# Web-Mining Agents
## Cooperating Agents for Information Retrieval

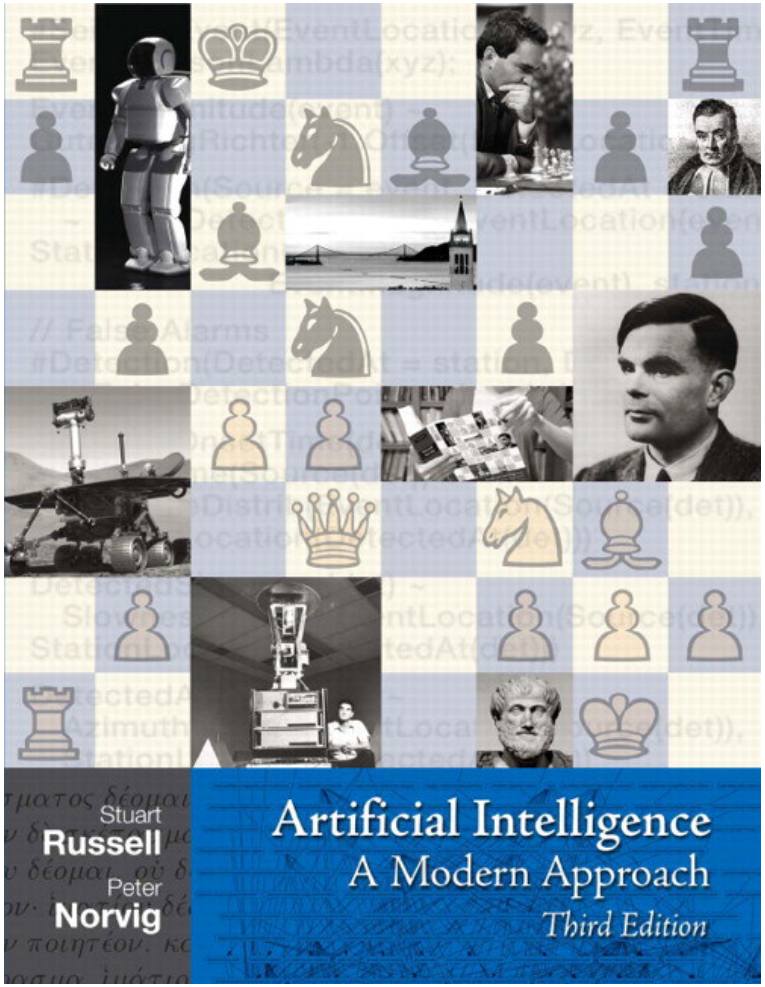
Prof. Dr. Ralf Möller
Universität zu Lübeck
Institut für Informationssysteme

Tanya Braun (Übungen)

# Organizational Issues: Assignments

- **Start**: Wed, 19.10., 2-4pm, AM S 1 (thereafter IFIS 2032), Class also Thu 2-4pm, Seminar room 2/3 (Cook/Karp) or IFIS 2032
- **Lab**: Fr. 2-4pm, Building 64, IFIS 2035 (3rd floor) (registration via Moodle right after this class)
- **Assignments** provided via Moodle after class on Thu.
- **Submission of solutions** by Wed 2pm, small kitchen IFIS (one week after provision of assignments)
- **Work on assignments** can/should be done in groups of 2 (pls. indicate name&group on submitted solution sheets)
- In **lab classes on Friday**, we discuss assignments from current week and understand solutions for assignments from previous week(s)

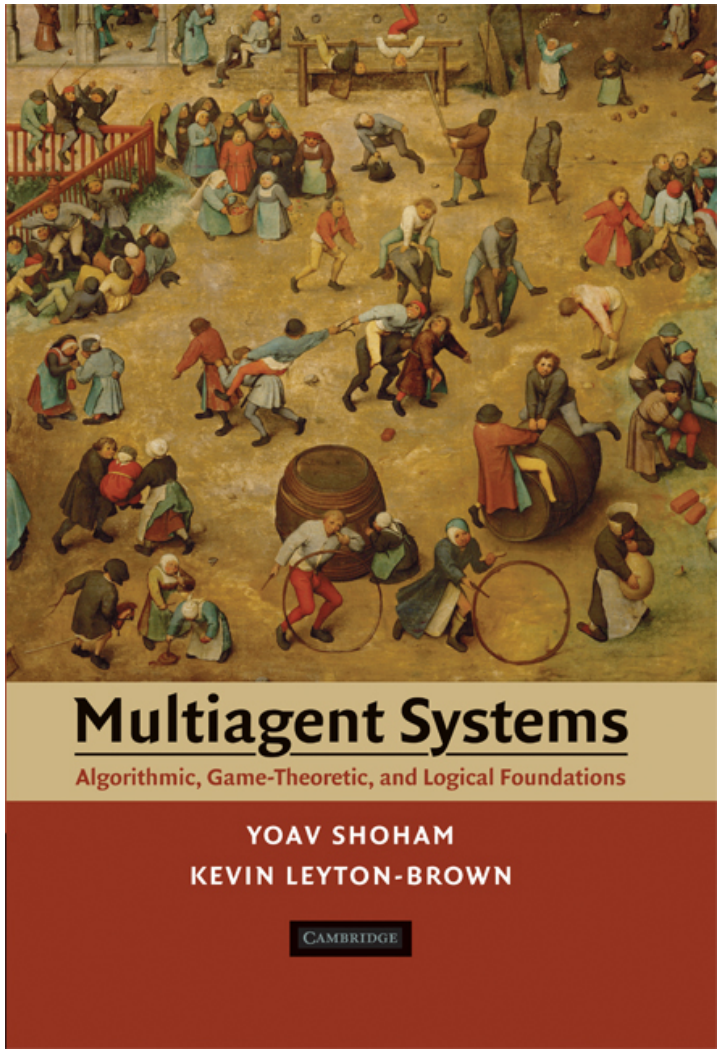
# Literature

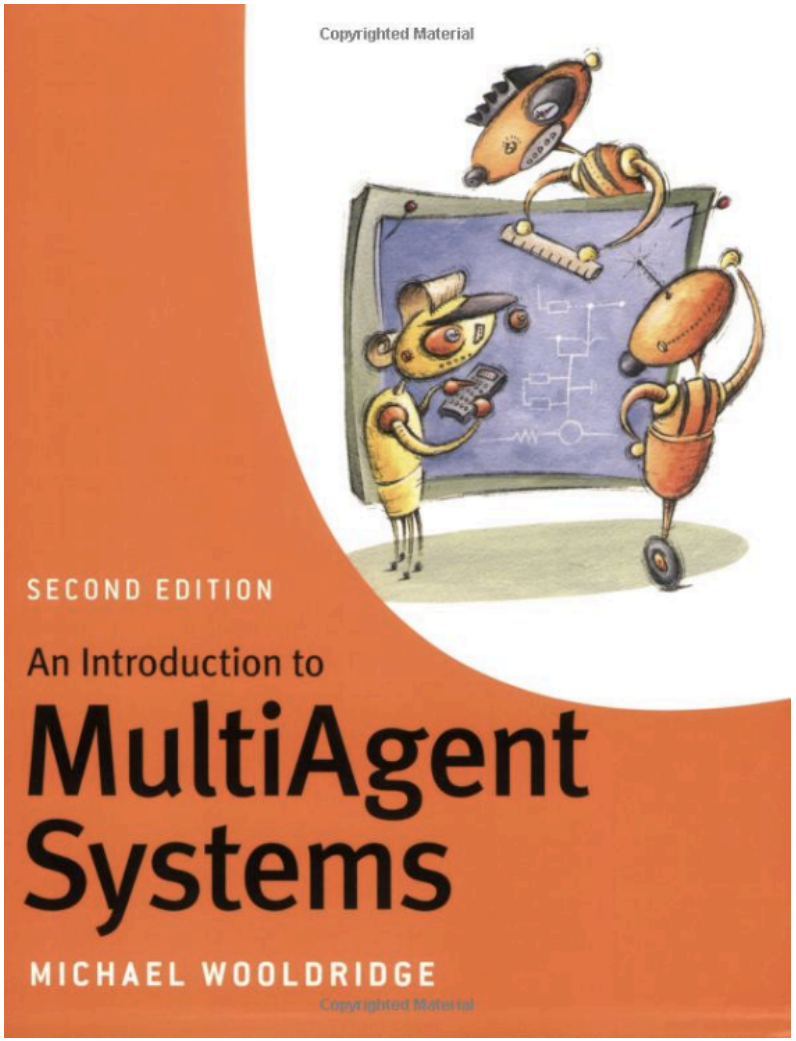



http://aima.cs.berkeley.edu

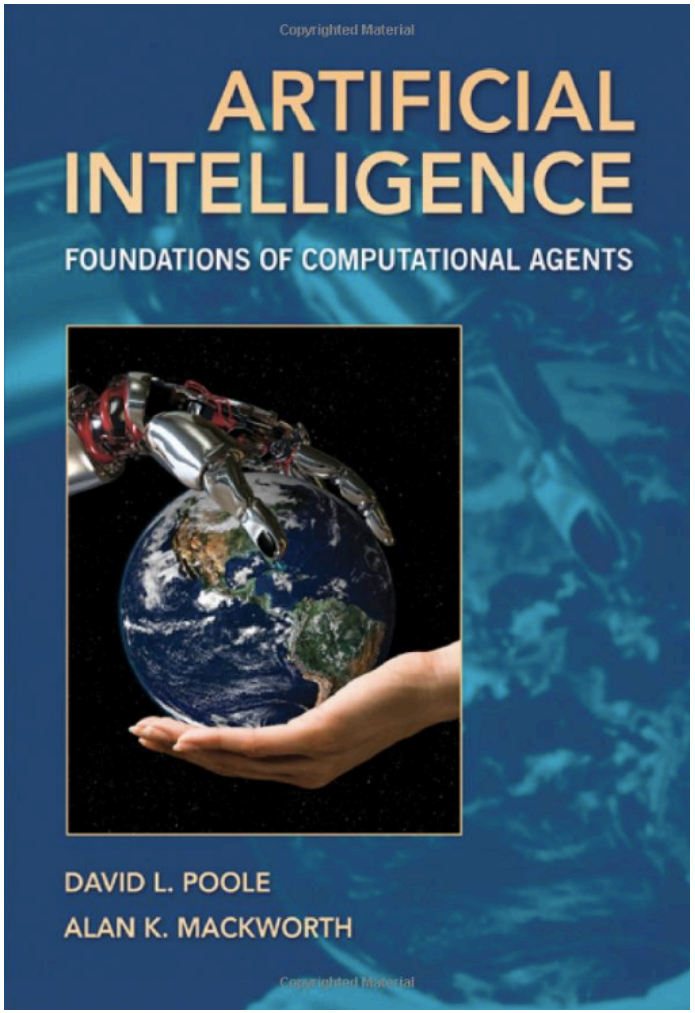
Chapters 2, 6, 13, 15- 17

# Literature


Multiagent Systems
Algorithmic, Game-Theoretic, and Logical Foundations
YOAV SHOHAM
KEVIN LEYTON-BROWN
CAMBRIDGE


Copyrighted Material
SECOND EDITION
An Introduction to
MultiAgent Systems
MICHAEL WOOLDRIDGE
Copyrighted Material

# Literature


Copyrighted Material
ARTIFICIAL INTELLIGENCE
FOUNDATIONS OF COMPUTATIONAL AGENTS
DAVID L. POOLE
ALAN K. MACKWORTH
Copyrighted Material

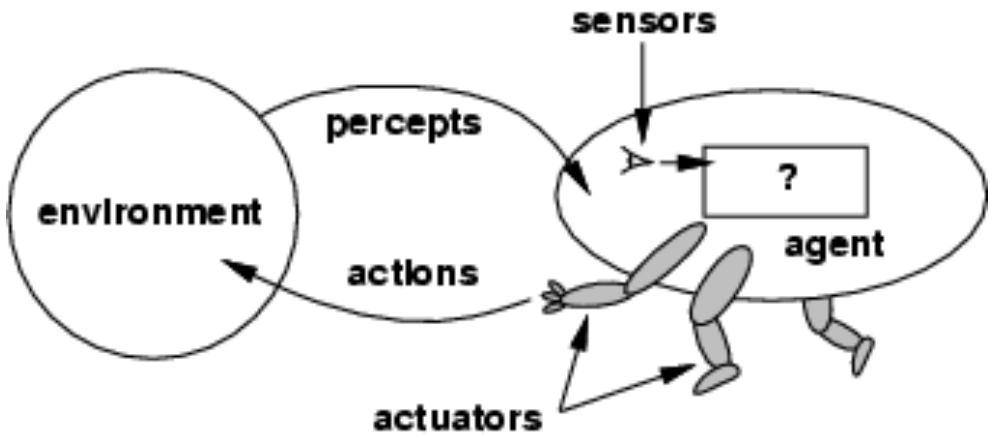
# What is an Agent?

- 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

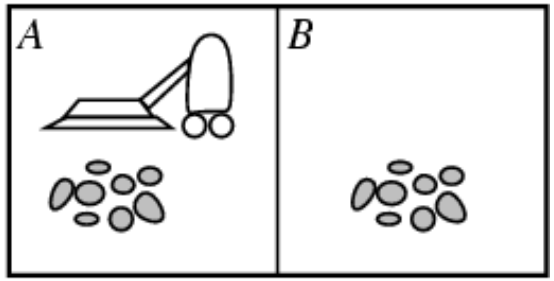



- The agent function maps from percept histories to actions:

$$[f: P^* \rightarrow A]$$

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

# Vacuum-Cleaner World




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

# A Vacuum-Cleaner Agent

| Percept sequence | Action |
|---|---|
| $[A, Clean]$ | $Right$ |
| $[A, Dirty]$ | $Suck$ |
| $[B, Clean]$ | $Left$ |
| $[B, Dirty]$ | $Suck$ |
| $[A, Clean], [A, Clean]$ | $Right$ |
| $[A, Clean], [A, Dirty]$ | $Suck$ |
| ⋮ | ⋮ |

# Capabilities

- An agent is capable of **flexible** action in some environment

- By flexible, we mean:
  - reactive
  - pro-active
  - social

# Reactivity

- If a program's environment is guaranteed to be fixed, the program need never worry about its own success or failure – program just executes blindly
  - Example of fixed environment: compiler

- The real world is not like that: things change, information is incomplete. Many (most?) interesting environments are *dynamic*

- A *reactive* system is one that maintains an ongoing interaction with its environment, and responds to changes that occur in it (in time for the response to be useful)

# Proactiveness

- Reacting to an environment is easy (e.g., stimulus → response rules)
- But we generally want agents to *do things for us*
- Hence *goal directed behavior*
- Pro-activeness = generating and attempting to achieve goals
  - Not driven solely by events
  - Taking the initiative
  - Recognizing opportunities

# Balancing Reactive and Goal-Oriented Behavior

- We want our agents to be reactive, responding to changing conditions in an appropriate (timely) fashion
- We want our agents to systematically work towards long-term goals
- These two considerations can be at odds with one another
- Designing an agent that can balance the two remains an open research problem

# Social Ability

- The real world is a *multi*-agent environment: we cannot go around attempting to achieve goals without taking others into account
- Some goals can only be achieved with the cooperation of others
- *Social ability* in agents is the ability to interact with other agents (and possibly humans) via some kind of *agent-communication language* with the goal to let other agents to make *commitments*

# 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
- Success to be measured w.r.t. an agent-local perspective
- 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 Agents

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

# Autonomous Agents

- Rationality is distinct 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)

# Other Properties

- *Mobility*: the ability of an agent to move around an electronic network, real environment
- *(Veracity*: an agent will not knowingly communicate false information)
- *Benevolence*: agents do not have conflicting goals, and every agent will therefore always try to do what is asked
- *Learning/adaption*: agents improve performance over time

# Agents and Objects

- Are agents just objects by another name?
- Object:
  - Encapsulates some state
  - Interacts synchronously with other objects
    - communicates via message passing
    - call methods / generic functions
  - Has methods, corresponding to operations that may be performed on this state

# Agents and Objects

- Main differences:
  - *Agents are autonomous*:
    agents embody stronger notion of autonomy than objects, and in particular, they decide for themselves whether or not to perform an action on request from another agent
  - *Agents are* capable of *flexible* (reactive, pro-active, social) behavior, and the standard object model has nothing to say about such types of behavior
  - *Agents are* inherently *multi-threaded*, in that each agent is assumed to have at least one thread of active control

# Main Features

- **P**erformance measure
- **E**nvironment
- **A**ctuators
- **S**ensors

Must first specify the setting for intelligent agent design

# 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

# PEAS

- Agent: Medical diagnosis system
  - Performance measure: Healthy patient, minimize costs, avoid 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 relative to number of parts on the ground
  - Environment: Area with parts, bins
  - Actuators: Jointed arm and hand
  - Sensors: Camera, joint angle sensors

# Environment Types

- Fully observable (vs. partially observable)
  - Agent's sensors give it access to complete state of the environment at each point in time.
- Deterministic (vs. stochastic):
  - Next state of the environment completely determined by the current state and executed action
  - If the environment is deterministic except for the actions of other agents, then the environment is strategic
- Episodic (vs. sequential):
  - Agent's experience divided into atomic "episodes"
    - Episode consists of the agent perceiving and then performing a single action
  - Choice of action in each episode depends only on the episode itself.

# Environment Types

- Static (vs. dynamic):
  - Environment unchanged while agent is deliberating
  - Environment is semidynamic if the environment itself does not change with the passage of time but the agent's performance score does

- Discrete (vs. continuous):
  - There is a limited number of distinct, clearly defined percepts and actions

- Single agent (vs. multiagent):
  - An agent operating by itself in an environment

# Environment Types

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

# 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 | | | |
| Discrete | | | |
| Single agent | | | |

# 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 | | | |
| Single agent | | | |

# 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 |

- Environment type largely determines the agent design
- Real world is (of course) partially observable, stochastic, sequential, dynamic, continuous, multi-agent

# 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

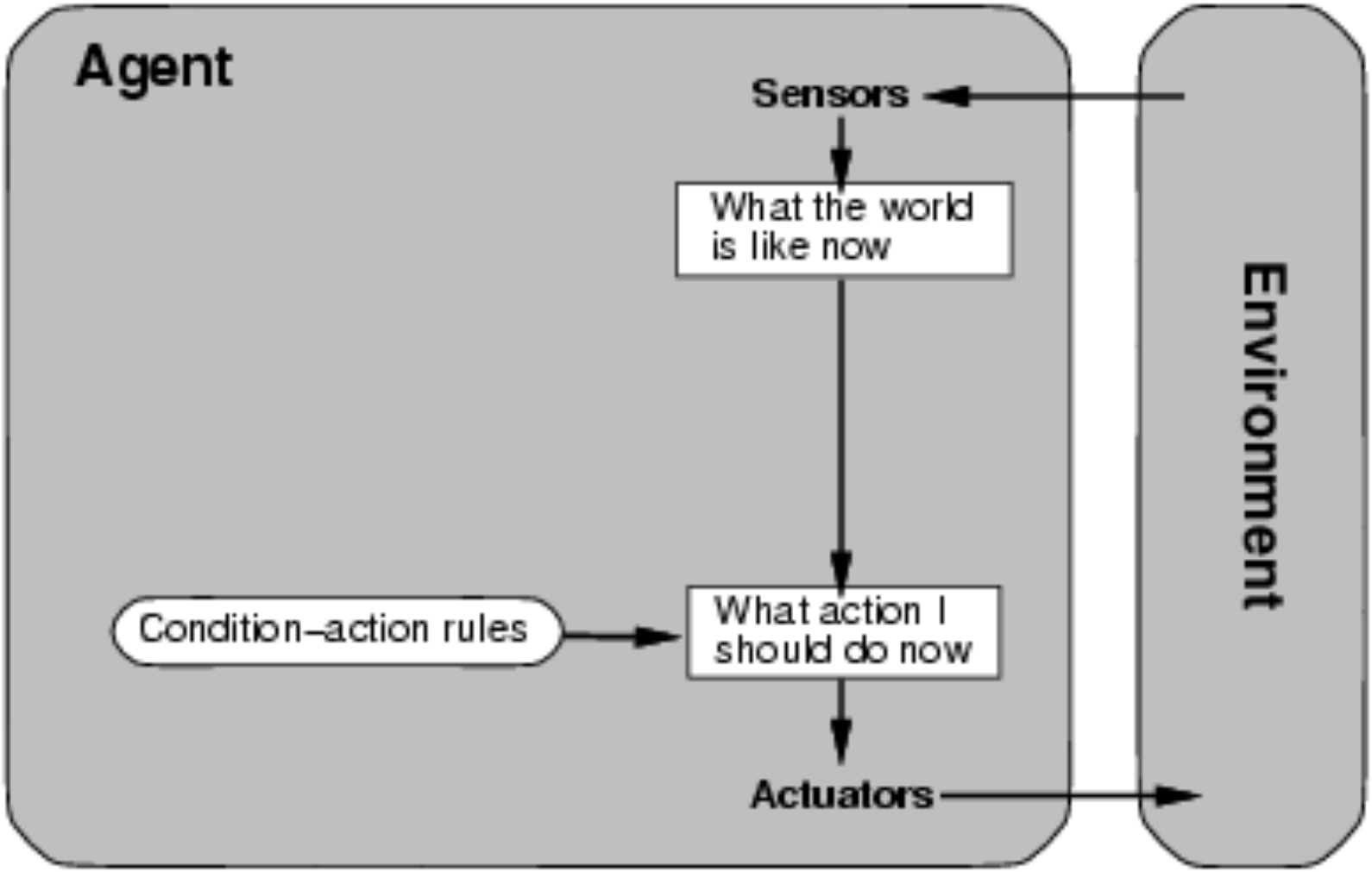
# Table-Lookup Agent

- Drawbacks:
  - Huge table
  - Take a long time to build the table
  - No autonomy
  - Even with learning, need a long time to learn the table entries

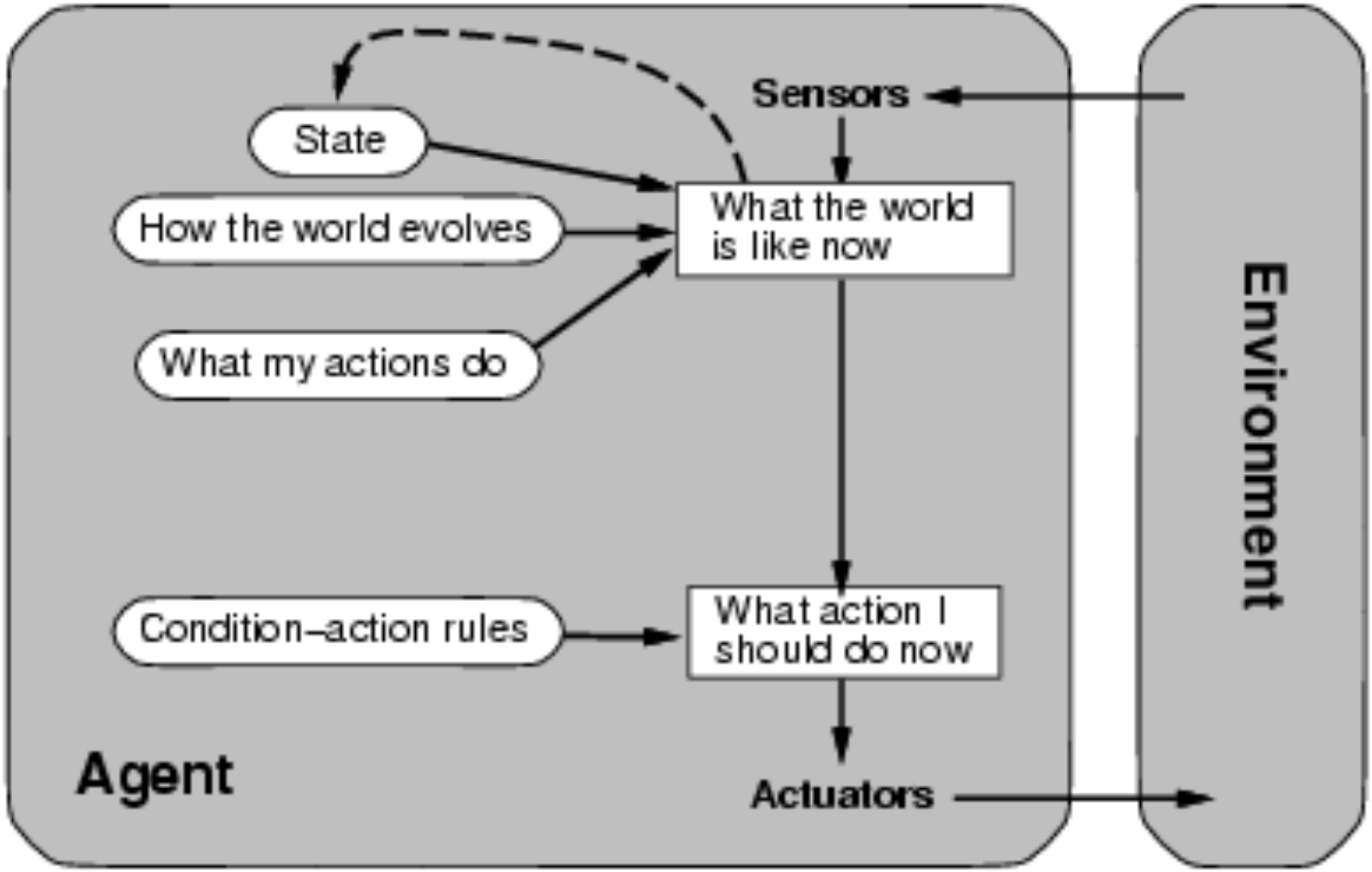
# Agent Types

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

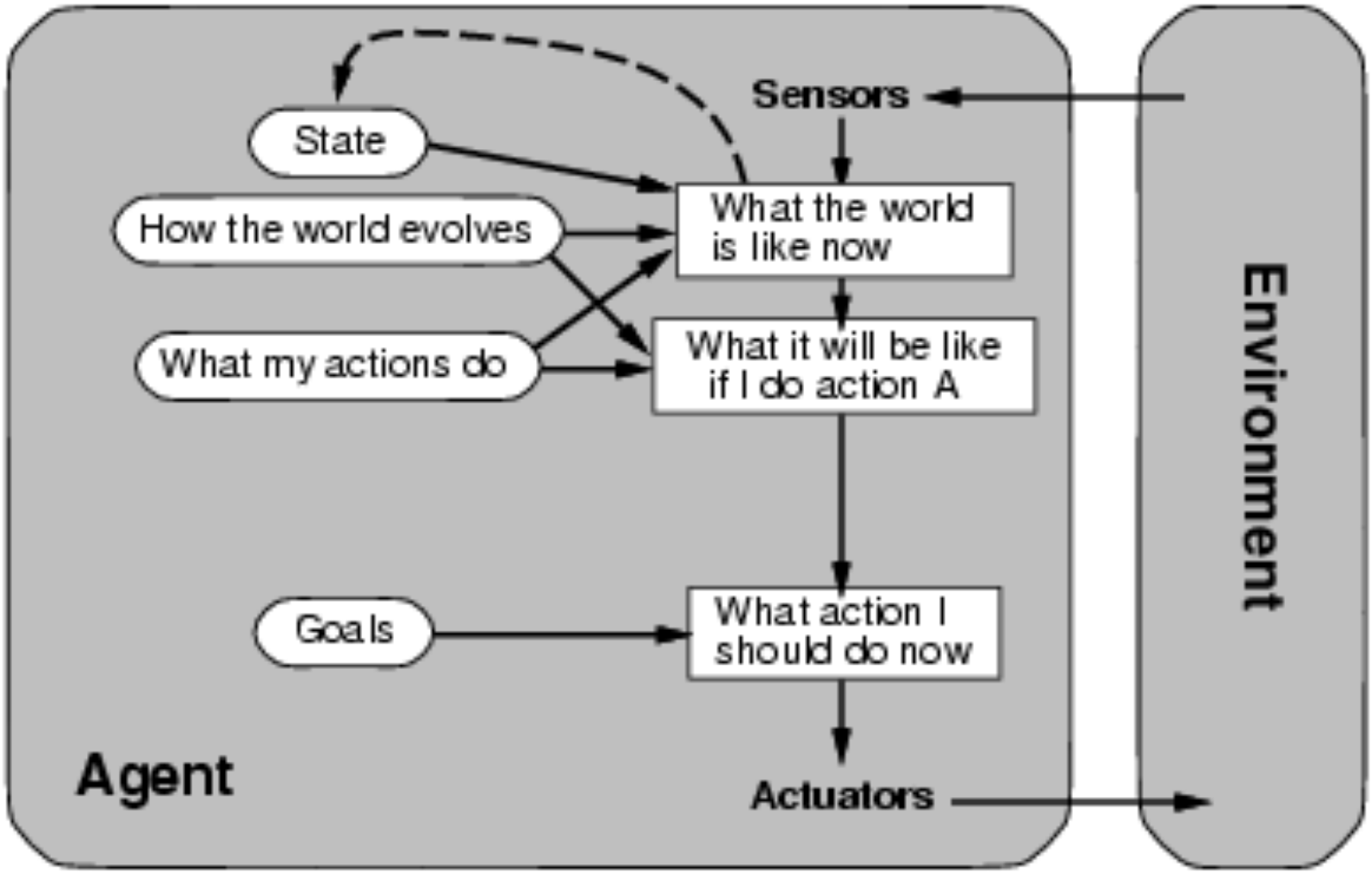
# Simple Reflex Agents



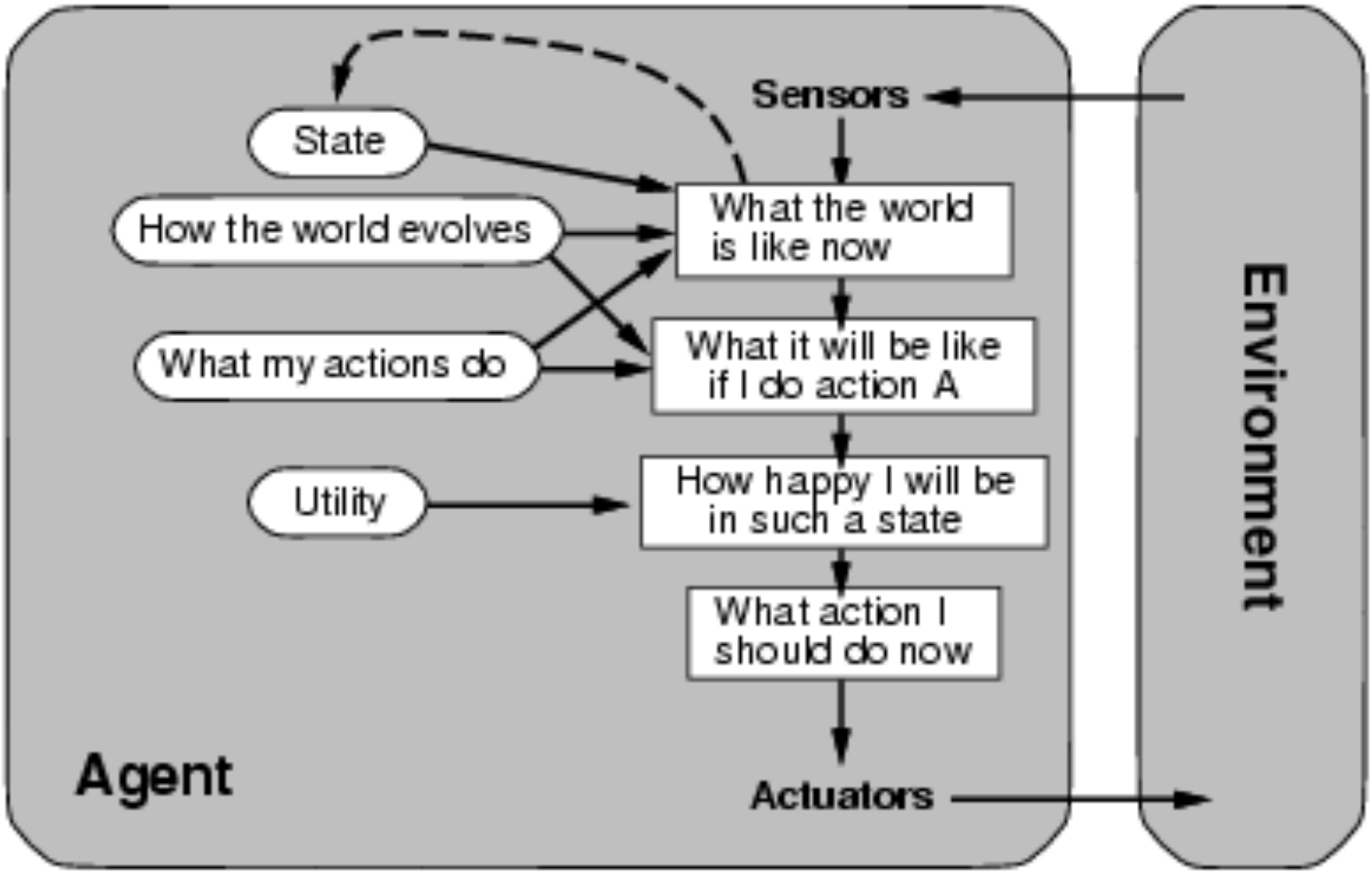
# Model-Based Reflex Agents



# Goal-Based Agents



# Utility-Based Agents



# Learning Agents