepts and foundation of Artificial Intelligence. These basic concepts include search, game playing, logic, uncertainty, probabilistic reasoning, and machine learning.

- Who?
  - Undergraduates and beginning graduate students. Centered towards CS or by permission.

Minerva: Museum Tour-Guide Robot

Readings

- Textbook:
  - Russell and Norvig (2010).
    Artificial Intelligence: A Modern Approach
    (3rd Edition ← Important!)
  - Online Resources, Code, and ERRATA:
    http://aima.cs.berkeley.edu/



Syllabus

- Introduction and Agents (chapters 1, 2)
- Search (chapters 3, 4, 5, 6)
- Logic (chapters 7, 8, 9)
- Uncertainty (chapters 13, 14)
- Machine Learning (chapter 18)

Assessment Overview

- **Midterm Exam** 20%

  9 Mar 2015 (During lecture, NO make-up)
- **Final Exam** 50%

  29 Apr 2015 (Afternoon)
- **Term Project: TBA** 25%
- **Graded Tutorials & Class Attendance** 5%

Public Holidays > Makeup Lectures

- 26 Jan (Week 3) AAAI Conference > 23 Feb (Recess Week) I3 Auditorium 12-2pm
- 2 Feb (Week 4) AAAI Conference > 5 Feb (Week 4, Thurs) I3 Auditorium 12-2pm

- 20 Feb (Week 6) Chinese New Year: NO tutorial sessions
- 3 Apr (Week 11) Good Friday: NO tutorial sessions

Freedom of Information Rule

- Collaboration is acceptable and encouraged
- You must always write the name(s) of your collaborators on your assignment.
- You will be assessed for the parts for which you claim is your own contribution.

On Collaboration

- You are free to meet with fellow students(s) and discuss assignments with them.
- Writing on a board or shared piece of paper is acceptable during the meeting; however, you **may not take any written (electronic or otherwise) record away from the meeting**.
- After the meeting, do something else for at least a half-hour before working on the assignment.
- This will ensure that you are able to reconstruct what you learned from the meeting, by yourself.

INTRODUCTION

AIMA Chapter 1

Outline

- What is AI?
- A brief history
- The state of the art

What is AI?

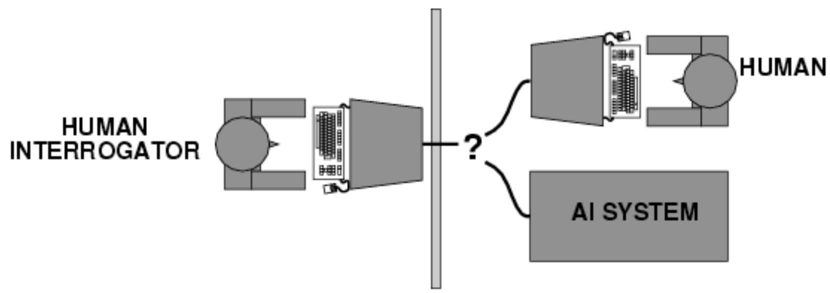
Views of AI fall into four categories:

Thinking humanly	Thinking rationally
Acting humanly	Acting rationally

AIMA advocates "acting rationally"

Acting Humanly: Turing Test

- Turing (1950). Computing Machinery and Intelligence:
  "Can machines think?" → "Can machines behave intelligently?"
- Operational test for intelligent behavior: The Imitation Game



- Predicted that, by 2000, a machine might have a 30% chance of fooling a lay person for 5 minutes
- Anticipated all major arguments against AI in following 60 years
- Suggested major components of AI: knowledge, reasoning, language understanding, learning

Thinking Humanly: Cognitive Modeling

- 1960s "cognitive revolution": information-processing psychology
- Requires scientific theories of internal activities of the brain
  - How to validate? Requires

    (1) Predicting and testing behavior of human subjects, or

    (2) Direct identification from neurological data
- Both approaches (roughly, Cognitive Science and Cognitive Neuroscience) are now distinct from AI

Thinking Rationally: "Laws of Thought"

Plato (left) and Aristotle (right), a detail of *The School of Athens*, a fresco by Raphael. Aristotle gestures to the earth, representing his belief in knowledge through empirical observation and experience.

- Aristotle: what are correct arguments/thought processes?
- Several Greek schools developed various forms of *logic*: *notation* and *rules of derivation* for thoughts; may or may not have proceeded to the idea of mechanization
- Problems:
  - Not all intelligent behavior can be mediated by logical deliberation
  - Being able to solve a problem "in principle" does not imply that it can be solved tractably in practice

Acting Rationally: Rational Agent

- Rational behavior: doing the "right thing"
- What is the "right thing" to do?
  That which is expected to achieve best outcome, given the available information
- Doesn't necessarily involve thinking – e.g., blinking reflex

Rational Agents

- An agent is an entity that perceives and acts
- This course is about designing rational agents
- Abstractly, an agent is a function from percept histories to actions, i.e., $f : P^* \rightarrow A$
- For any given class of environments and tasks, we seek the agent (or class of agents) with the best performance
- Caveat: computational limitations make perfect rationality unachievable

  → design best program for the given machine resource

AI Prehistory

Philosophy	Logic, methods of reasoning, mind as physical system foundations of learning, language, rationality
Mathematics	Formal representation and proof algorithms, computation, (un)decidability, (in)tractability, probability
Economics	Utility, decision theory
Neuroscience	Physical substrate for mental activity
Psychology	Phenomena of perception and motor control, experimental techniques
Computer engineering	Building fast computers
Control theory	Design systems that maximize an objective function over time
Linguistics	Knowledge representation, grammar

Abridged History of AI

- 1943 McCulloch & Pitts: Boolean circuit model of brain
- 1950 Turing's "Computing Machinery and Intelligence"
- 1956 Dartmouth meeting: "Artificial Intelligence" adopted
- 1952–69 Look, Ma, no hands!
- 1950s Early AI programs, including Samuel's checkers program, Newell & Simon's Logic Theorist, Gelernter's Geometry Engine

Abridged History of AI

- 1965 Robinson's complete algorithm for logical reasoning
- 1966–73 AI discovers computational complexity
  Neural network research almost disappears
- 1969–79 Early development of knowledge-based systems
- 1980– AI becomes an industry
- 1986– Neural networks return to popularity
- 1987– AI becomes a science
- 1995– The emergence of intelligent agents

State of the Art

- IBM’s Deep Blue defeated the reigning world chess champion Garry Kasparov in 1997
- CMU’s BOSS (self-driving SUV) won Urban Challenge in 2007: it can drive safely in traffic, obey traffic rules, avoid pedestrians, perform complex maneuvers (e.g. parking, negotiating intersections) in urban environment
- During the 1991 Gulf War, US forces deployed an AI logistics planning and scheduling program that involved up to 50,000 vehicles, cargo, and people
- NASA's on-board autonomous planning program controlled the scheduling of operations for a spacecraft
- IBM Watson competed and won on Jeopardy in 2011

INTELLIGENT AGENTS

AIMA Chapter 2

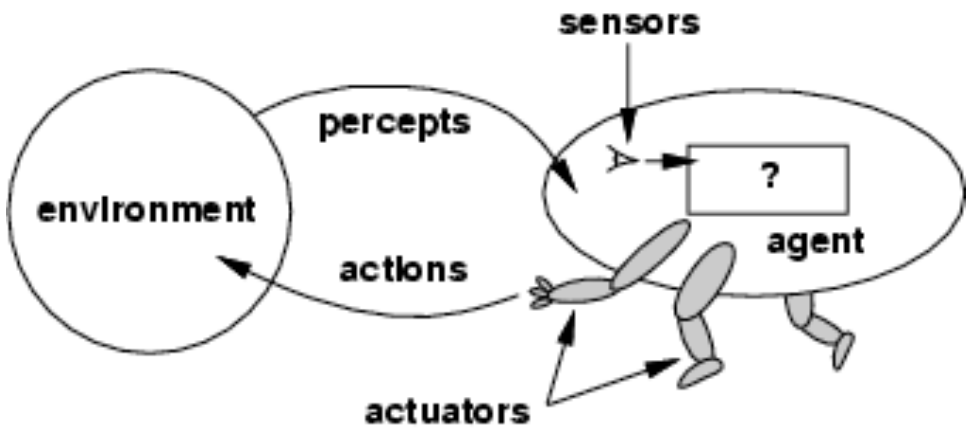
Outline

- Agents and environments
- Rational agent
- Specifying task environment: PEAS
- Properties of task environment
- Agent types

Agents

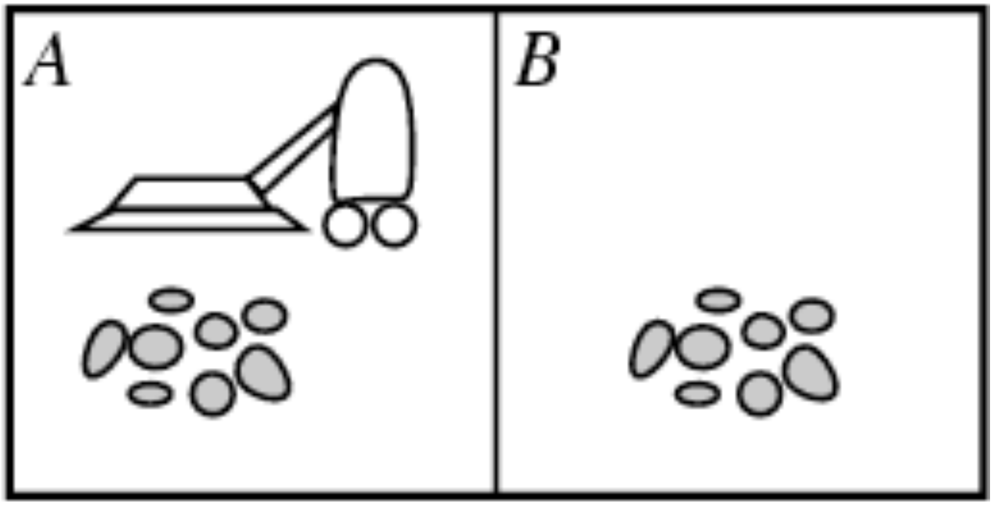
- An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators
- Human agent: eyes, ears, and other organs for sensors; hands, legs, mouth, and other body parts for actuators
- Robotic agent: cameras and laser range finders for sensors; various motors for actuators

Agents and Environments



- The agent function maps from percept histories/ sequences to actions, i.e., $f : P^* \rightarrow A$
- The agent program runs on the physical architecture to perform f
- agent = architecture + program

Vacuum-Cleaner World



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

Vacuum-Cleaner Agent Function

Percept sequence	Action
$[A, Clean]$	$Right$
$[A, Dirty]$	$Suck$
$[B, Clean]$	$Left$
$[B, Dirty]$	$Suck$
$[A, Clean]$, $[A, Clean]$	$Right$
$[A, Clean]$, $[A, Dirty]$	$Suck$
$\vdots$	$\vdots$

Rational Agents

- 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: objective criterion for measuring 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 Agents

- Rational Agent: For each possible percept sequence, a rational agent should select an action that maximizes its expected performance, given the evidence provided by the percept sequence and whatever built-in knowledge the agent has.

Rational Agents

- Rationality is different from omniscience (all-knowing with infinite knowledge)
- Agents can perform actions in order to modify future percepts so as to obtain useful information (information gathering, exploration)
- An agent is autonomous if its behavior is determined by its own experience (with ability to learn and adapt)

Specifying Task Environment: 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
  - Environment
  - Actuators
  - Sensors

Specifying Task Environment: PEAS

- 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

Specifying Task Environment: 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

Specifying Task Environment: 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)

Specifying Task Environment: PEAS

Agent: Interactive English tutor

- Performance measure: Maximize student's score on test
- Environment: Set of students, testing agency
- Actuators: Screen display (exercises, suggestions, corrections)
- Sensors: Keyboard entry

Properties of Task Environments

- Fully observable (vs. partially observable): An agent's sensors give it access to the complete state of the environment at each point in time.
- Deterministic (vs. stochastic): The next state of the environment is completely determined by the current state and the action executed by the agent.
- 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 does not depend on actions in past episodes.

Properties of Task Environments

- Static (vs. dynamic): The environment is unchanged while an agent is deliberating.
- Discrete (vs. continuous): A finite number of distinct states, percepts, and actions.
- Single agent (vs. multi-agent): An agent operating by itself in an environment.

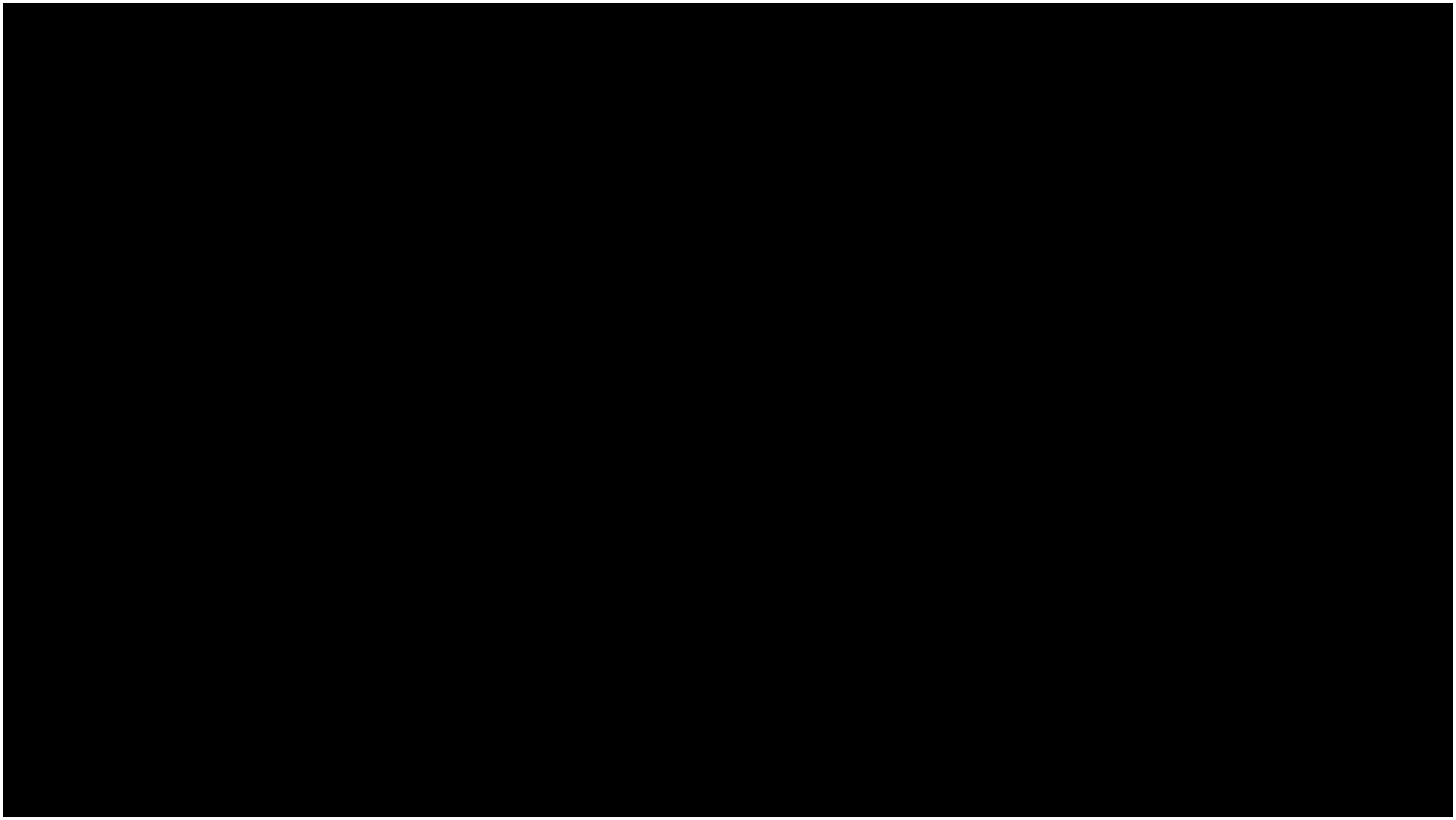
Properties of Task Environments

Task Environment	Crossword puzzle	Part-picking robot	Taxi driving
Fully observable	Yes	No	No
Deterministic	Yes	No	No
Episodic	No	Yes	No
Static	Yes	No	No
Discrete	Yes	No	No
Single agent	Yes	Yes	No

The properties of task environment largely determine the agent design.

The real world is (of course) partially observable, stochastic, sequential, dynamic, continuous, multi-agent.

Active Multi-Camera Surveillance



Agent Functions and Programs

- An agent is completely specified by the agent function mapping percept sequences to actions
- One agent function (or a small equivalence class) is rational

- Aim: Find a way to implement the rational agent function concisely

Table-Lookup Agent

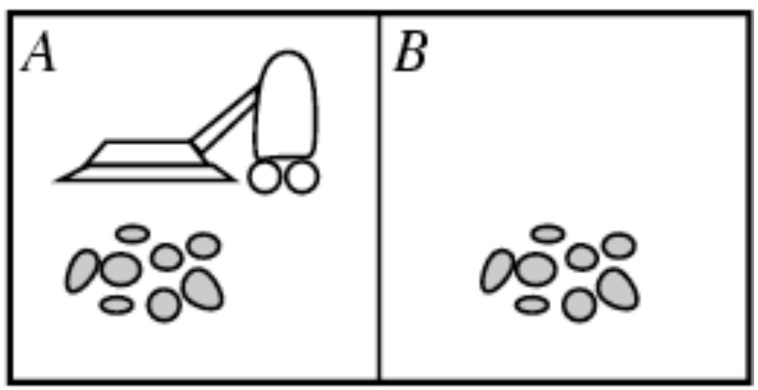

```
function TABLE-DRIVEN-AGENT(percept) returns action
    static: percepts, a sequence, initially empty
            table, a table of actions, indexed by percept sequences, fully specified

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

- Drawbacks:
  - Huge table to store
  - Take a long time to build the table
  - No autonomy: no agent can ever learn all the correct table entries from its experiences
  - Designer has no guidance on filling in the correct table entries

Vacuum-Cleaner Agent Program

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*



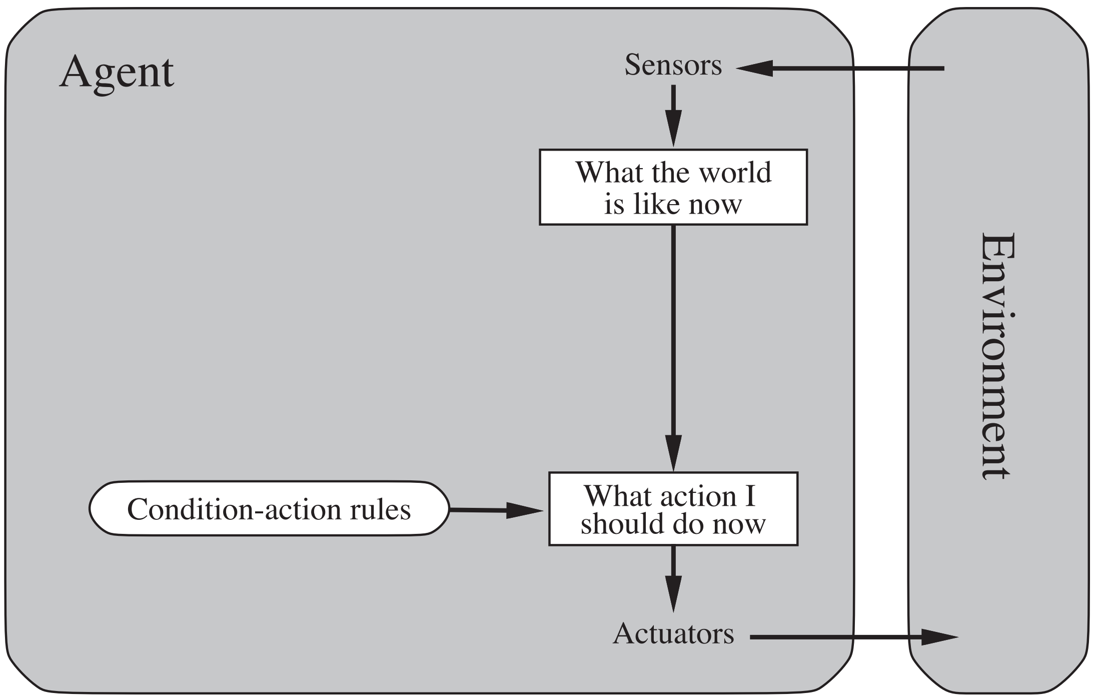
Agent Types

- Four basic types in order of increasing generality:
  - Simple reflex agent
  - Model-based reflex agent
  - Goal-based agent
  - Utility-based agent

Simple Reflex Agent

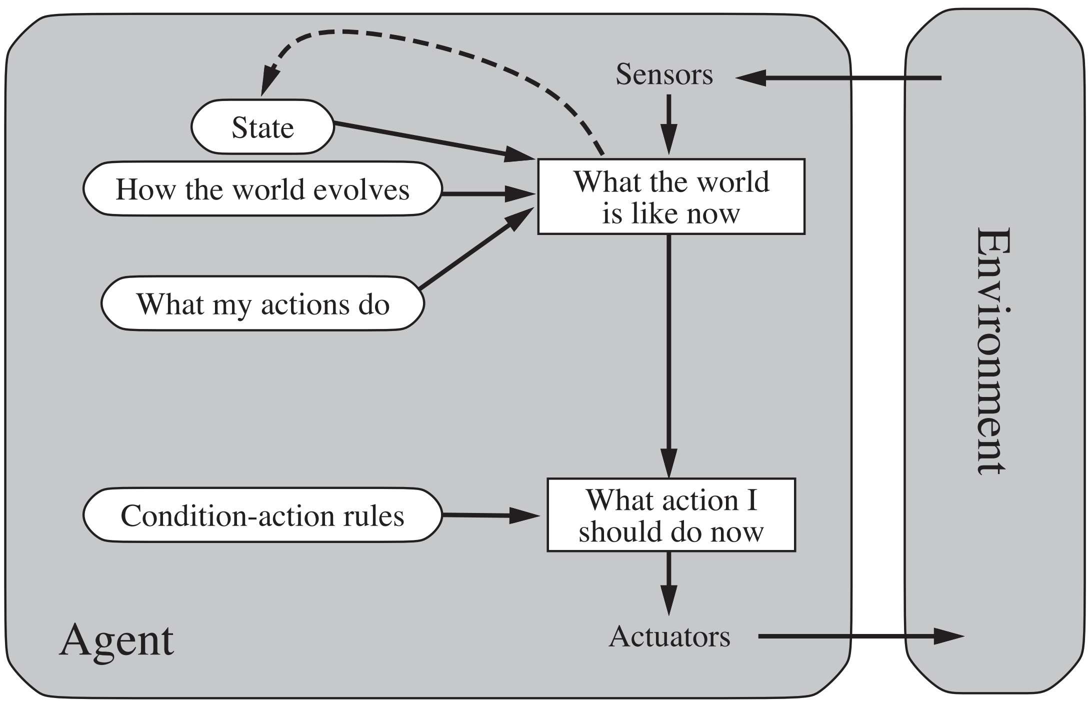
function SIMPLE-REFLEX-AGENT(*percept*) **returns** an action
  persistent: *rules*, a set of condition–action rules

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

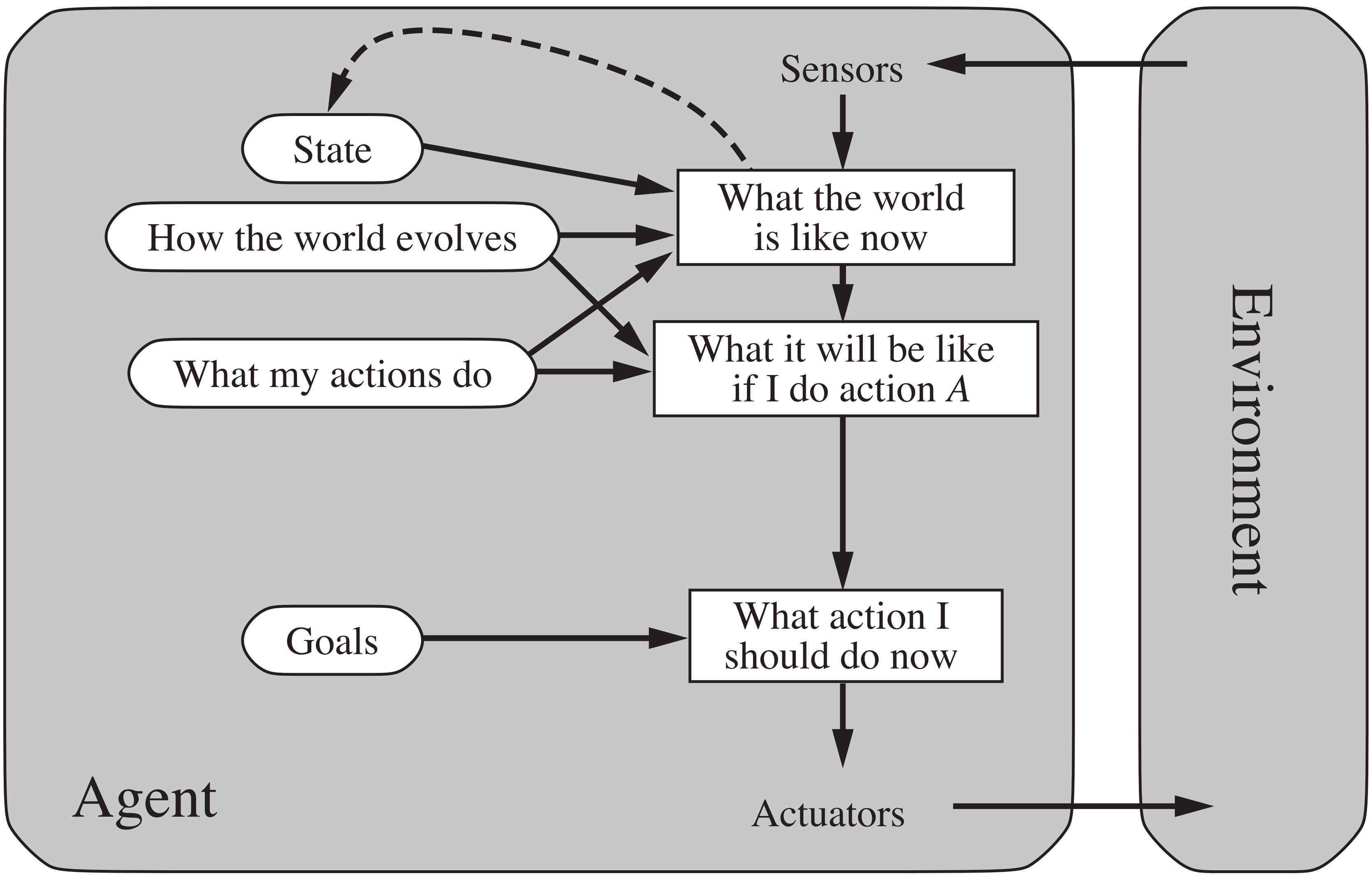


Model-Based Reflex Agent

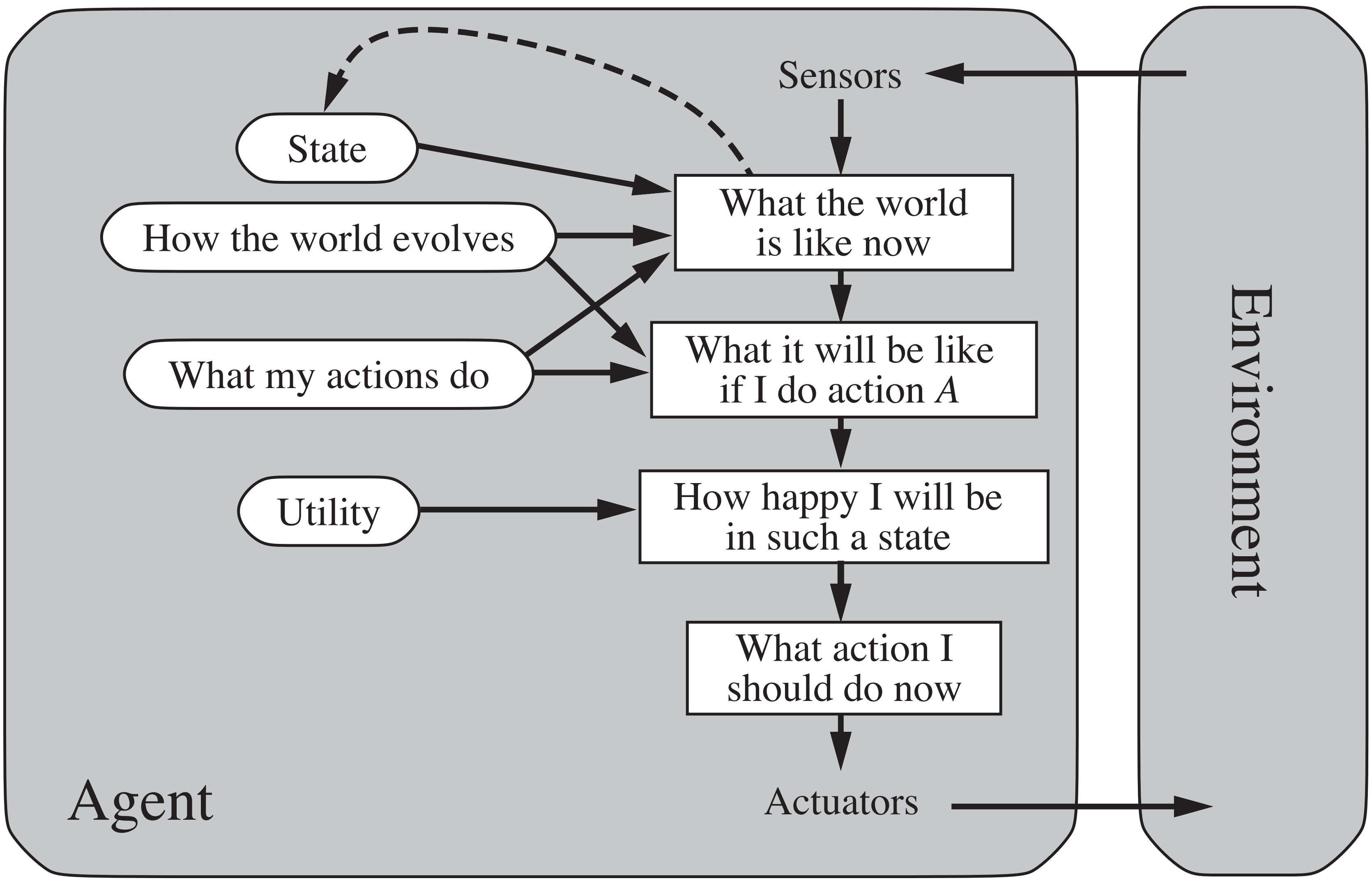

```
function MODEL-BASED-REFLEX-AGENT(percept) returns an action
  persistent: state, the agent's current conception of the world state
              model, a description of how the next state depends on current state and action
              rules, a set of condition–action rules
              action,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

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



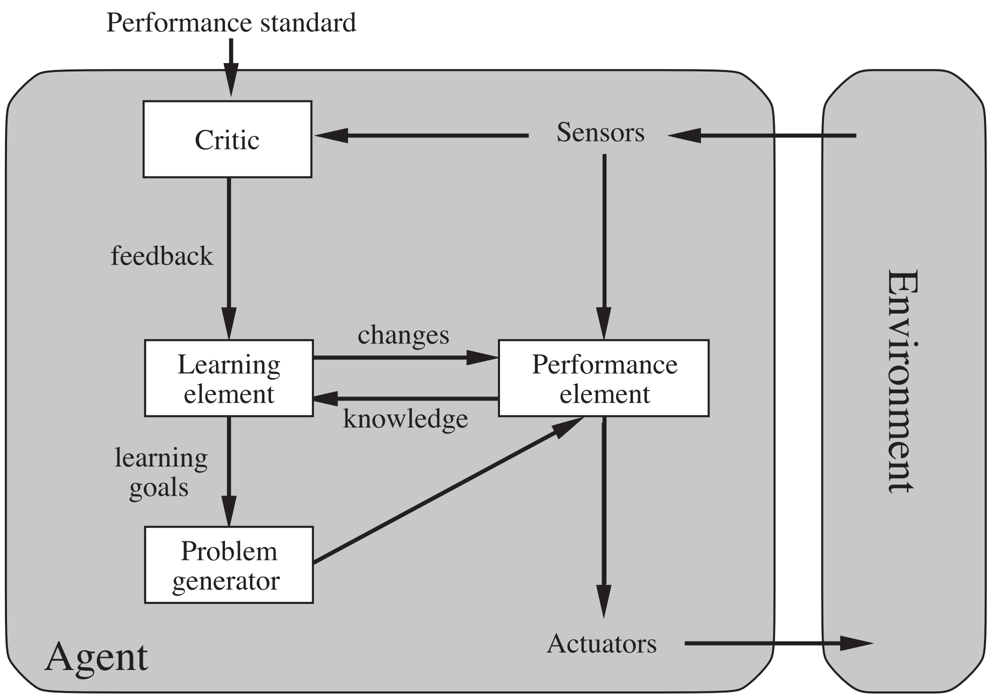
Goal-Based Agent



Utility-Based Agent



Learning Agent



- Performance element: selects the external actions
- Learning element: improves agent to perform better
- Critic: provides feedback on how well the agent is doing
- Problem generator: suggests explorative actions that will lead to new, informative (but not necessarily better) experiences

Exploitation vs. Exploration

- An agent operating in the real world must often choose between:
  - maximizing its expected utility according to its current knowledge about the world; and
  - trying to learn more about the world since this may improve its future gains.
- This problem is known as the trade-off between ***exploitation*** and ***exploration***