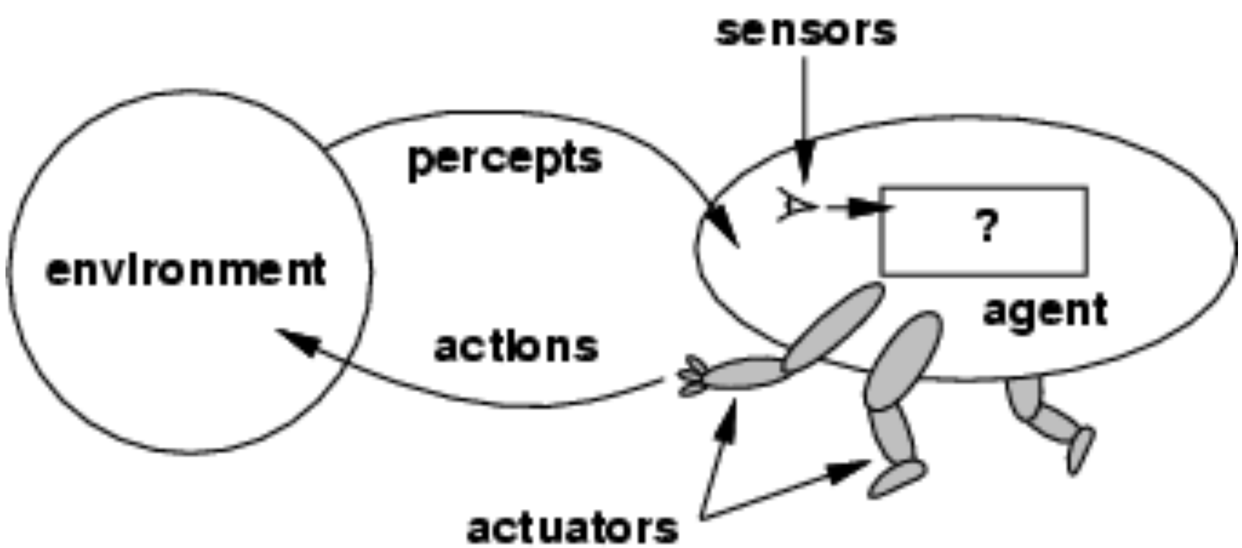
Intelligent Agents and Architectures



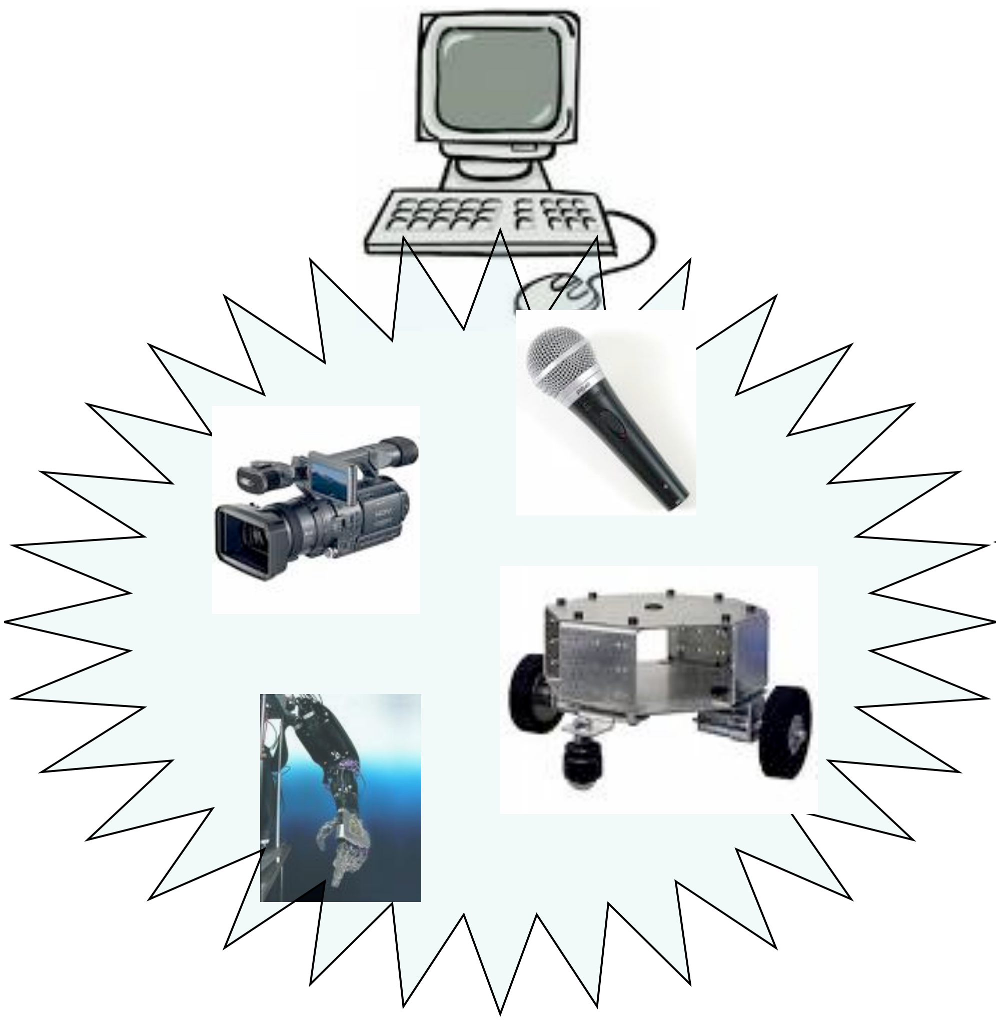
Learning Objectives

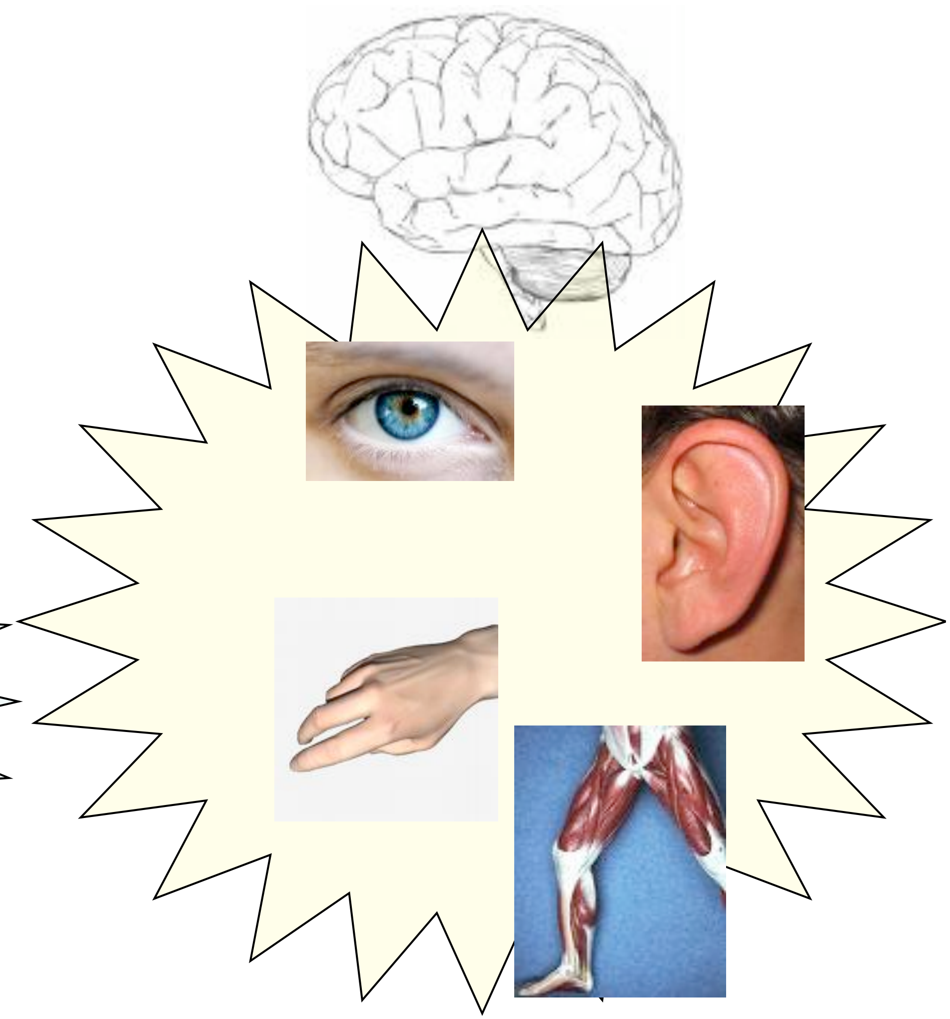
- review terminology of intelligent agents
- understand concept of autonomy
- compare different agent architectures

Readings for this class

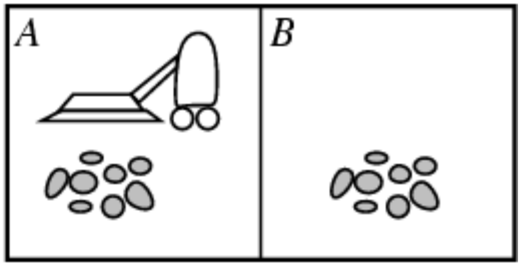
- Chapter 5-5.4

What is an agent?





Vacuum-cleaner world



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

The Lowly Dung Beetle

After digging its nest and laying its eggs, the dung beetle fetches a ball of dung from a nearby en route heap to plug the entrance; if the ball of dung is removed from its grasp, the beetle continues on and pantomimes plugging the nest with the nonexistent dung ball.

from Hanski & Cambefort, 1991

The Sphex Wasp

The female will dig a burrow, go out and sting a caterpillar and drag it to the burrow, enter the burrow again to check all is well, drag the caterpillar inside, and lay its eggs… but if an entomologist moves the caterpillar a few inches away while the sphex is doing the check, it will revert back to the "drag" step of its plan, and will continue the plan without modification, even after dozens of caterpillar-moving interventions.

Rational Agents

- rational agent relies on its own percepts (sensors) and experience (learning), not just prior "hardwired" knowledge of its designer

  ➔"autonomous"

Automatic vs. Autonomous

Automatic *means that a system will do exactly as programmed, it has no choice.*

Autonomous *means that a system has a choice to make free of outside influence, i.e., an autonomous system has free will.*

Brian T Clough, "Metrics, Schmetrics! How The Heck Do You Determine A UAV's Autonomy Anyway"

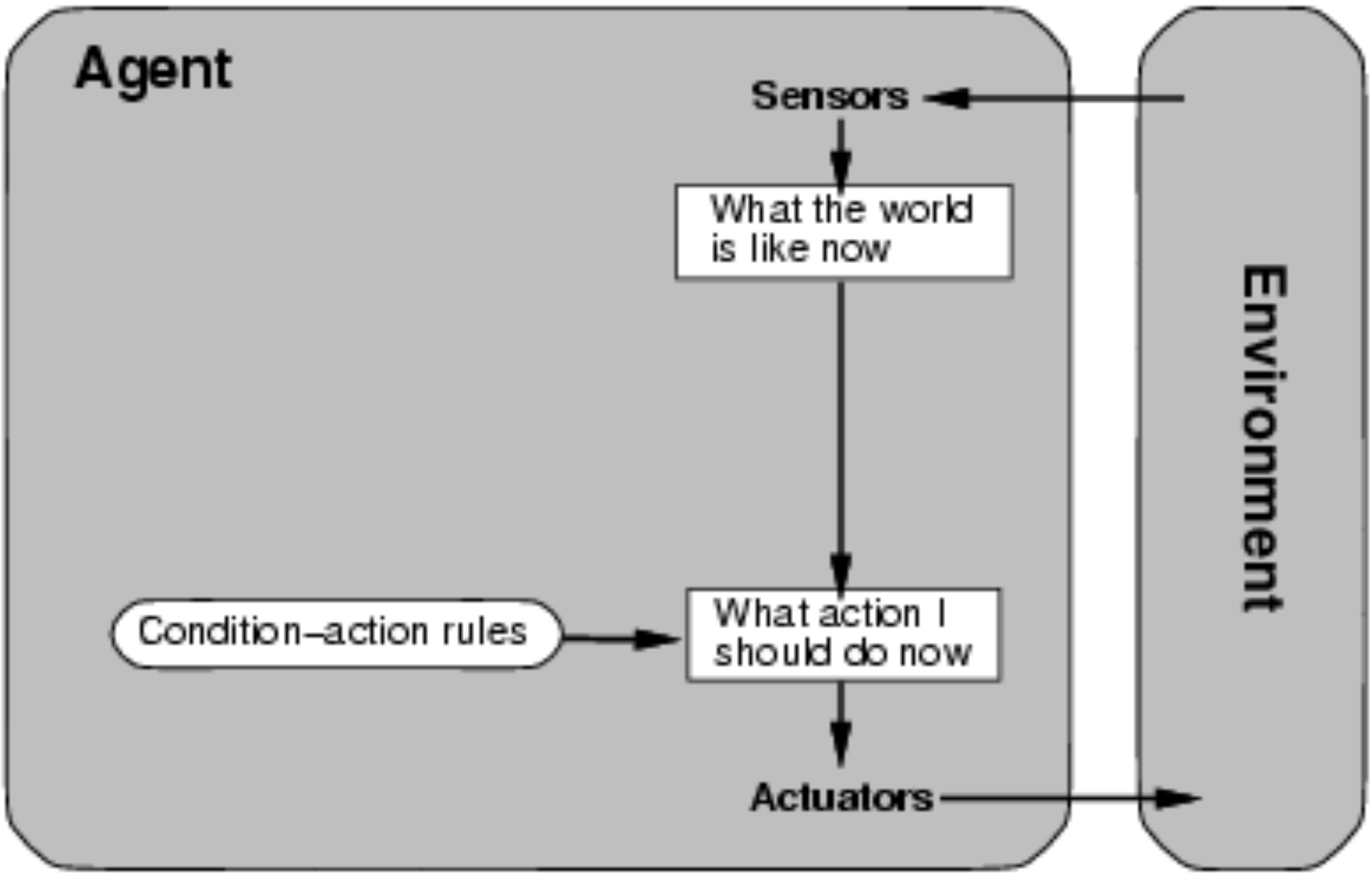
Agent functions and programs

- an agent is completely specified by the *agent function* mapping percept sequences to actions
- we are interested in an efficient implementation of a *rational* agent function

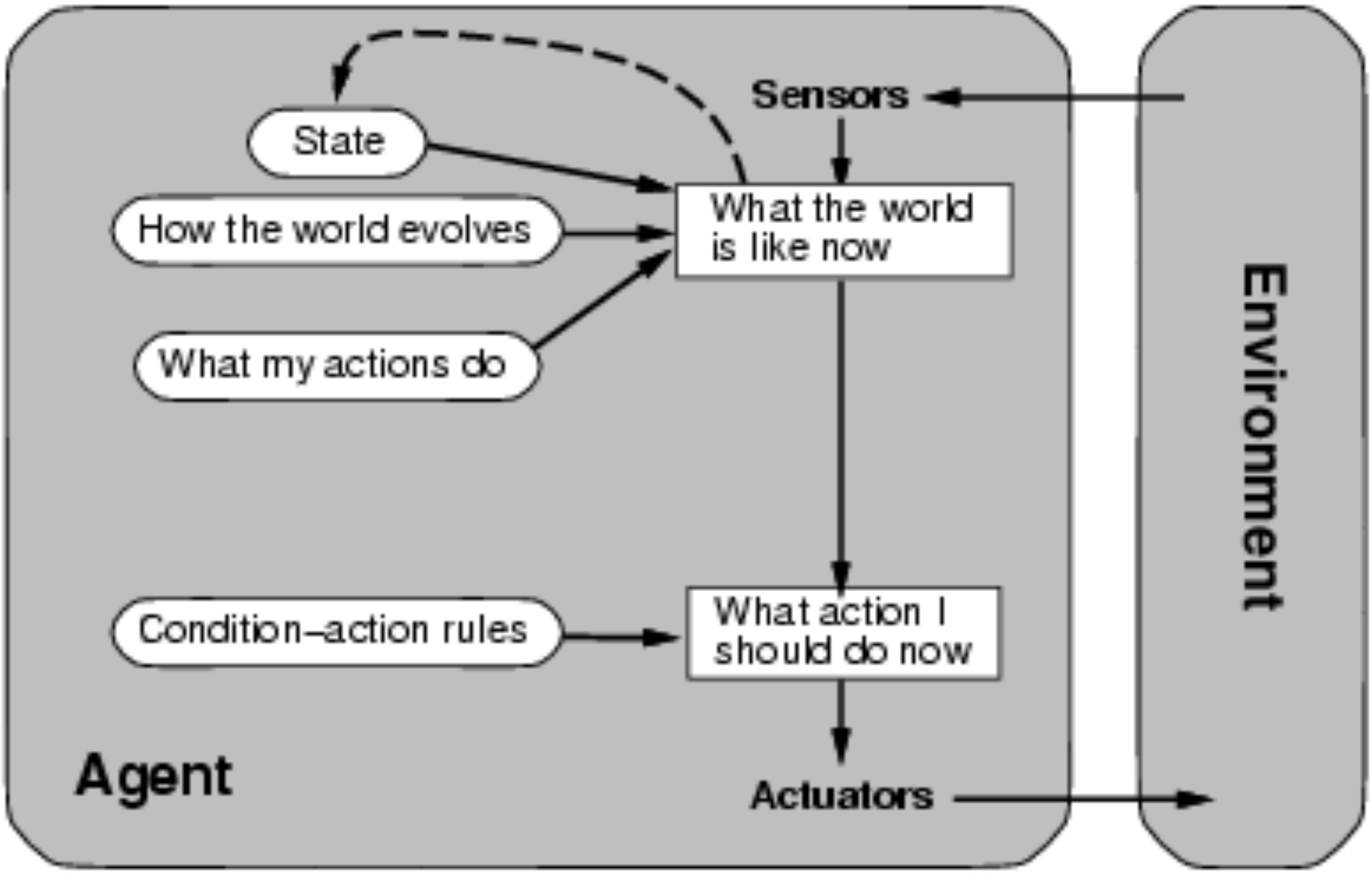
Table-lookup agent

- could construct a look-up table with appropriate action to take for any percept sequence

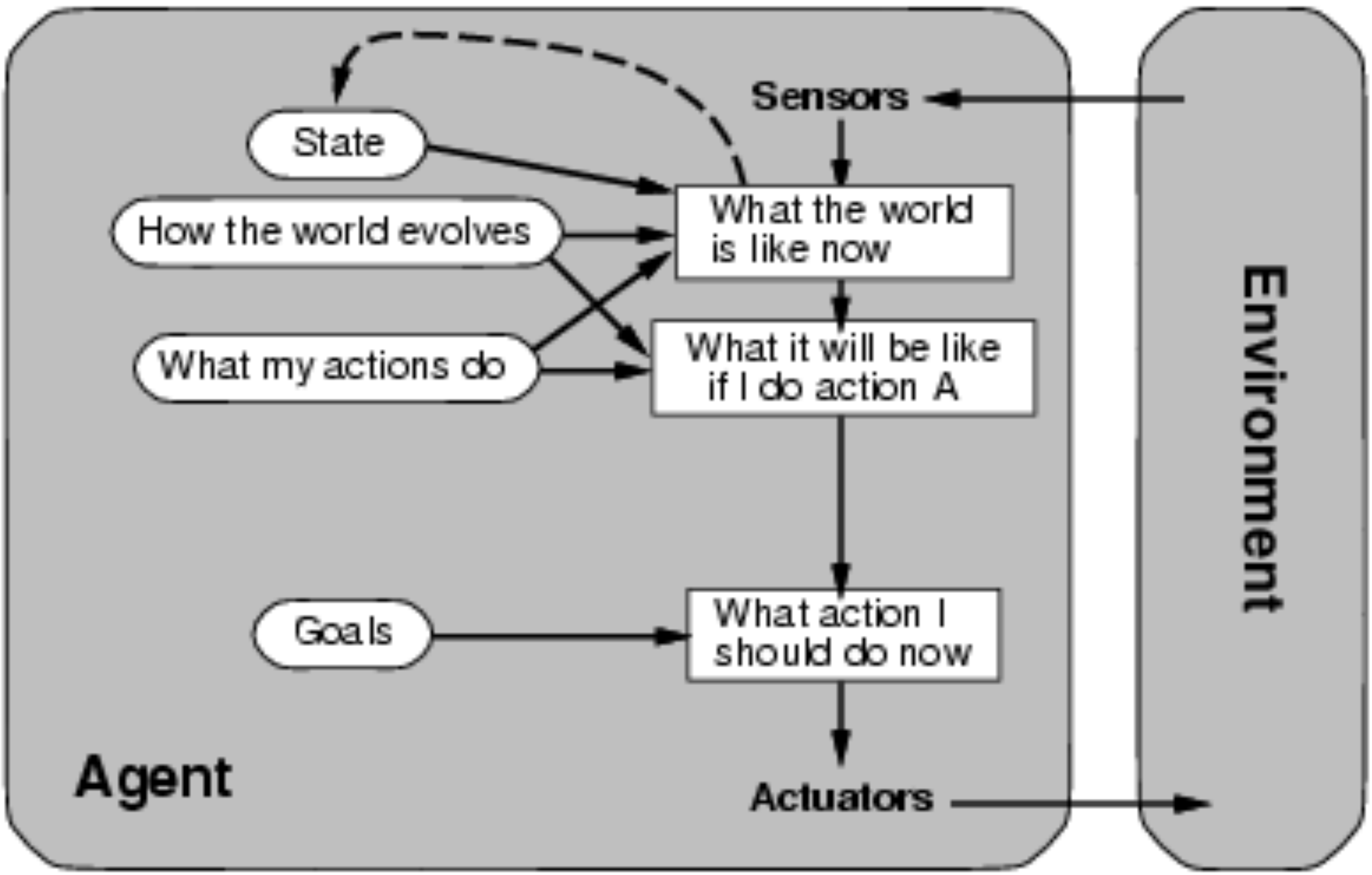
Simple reflex agents



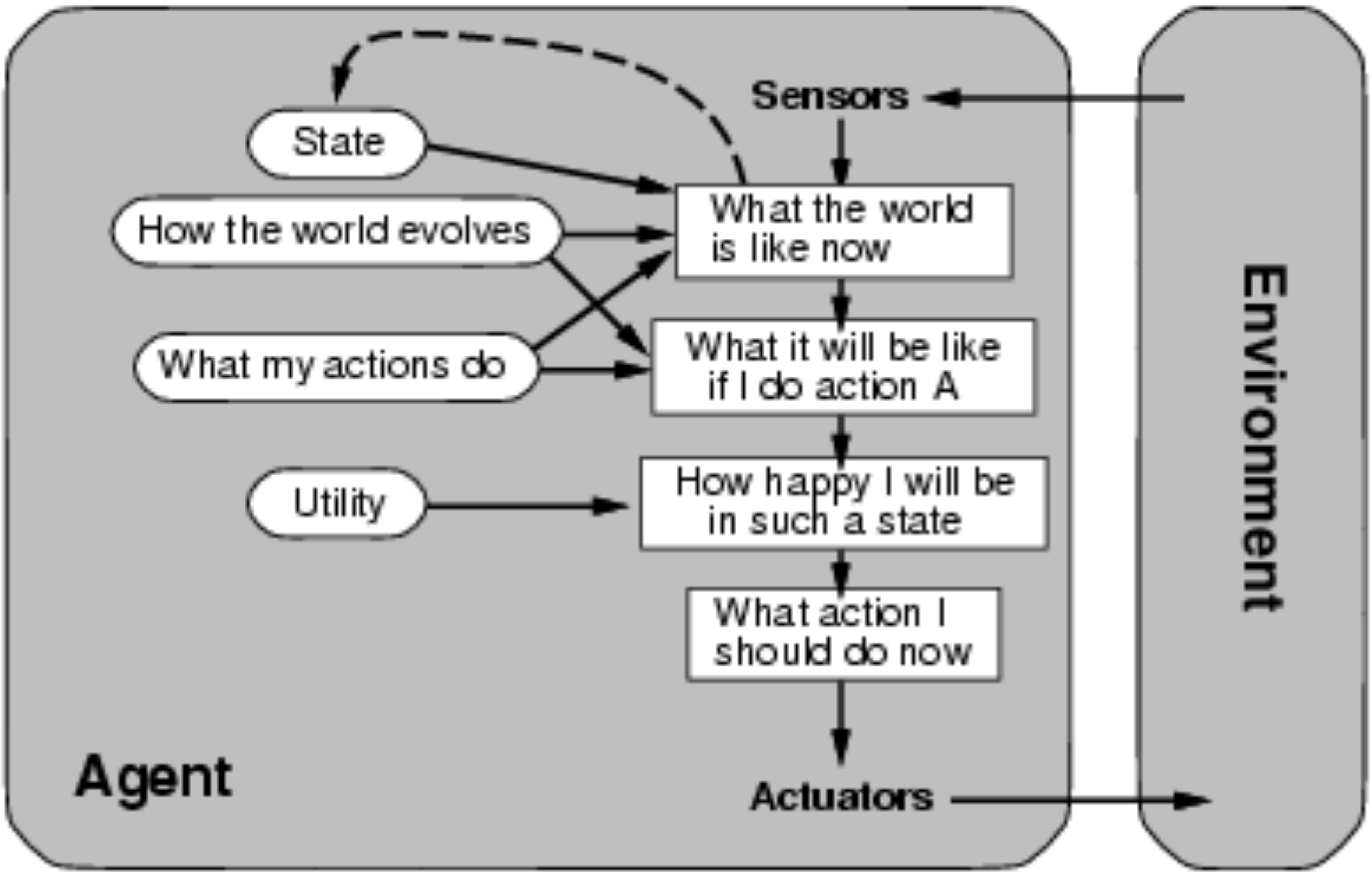
Model-based reflex agents



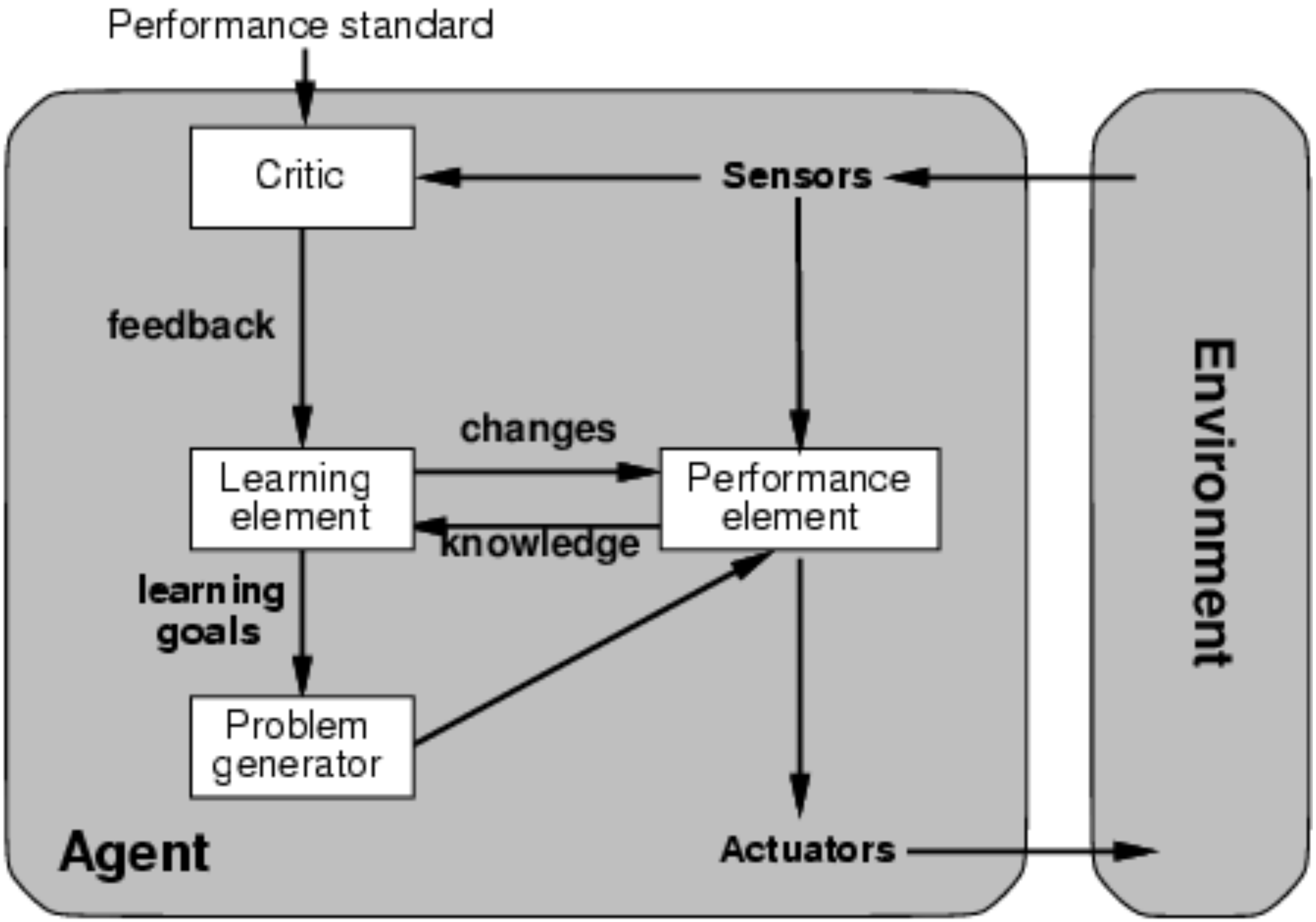
Goal-based agents



Utility-based agents



Learning agents



Task Environments: PEAS description

- **p**erformance measure
- **e**nvironment
- **a**ctuators
- **s**ensors

PEAS example: chess player

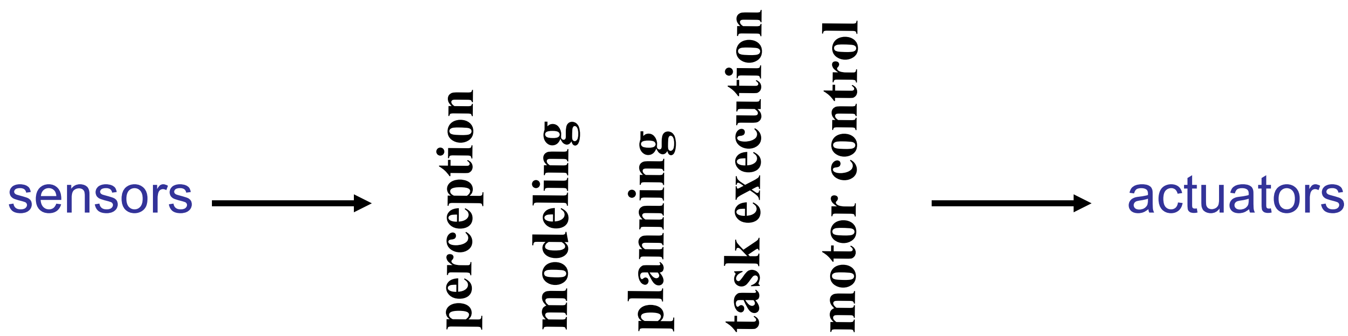
performance measure	2 pts/win, 1/draw, 0/loss
environment	chess board, pieces, rules, move history
actuators	move piece, resign
sensors	observe board position, time

Task environments

- fully vs. partially observable
- single-agent vs. multi-agent
- deterministic vs. stochastic
- episodic vs. sequential
- static vs. dynamic
- discrete vs. continuous
- known vs. unknown

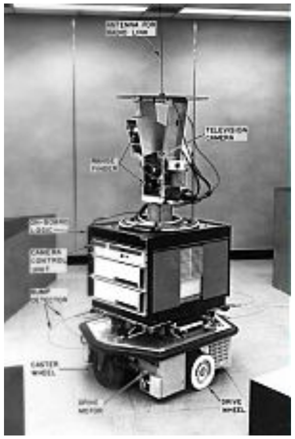
Deliberative Architectures

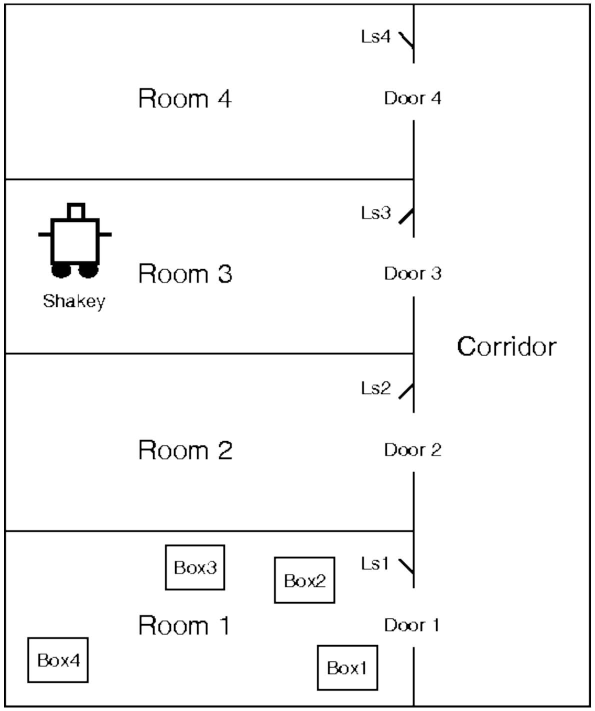
- SMPA: sense, model, plan, act
- use internal, symbolic representation (model) of world for reasoning/planning



Shakey (1966-1972)

- PLANEX: accepts goals, maintains world state, calls planner, executes best plan
- STRIPS: propositional logic planning system
- ILA: intermediate-level actions
- LLA: low-level actions





Strengths of Deliberation

- allows agent to look-ahead at possible outcomes of action (search) without actually performing action
- powerful learning (adaptation, compilation) schemes can be incorporated

Weaknesses

- cost of generating model, inaccuracies
- cost of reasoning about low-level actions
- too expensive for real-time behaviour
- the “microworlds” problem
  - AI programs operated in small domains
  - relied on very simple world models
  - couldn't scale to real world scenarios

Which robot is more capable?

Brooks' criticism

- intelligence does not require:
  - explicit representation
  - abstract, symbolic reasoning

"the world is its own best model"

Brooks' key ideas

- **Situatedness**: agents must behave successfully in their environment, directly interacting with the world as opposed to modeling it
- **Embodiment**: real interaction with the physical world is critical to resolve the symbol grounding problem
- **Intelligence**: development of perception and mobility drove evolutionary progress; intelligence did not emerge top-down
- **Emergence**: interaction of individual modules, within a complex environment, gives appearance of intelligence

Reactive Architectures

- behaviour-based robotics
- act using stimulus/response behaviour
- no centralized control
- example: subsumption architecture
  - layered task decomposition
  - tight connection of perception to action

Subsumption Architecture

sensors →

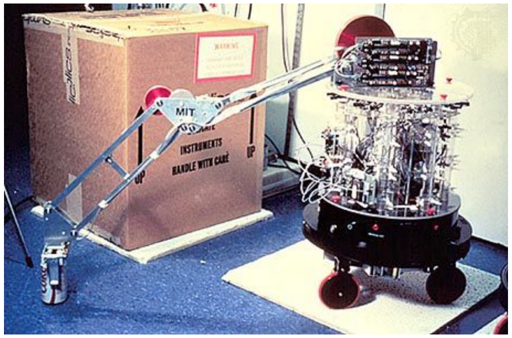
reason about objects
plan changes to world
identify objects
monitor changes
build maps
explore
wander
avoid objects

→ actuators

Strengths

- competence can be achieved without explicit reasoning or rule following [Dreyfus, 1972]
- internal models are expensive and inaccurate so best to avoid them
- complex behaviour emerges through interaction of simple behaviours

Homework



- read Ch. 6-6.2
- design a finite state machine controller for a robot with the following sensors and effectors

Sensors	**Effectors**
Laser rangefinder	three-wheel base
Compass	2-joint arm
IR beam-break (between fingers)	gripper
IR proximity sensors (around base)	
Contact sensor (on hand)	

- goal: roam 2nd floor of Trottier building, collect empty soda cans and bring them to the recycling bin