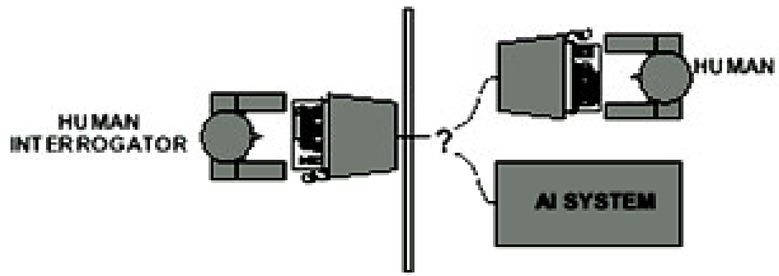
Last Time: Acting Humanly: The Full Turing Test

- Alan Turing's 1950 article *Computing Machinery and Intelligence* discussed conditions for considering a machine to be intelligent
  - "Can machines think?" ←→ "Can machines behave intelligently?"
  - The Turing test (The Imitation Game): Operational definition of intelligence



- Computer needs to possess: Natural language processing, Knowledge representation, Automated reasoning, and Machine learning
- Problem: 1) Turing test is not reproducible, constructive, and amenable to mathematic analysis. 2) What about physical interaction with interrogator and environment?
- Total Turing Test: Requires physical interaction and needs perception and actuation.

Last time: The Turing Test

http://www.ai.mit.edu/projects/infolab/

This time: Outline

- Intelligent Agents (IA)
- Environment types
- IA Behavior
- IA Structure
- IA Types
- Chapter 2 of AIMA covers this material – the terminology is slightly different, but should be easy enough to map to what we cover here

What is an (Intelligent) Agent?

- An over-used, over-loaded, and misused term.
- Anything that can be *viewed as* **perceiving** its **environment** through **sensors** and **acting** upon that environment through its **actuators or effectors** to maximize progress towards its **goals**.

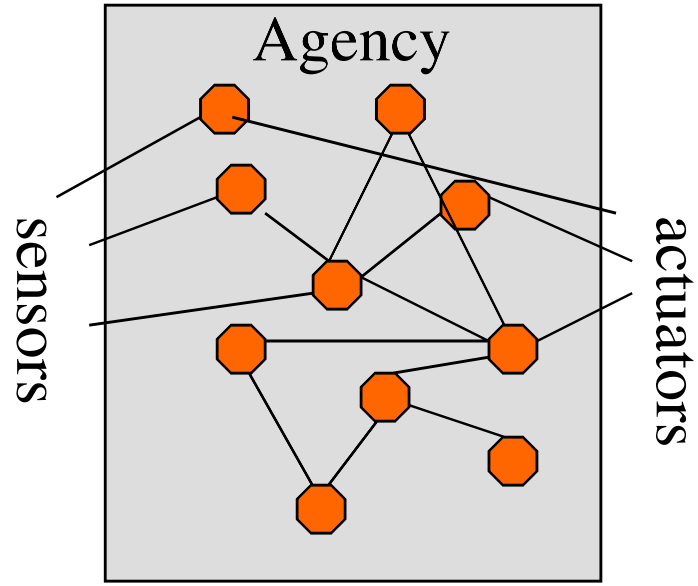
What is an (Intelligent) Agent?

- **PAGE** (Percepts, Actions, Goals, Environment)

- Task-specific & specialized: well-defined goals and environment
- The notion of an agent is meant to be a tool for analyzing systems,
  - It is not a different hardware or new programming languages

Intelligent Agents and Artificial Intelligence

- **Example:** Human mind as network of thousands or millions of agents working in parallel. To produce real artificial intelligence, this school holds, we should build computer systems that also contain many agents and systems for arbitrating among the agents' competing results.



- Distributed decision-making and control

- Challenges:
  - Action selection: What next action to choose
  - Conflict resolution

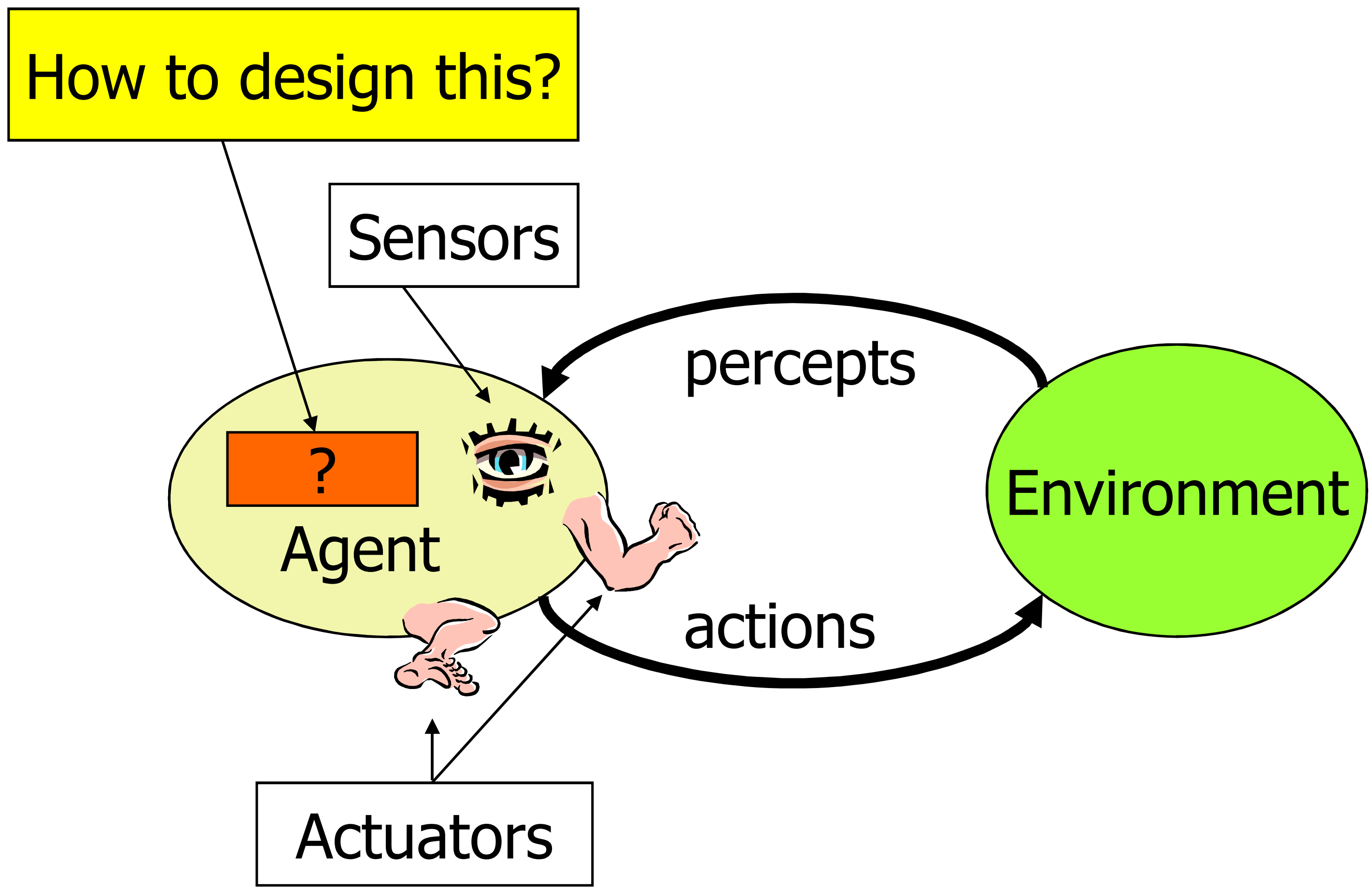
Agent Types

We can split agent research into two main strands:

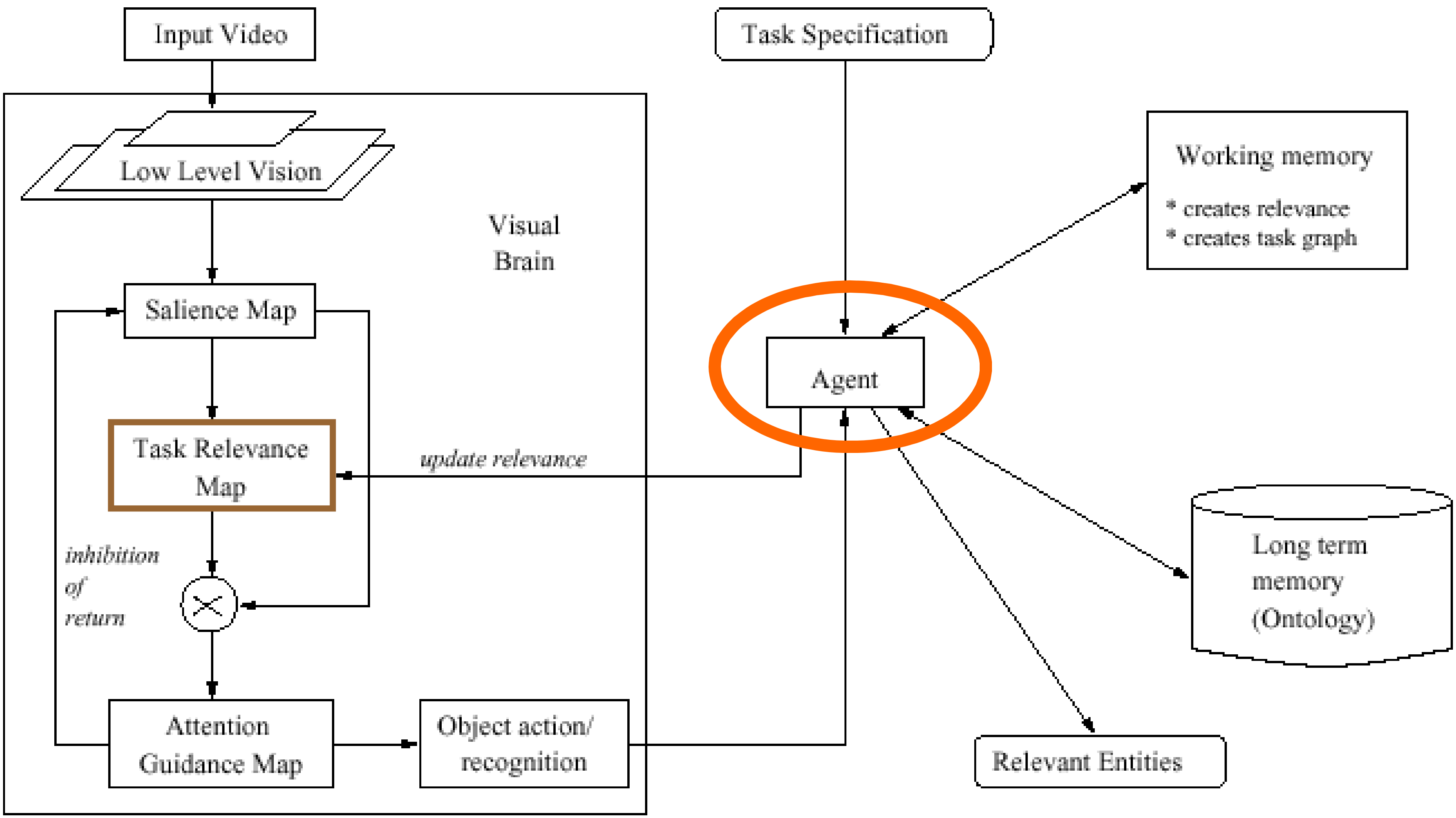
- Distributed Artificial Intelligence (DAI) –
  Multi-Agent Systems (MAS) (1980 – 1990)

- Much broader notion of "agent" (1990's – present)
  - interface, reactive, mobile, information

Rational Agents



The Beobot example: http://ilab.usc.edu/beobots/



A Windshield Wiper Agent

How do we design a agent that can wipe the windshields when needed?

- Goals?
- Percepts?
- Sensors?
- Actuators?
- Actions?
- Environment?

A Windshield Wiper Agent (Cont'd)

- Goals: Keep windshields clean & maintain visibility
- Percepts: Raining, Dirty
- Sensors: Camera (moist sensor)
- Actuators: Wipers (left, right, back)
- Actions: Off, Slow, Medium, Fast
- Environment: Inner city, freeways, highways, weather …

Towards Autonomous Vehicles

http://iLab.usc.edu

http://beobots.org

Interacting Agents

Collision Avoidance Agent (CAA)

- Goals: Avoid running into obstacles
- Percepts ?
- Sensors?
- Actuators ?
- Actions ?
- Environment: Freeway

Lane Keeping Agent (LKA)

- Goals: Stay in current lane
- Percepts ?
- Sensors?
- Actuators ?
- Actions ?
- Environment: Freeway

Interacting Agents

Collision Avoidance Agent (CAA)	
• Goals:	Avoid running into obstacles
• Percepts:	Obstacle distance, velocity, trajectory
• Sensors:	Vision, proximity sensing
• Actuators:	Steering Wheel, Accelerator, Brakes, Horn, Headlights
• Actions:	Steer, speed up, brake, blow horn, signal (headlights)
• Environment:	Freeway
Lane Keeping Agent (LKA)	
• Goals:	Stay in current lane
• Percepts:	Lane center, lane boundaries
• Sensors:	Vision
• Actuators:	Steering Wheel, Accelerator, Brakes
• Actions:	Steer, speed up, brake
• Environment:	Freeway

Conflict Resolution by Action Selection Agents

- **Override:** CAA overrides LKA
- **Arbitrate:** if Obstacle is Close then CAA else LKA
- **Compromise:** Choose action that satisfies both agents
- Any combination of the above
- **Challenges:** Doing the right thing

The Right Thing = The Rational Action

- **Rational Action:** The action that maximizes the expected value of the performance measure given the percept sequence to date

  - Rational = Best ?
  - Rational = Optimal ?
  - Rational = Omniscience ?
  - Rational = Clairvoyant ?
  - Rational = Successful ?

The Right Thing = The Rational Action

- **Rational Action:** The action that maximizes the expected value of the performance measure <u>given the percept sequence to date</u>

  - Rational = Best — Yes, to the best of its knowledge
  - Rational = Optimal — Yes, to the best of its abilities (incl. its constraints)
  - Rational ≠ Omniscience
  - Rational ≠ Clairvoyant
  - Rational ≠ Successful

Behavior and performance of IAs

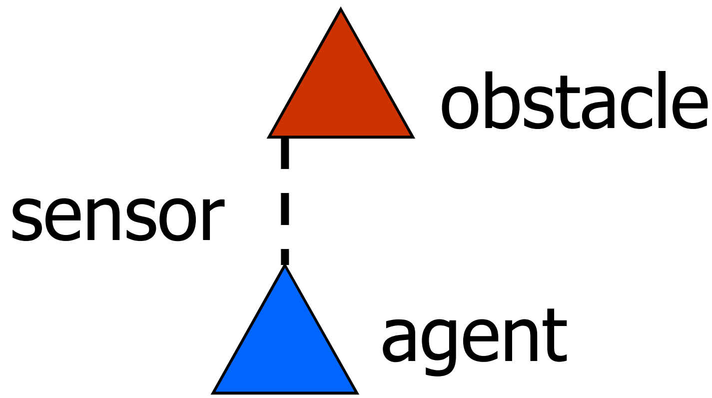
- **Perception** (sequence) to **Action Mapping:** $f: \mathcal{P}^* \rightarrow \mathcal{A}$
  - **Ideal mapping:** specifies which actions an agent ought to take at any point in time
  - **Description:** Look-Up-Table, Closed Form, etc.

- **Performance measure:** a *subjective* measure to characterize how successful an agent is (e.g., speed, power usage, accuracy, money, etc.)

- (degree of) **Autonomy:** to what extent is the agent able to make decisions and take actions on its own?

Look up table

Distance	Action
10	No action
5	Turn left 30 degrees
2	Stop



Closed form

- Output (degree of rotation) = F(distance)
- E.g., F(d) = 10/d   (distance cannot be less than 1/10)

How is an Agent different from other software?

- Agents are **autonomous**, that is, they act on behalf of the user
- Agents contain some level of **intelligence**, from fixed rules to learning engines that allow them to adapt to changes in the environment
- Agents don't only act **reactively**, but sometimes also **proactively**

How is an Agent different from other software?

- Agents have **social ability**, that is, they communicate with the user, the system, and other agents as required
- Agents may also **cooperate** with other agents to carry out more complex tasks than they themselves can handle
- Agents may **migrate** from one system to another to access remote resources or even to meet other agents

Environment Types

- Characteristics
  - Accessible vs. inaccessible
  - Deterministic vs. nondeterministic
  - Episodic vs. nonepisodic
  - Hostile vs. friendly
  - Static vs. dynamic
  - Discrete vs. continuous
  - Single Agent vs Multiple Agent

Environment Types

- Characteristics
  - Accessible vs. inaccessible (Fully vs Partially Observable)
    - Sensors give access to **complete** state of the environment.
  - Deterministic vs. nondeterministic (stochastic in AIMA)
    - The next state can be determined based on the current state and the action.
  - Episodic vs. nonepisodic (Sequential)
    - Episode: each episode is a set of perceive and action pairs
    - The quality of action does not depend on the previous episode
    - Episodic environments are simpler – no need to plan ahead – for example an agent to detect defective parts in assembly line
    - Non-Episodic examples include chess (need to look ahead)

Environment Types

- Characteristics
  - Hostile vs. friendly
  - Static vs. dynamic
    - Dynamic if the environment changes during deliberation
  - Discrete vs. continuous
    - Chess vs. driving
  - Single Agent vs. Multiple Agent
    - Cooperative vs Competitive (similar to hostile/friendly with multiple agents competing)

Environment types

Environment	Observable	Deterministic	Episodic	Static	Discrete
Crossword Puzzle **[S]**					
Chess with a Clock **[M]**					
Driving a Taxi **[S]**					
Interactive English Tutor **[M]**					
Part Sorting Robot **[S]**					

Environment types

Environment	Observable	Deterministic	Episodic	Static	Discrete
Crossword Puzzle **[S]**	Fully	Deterministic	Non-Episodic	Yes	Yes
Chess with a Clock **[M]**					
Driving a Taxi **[S]**					
Interactive English Tutor **[M]**					
Part Sorting Robot **[S]**					

Environment types

Environment	Observable	Deterministic	Episodic	Static	Discrete
Crossword Puzzle **[S]**	Fully	Deterministic	Non-Episodic	Yes	Yes
Chess with a Clock **[M]**	Fully	Deterministic	Non-Episodic	Yes	Yes
Driving a Taxi **[S]**					
Interactive English Tutor **[M]**					
Part Sorting Robot **[S]**					

Environment types

Environment	Observable	Deterministic	Episodic	Static	Discrete
Crossword Puzzle [S]	Fully	Deterministic	Non-Episodic	Yes	Yes
Chess with a Clock [M]	Fully	Deterministic	Non-Episodic	Yes	Yes
Driving a Taxi [S]	Partial	Non-Deterministic	Non-Episodic	Dynamic	Continuous
Interactive English Tutor [M]					
Part Sorting Robot [S]					

Environment types

Environment	Observable	Deterministic	Episodic	Static	Discrete
Crossword Puzzle **[S]**	Fully	Deterministic	Non-Episodic	Yes	Yes
Chess with a Clock **[M]**	Fully	Deterministic	Non-Episodic	Yes	Yes
Driving a Taxi **[S]**	Partial	Non-Deterministic	Non-Episodic	Dynamic	Continuous
Interactive English Tutor **[M]**	Partial	Non-Deterministic	Non-Episodic	Dynamic	Yes
Part Sorting Robot **[S]**					

Environment types

Environment	Observable	Deterministic	Episodic	Static	Discrete
Crossword Puzzle **[S]**	Fully	Deterministic	Non-Episodic	Yes	Yes
Chess with a Clock **[M]**	Fully	Deterministic	Non-Episodic	Semi-Static	Yes
Driving a Taxi **[S]**	Partial	Non-Deterministic	Non-Episodic	Dynamic	Continuous
Interactive English Tutor **[M]**	Partial	Non-Deterministic	Non-Episodic	Dynamic	Yes
Part Sorting Robot **[S]**	Partial	Non-Deterministic	Yes	Dynamic	Continuous

Structure of Intelligent Agents

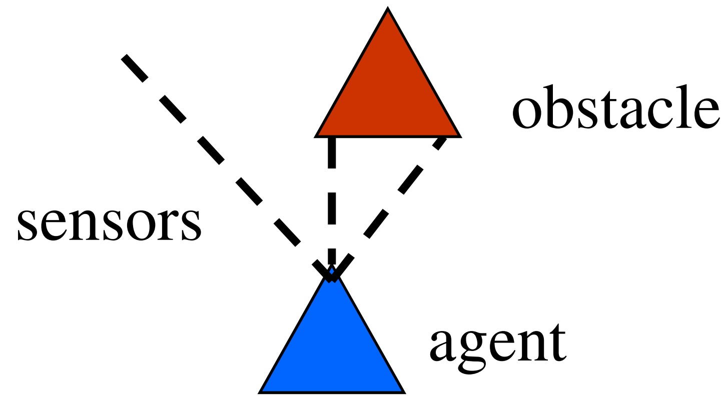
- Agent = architecture + program
- **Agent program:** the implementation of $f: \mathcal{P}^* \to \mathcal{A}$, the agent's perception-action mapping

function Skeleton-Agent(*Percept*) **returns** *Action*
    memory ← UpdateMemory(memory, *Percept*)
    Action ← ChooseBestAction(memory)
    memory ← UpdateMemory(memory, *Action*)
return *Action*

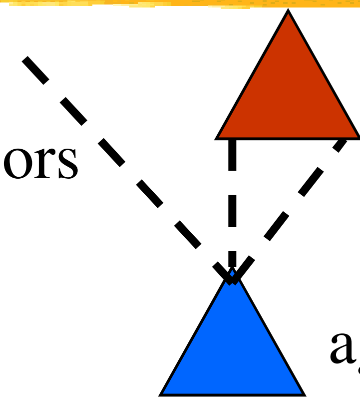
- **Architecture:** a device that can execute the agent program (e.g., general-purpose computer, specialized device, beobot, etc.)

Using a look-up-table to encode $f: \mathcal{P}^* \rightarrow \mathcal{A}$

- **Example:** Collision Avoidance
  - Sensors: 3 proximity sensors
  - Actuators: Steering Wheel, Brakes
- How to generate?
- How large?
- How to select action?



Using a look-up-table to encode f : $\mathcal{P}^* \rightarrow \mathcal{A}$



- Example: Collision Avoidance
  - Sensors: 3 proximity sensors
  - Actuators: Steering Wheel, Brakes

- How to generate: for each $p \in \mathcal{P}_\ell \times \mathcal{P}_m \times \mathcal{P}_r$ generate an appropriate action, $a \in \mathcal{S} \times \mathcal{B}$

- How large: size of table = #possible percepts times # possible actions = $|\mathcal{P}_\ell|\ |\mathcal{P}_m|\ |\mathcal{P}_r|\ |\mathcal{S}|\ |\mathcal{B}|$
  E.g., P = {close, medium, far}^{3} (for left, middle, right)
  S = {left, straight, right}
  B = {on, off}
  then size of table = 27*3*2 = 162

- How to select action? Search.

Agent types

- Reflex agents
- Reflex agents with internal states (Model-Based Reflex Agent in AIMA Chapter 2)
- Goal-based agents
- Utility-based agents
- Learning agents

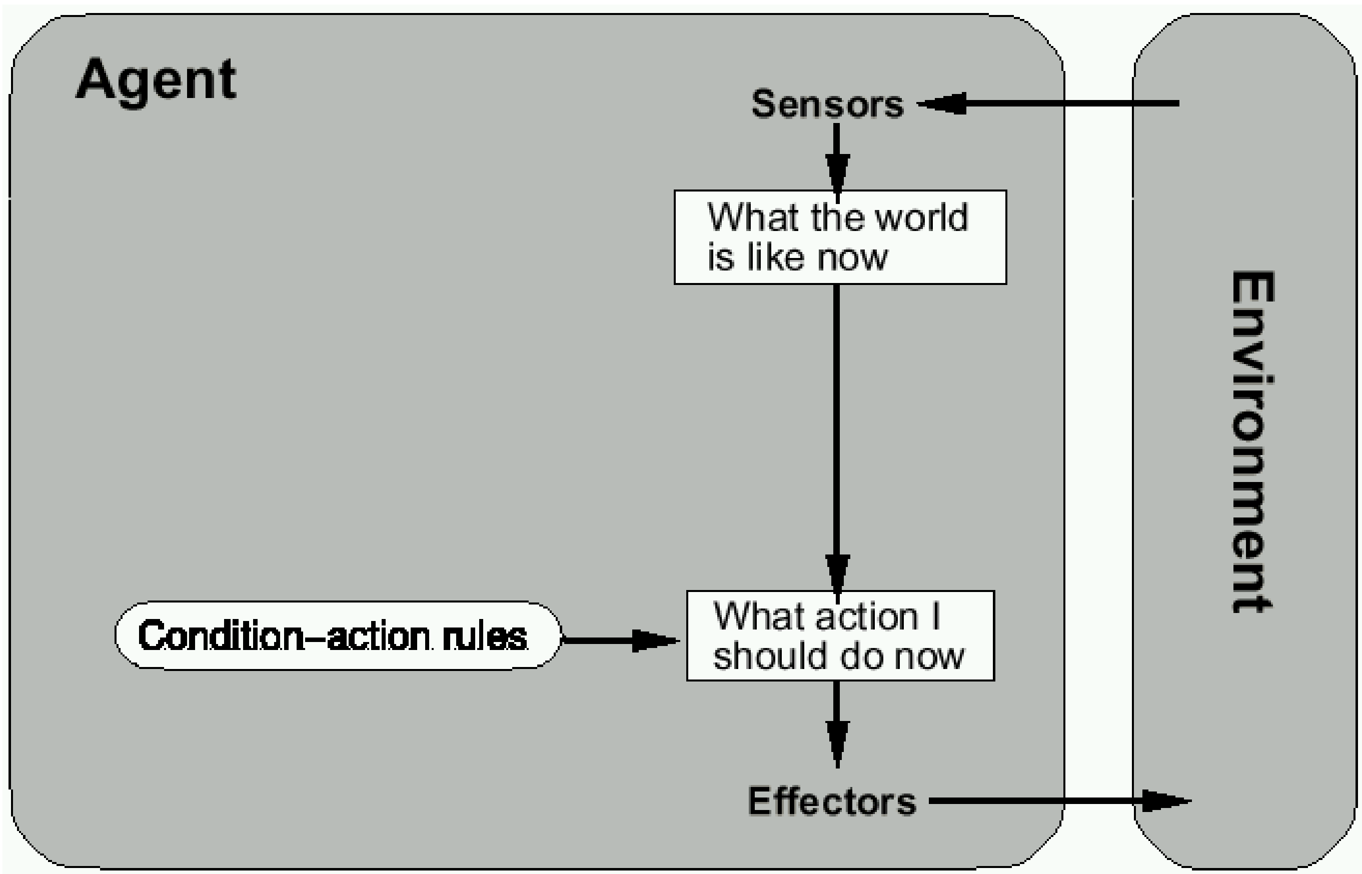
Agent types

- Reflex agents
  - Reactive: No memory
- Reflex agents with internal states
  - W/o previous state, may not be able to make decision
    - E.g. brake lights at night.
- Goal-based agents
  - Goal information needed to make decision

Agent types

- Utility-based agents
  - How well can the goal be achieved (degree of happiness)
  - What to do if there are conflicting goals?
    - Speed and safety
  - Which goal should be selected if several can be achieved?

Reflex agents

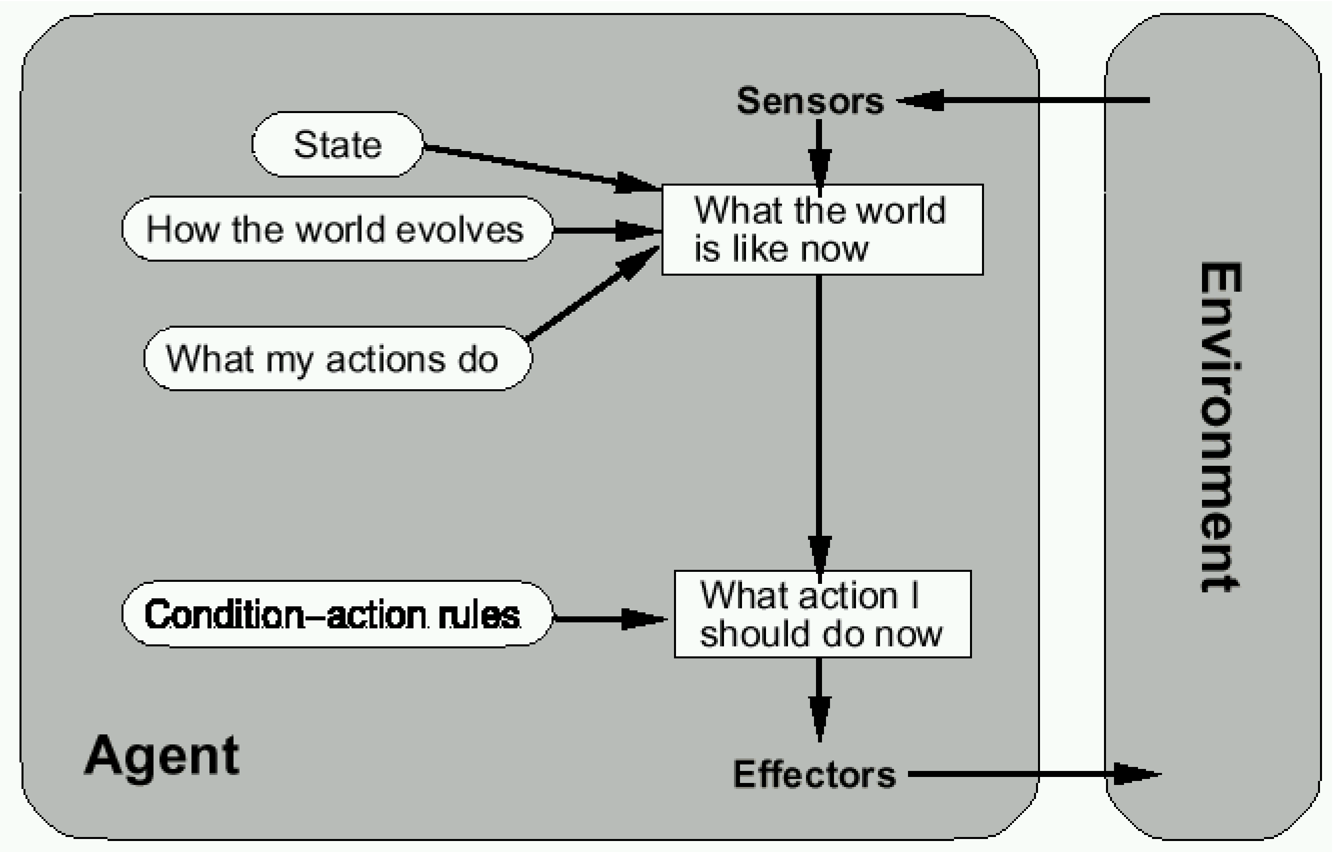


Reflex agents

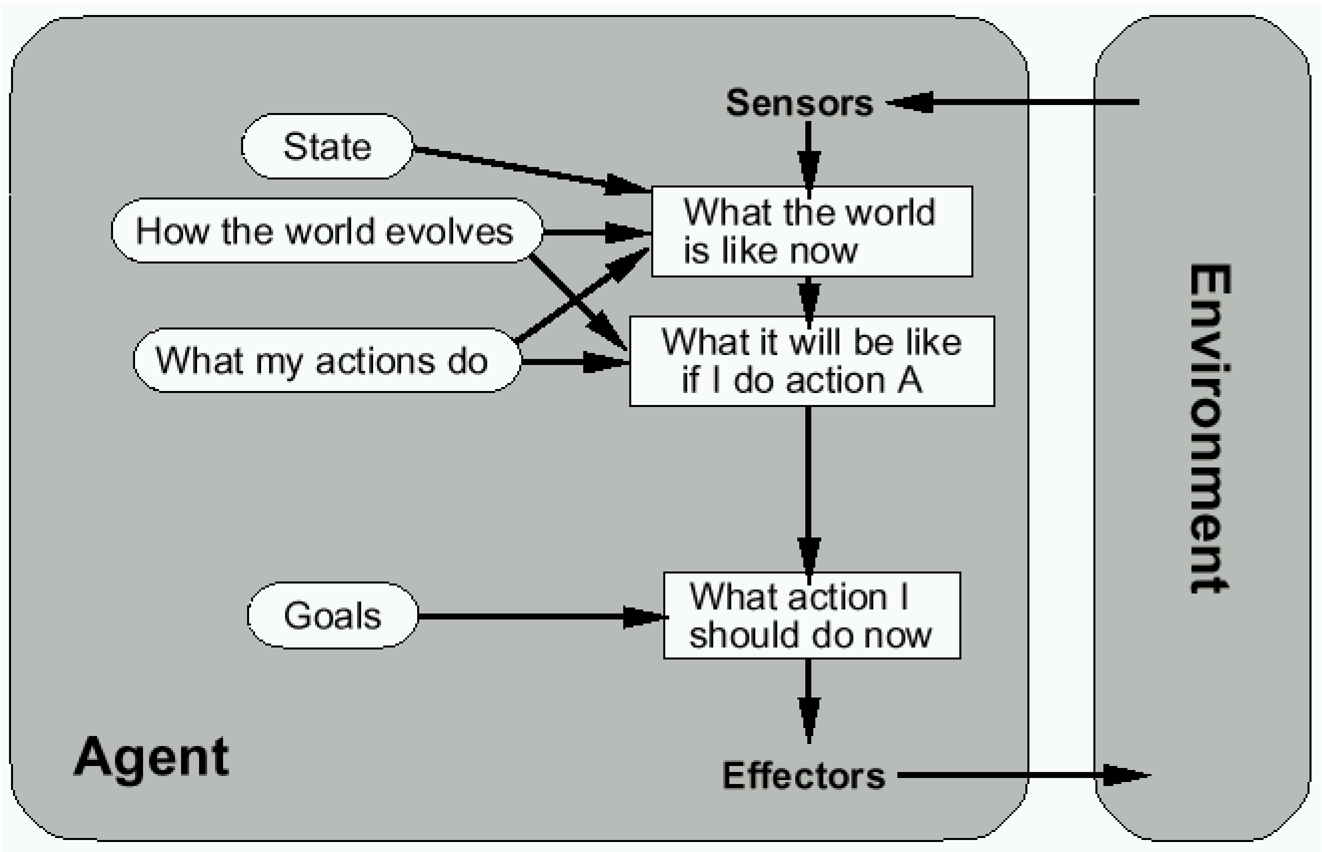
- Reflex agents do not have internal symbolic models.
- Act by stimulus-response to the current state of the environment.
- Each reflex agent is simple and interacts with others in a basic way.
- Complex patterns of behavior emerge from their interaction.

- **Benefits:** robustness, fast response time
- **Challenges:** scalability, how intelligent?
and how do you debug them?

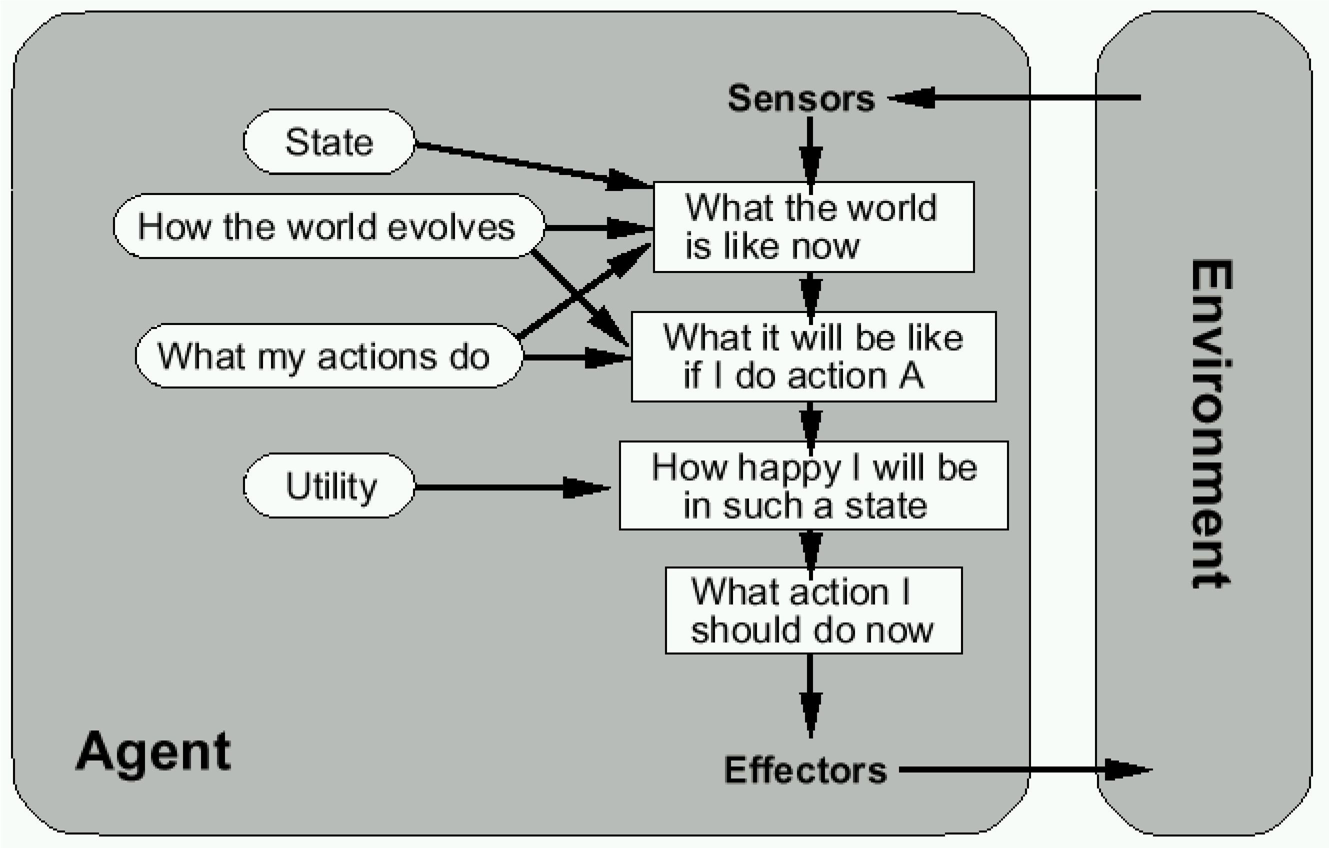
Reflex agents w/ state – also called Model-based Reflex Agent



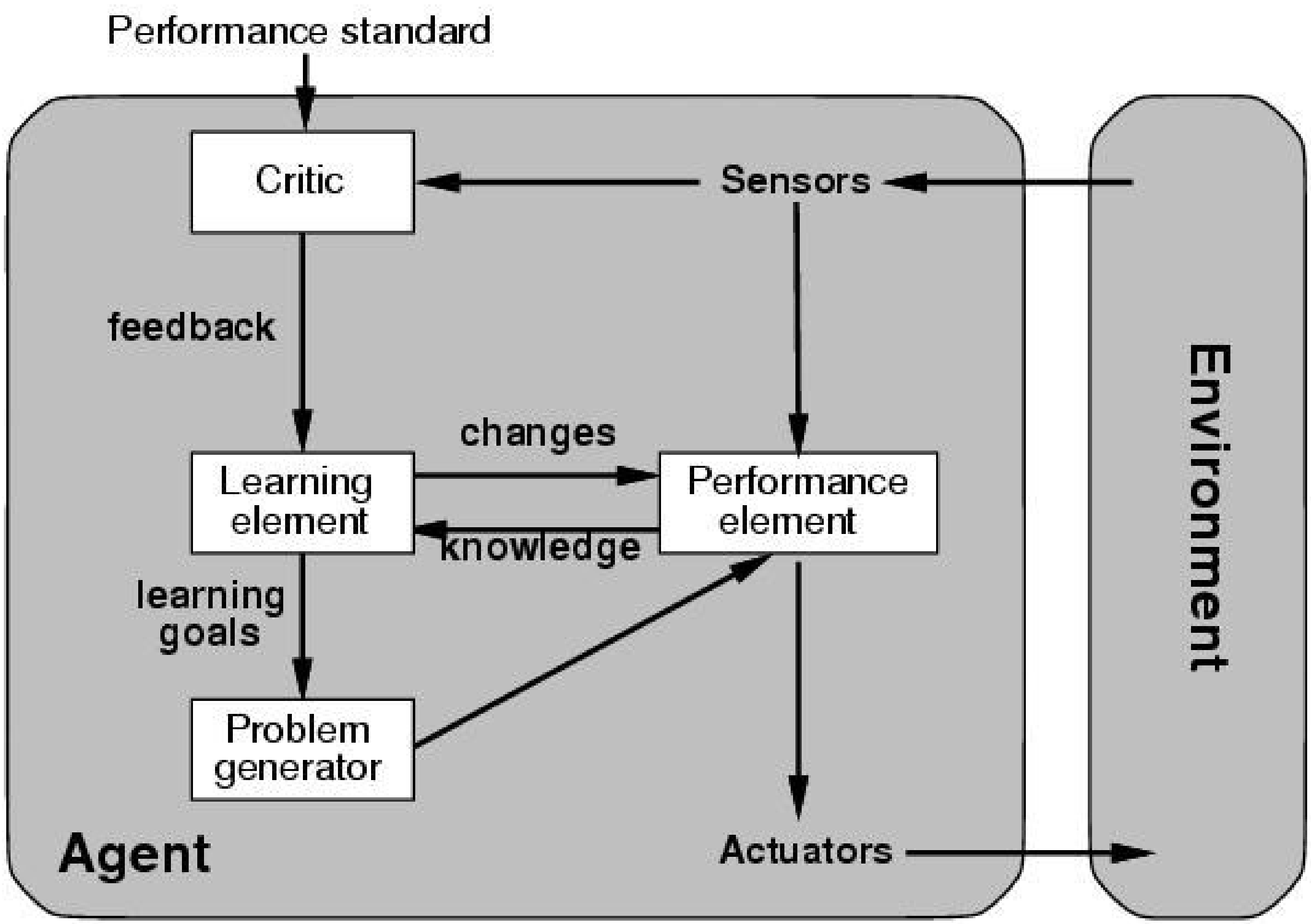
Goal-based agents



Utility-based agents



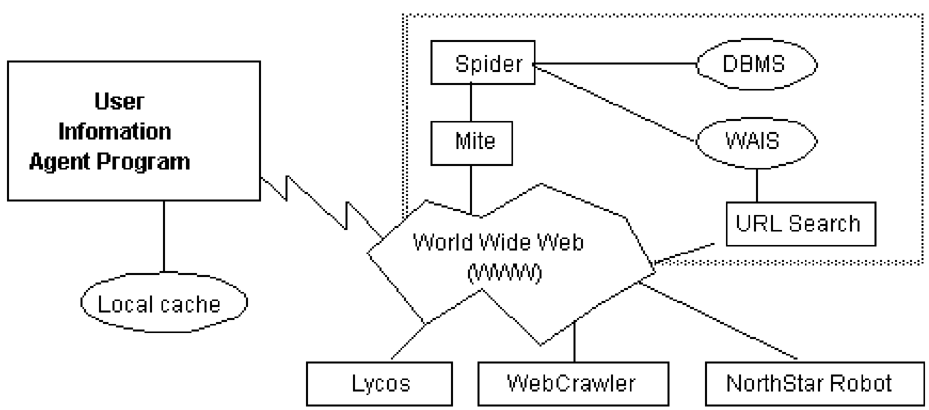
Learning agents



Information agents ("Bots")

- Manage the explosive growth of information.
- Manipulate or collate information from many distributed sources.
- Information agents can be mobile or static.

- Examples:
  - BargainFinder comparison shops among Internet stores for CDs
  - FIDO the Shopping Doggie (out of service)
  - Internet Softbot infers which internet facilities (finger, ftp, gopher) to use and when from high-level search requests.
- Challenge: ontologies for annotating Web pages (eg, SHOE).



Summary

- **Intelligent Agents:**
  - Anything that can be *viewed as* **perceiving** its **environment** through **sensors** and **acting** upon that environment through its **Actuators** to maximize progress towards its **goals**.
  - PAGE (Percepts, Actions, Goals, Environment)
  - Described as a Perception (sequence) to Action Mapping: $f: \mathcal{P}^* \rightarrow \mathcal{A}$
  - Using look-up-table, closed form, etc.

- **Agent Types:** Reflex, state-based, goal-based, utility-based

- **Rational Action:** The action that maximizes the expected value of the performance measure <u>given the percept sequence to date</u>