CMSC 671
Fall 2010

Thu 9/2/10
Agents

Prof. Laura Zavala, laura.zavala@umbc.edu, ITE 373, 410-455-8775

Last Class

- **AI**: Design of agents that act rationally
- **Why AI:** Engineering, Cognitive Science, Philosophy
- **Let's do some AI:** It is straightforward to write a computer program to play tic-tac-toe perfectly
  - State space complexity = 765, Game tree complexity = 26830
- **Some AI milestones**
  - 1997: Deep Blue beats Garry Kasparov (world champion)
  - 2007: Checkers is solved! Checkers program CHINOOK cannot lose (it can draw)

Possible Approaches

	Like humans	Well
Think	GPS	Rational agents
Act	Eliza	Heuristic systems

AI tends to work mostly in this area

What Can AI Systems Do?

- **Computer vision:** face recognition from a large set
- **Robotics:** autonomous (mostly) automobile
- **Natural language processing:** simple machine translation
- **Expert systems:** medical diagnosis in a narrow domain
- **Spoken language systems:** ~1000 word continuous speech
- **Planning and scheduling:** Hubble Telescope experiments
- **Learning:** text categorization into ~1000 topics
- **User modeling:** Bayesian reasoning in Windows help (the infamous paper clip…)
- **Games:** Grand Master level in chess (world champion), perfect play in checkers, professional-level Go players

What Can't AI Systems Do Yet?

Exhibit true autonomy and intelligence!

- Understand natural language robustly (e.g., read and understand articles in a newspaper)
- Surf the web
- Interpret an arbitrary visual scene
- Learn a natural language
- Play Go as well as the best human players
- Construct plans in dynamic real-time domains
- Refocus attention in complex environments
- Perform life-long learning

Today's class

- What's an agent?
  - Definition of an agent
  - Rationality and autonomy
  - Types of agents
  - Properties of environments
- Lisp

Intelligent Agents

Chapter 2

Agents

Agents

An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators.

Human agent has, for example:

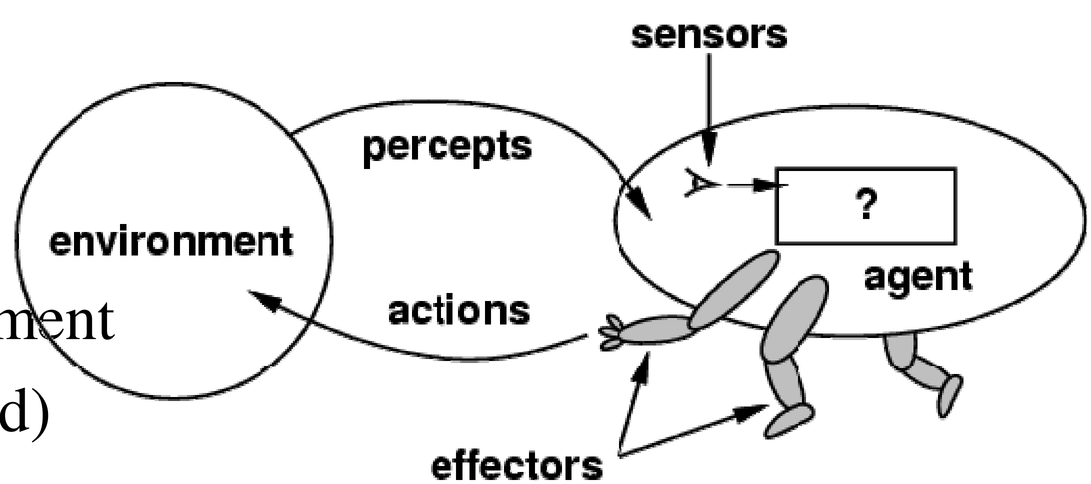
- eyes, ears, and other organs as sensors;
- hands, legs, mouth, and other body parts as actuators.

Robotic agent has, for example:

- cameras and infrared range finders as sensors;
- various motors as actuators.

How do you design an intelligent agent?

- Definition: An **intelligent agent** perceives its environment via **sensors** and acts rationally upon that environment with its **effectors**.
- A discrete agent receives **percepts** one at a time, and maps this percept sequence to a sequence of discrete **actions**.
- Properties
    - Autonomous
    - Reactive to the environment
    - Pro-active (goal-directed)
    - Interacts with other agents via the environment



What do you mean, sensors/percepts and effectors/actions?

sensors/percepts, effectors/actions

- Humans
  - Sensors: Eyes (vision), ears (hearing), skin (touch), tongue (gustation), nose (olfaction), neuromuscular system (proprioception)
  - Percepts:
    - At the lowest level – electrical signals from these sensors
    - After preprocessing – objects in the visual field (location, textures, colors, …), auditory streams (pitch, loudness, direction), …
  - Effectors: limbs, digits, eyes, tongue, …
  - Actions: lift a finger, turn left, walk, run, carry an object, …

sensors/percepts, effectors/actions

- The Point: percepts and actions need to be carefully defined, possibly at different levels of abstraction

A more specific example: Automated taxi driving system

- **Percepts**: Video, sonar, speedometer, odometer, engine sensors, keyboard input, microphone, GPS, …
- **Actions**: Steer, accelerate, brake, horn, speak/display, …
- **Goals**: Maintain safety, reach destination, maximize profits (fuel, tire wear), obey laws, provide passenger comfort, …
- **Environment**: U.S. urban streets, freeways, traffic, pedestrians, weather, customers, …

Different aspects of driving may require different types of agent programs!

Rationality

- An ideal **rational agent** should, for each possible percept sequence, do whatever actions will maximize its expected performance measure based on

    (1) the percept sequence, and

    (2) its built-in and acquired knowledge.

- *Rationality includes information gathering*
- *Rationality need a performance measure*

Autonomy

- A system is autonomous to the extent that its own behavior is determined by its own experience.
- Therefore, a system is not autonomous if it is guided by its designer according to a priori decisions.
- Can computers/robots ever be autonomous?

Autonomy

- *How do humans achieve autonomy?*

- To survive, agents must have:
  - Enough built-in knowledge to survive.
  - The ability to learn.

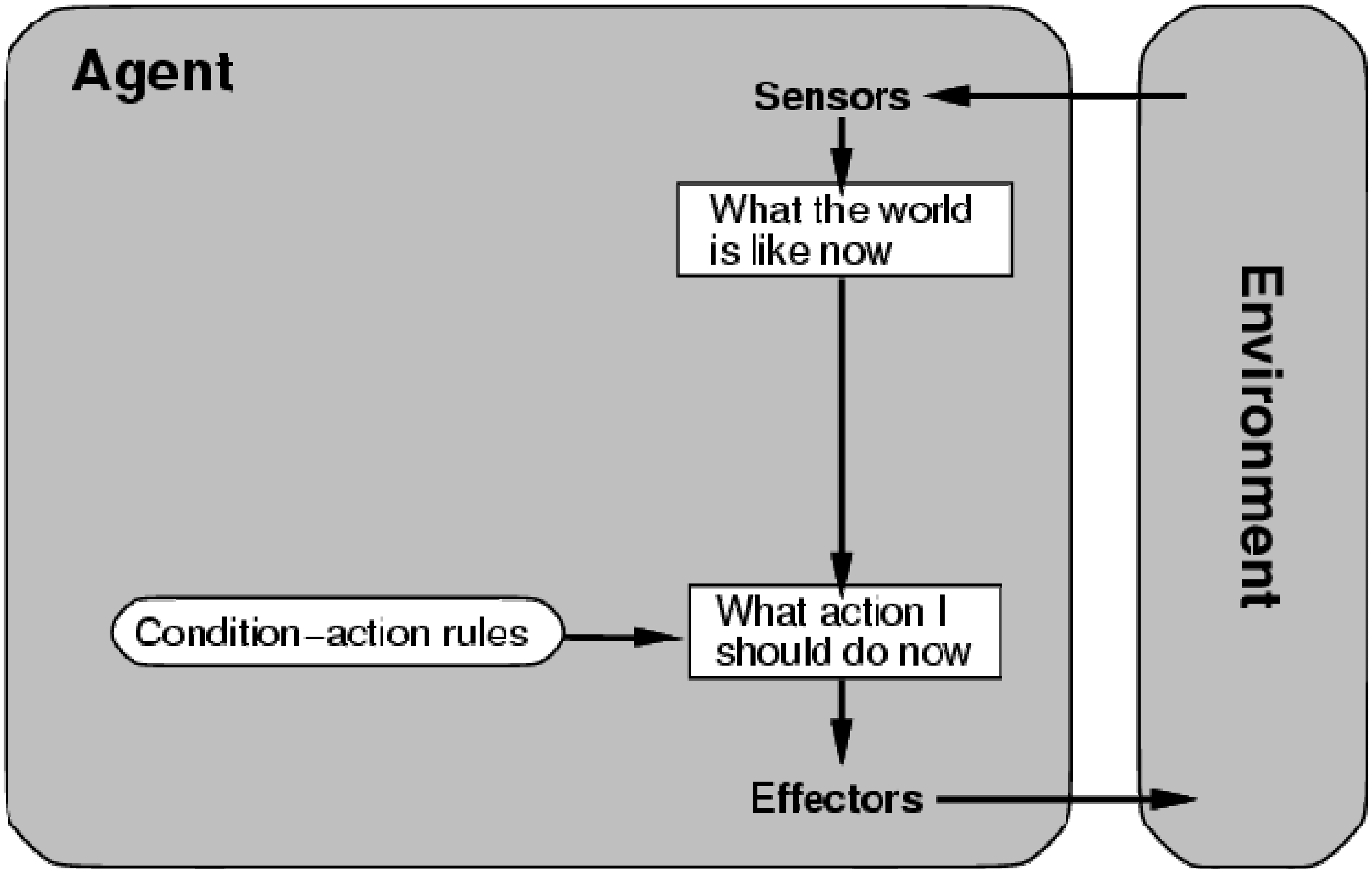
Some agent types

- **(0) Table-driven agents**
  - use a percept sequence/action table in memory to find the next action. They are implemented by a (large) **lookup table**.
- **(1) Simple reflex agents**
  - are based on **condition-action rules**, implemented with an appropriate production system. They are stateless devices which do not have memory of past world states.
- **(2) Agents with memory**
  - have **internal state**, which is used to keep track of past states of the world.

Some agent types

- **(3) Agents with goals**
  - are agents that, in addition to state information, have **goal information** that describes desirable situations. Agents of this kind take future events into consideration.
- **(4) Utility-based agents**
  - base their decisions on **classic axiomatic utility theory** in order to act rationally.

Table-driven/reflex agent architecture



(0) Table-driven agents

- **Table lookup** of percept-action pairs mapping from every possible perceived state to the optimal action for that state

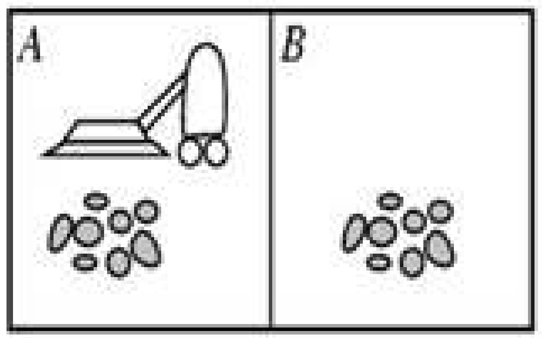
- *Problems?*
  - Too big to generate and to store (Chess has about 10^{120} states, for example)
  - No knowledge of non-perceptual parts of the current state
  - Not adaptive to changes in the environment; requires entire table to be updated if changes occur
  - Looping: Can't make actions conditional on previous actions/states

(1) Simple reflex agents

- **Rule-based reasoning** to map from percepts to optimal action; each rule handles a **collection** of perceived states
- *Problems?*
  - Still usually too big to generate and to store
  - Still no knowledge of non-perceptual parts of state
  - Still not adaptive to changes in the environment; requires collection of rules to be updated if changes occur
  - Still can’t make actions conditional on previous state

Example

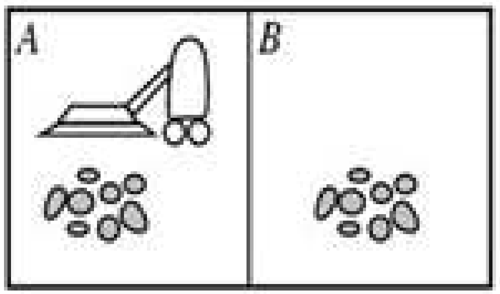
The Vacuum-Cleaner Mini-World



- Environment: square A and B
- Percepts: location and status, e.g., [A, Dirty]
- Actions: left, right, suck, and no-op

Example

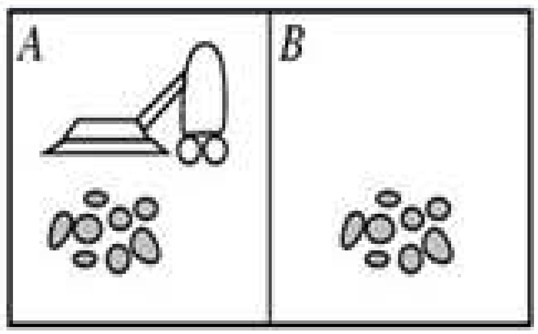
The Vacuum-Cleaner Mini-World



World State	Action
[A, Clean]	Right
[A, Dirty]	Suck
[B, Clean]	Left
[B, Dirty]	Suck
[A, Dirty], [A, Clean]	Right
[A, Clean], [B, Dirty]	Suck
[B, Dirty], [B, Clean]	Left
[B, Clean], [A, Dirty]	Suck
[A, Clean], [B, Clean]	No-op
[B, Clean], [A, Clean]	No-op

Example

The Vacuum-Cleaner Mini-World



function REFLEX-VACUUM-AGENT ([*location*, *status*]) return an action

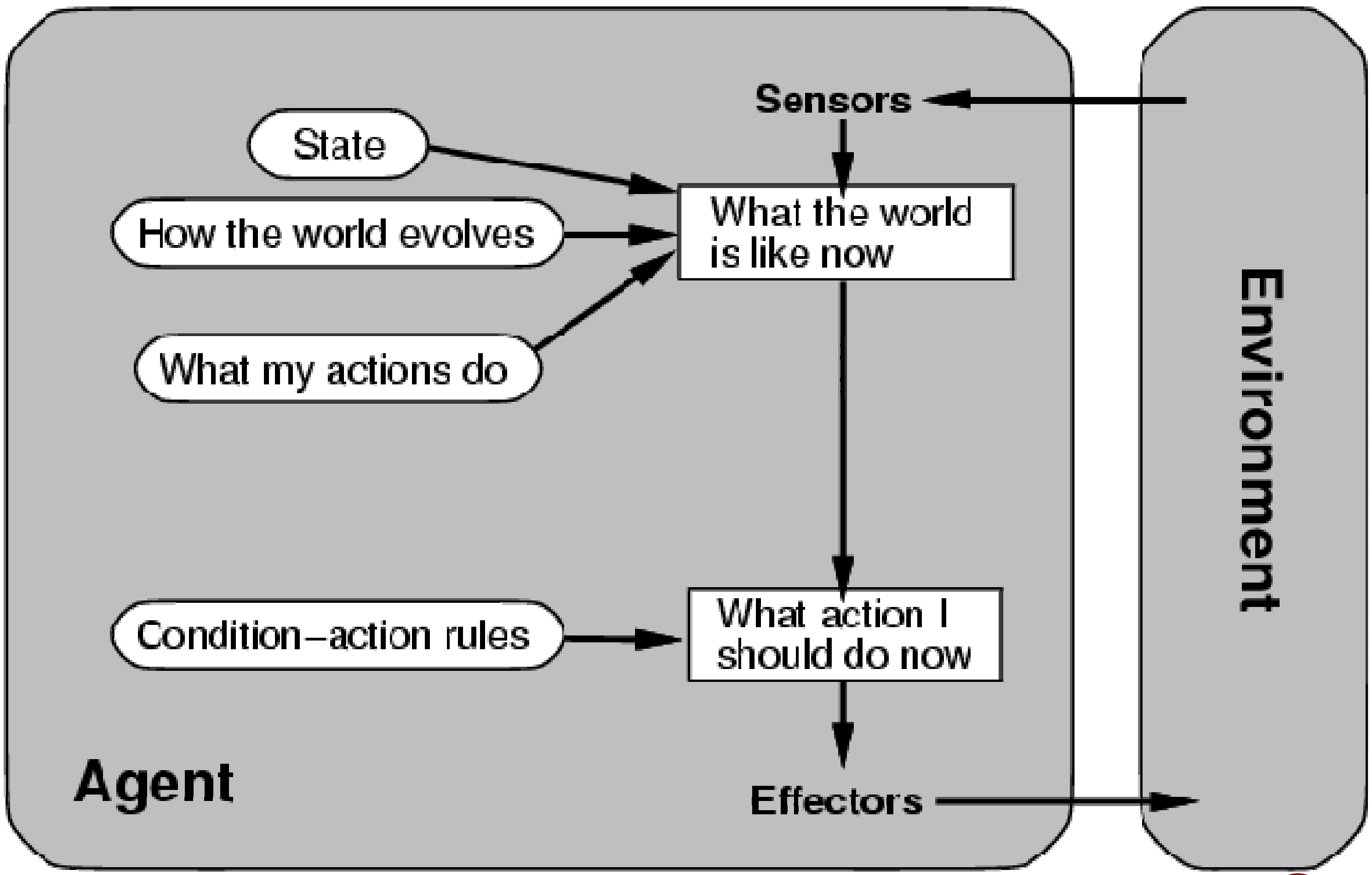
if *status* == *Dirty* then return *Suck*

else if *location* == *A* then return *Right*

else if *location* == *B* then return *Left*

Does not work this way. Need full state space (table) or memory.

(2) Architecture for an agent with memory



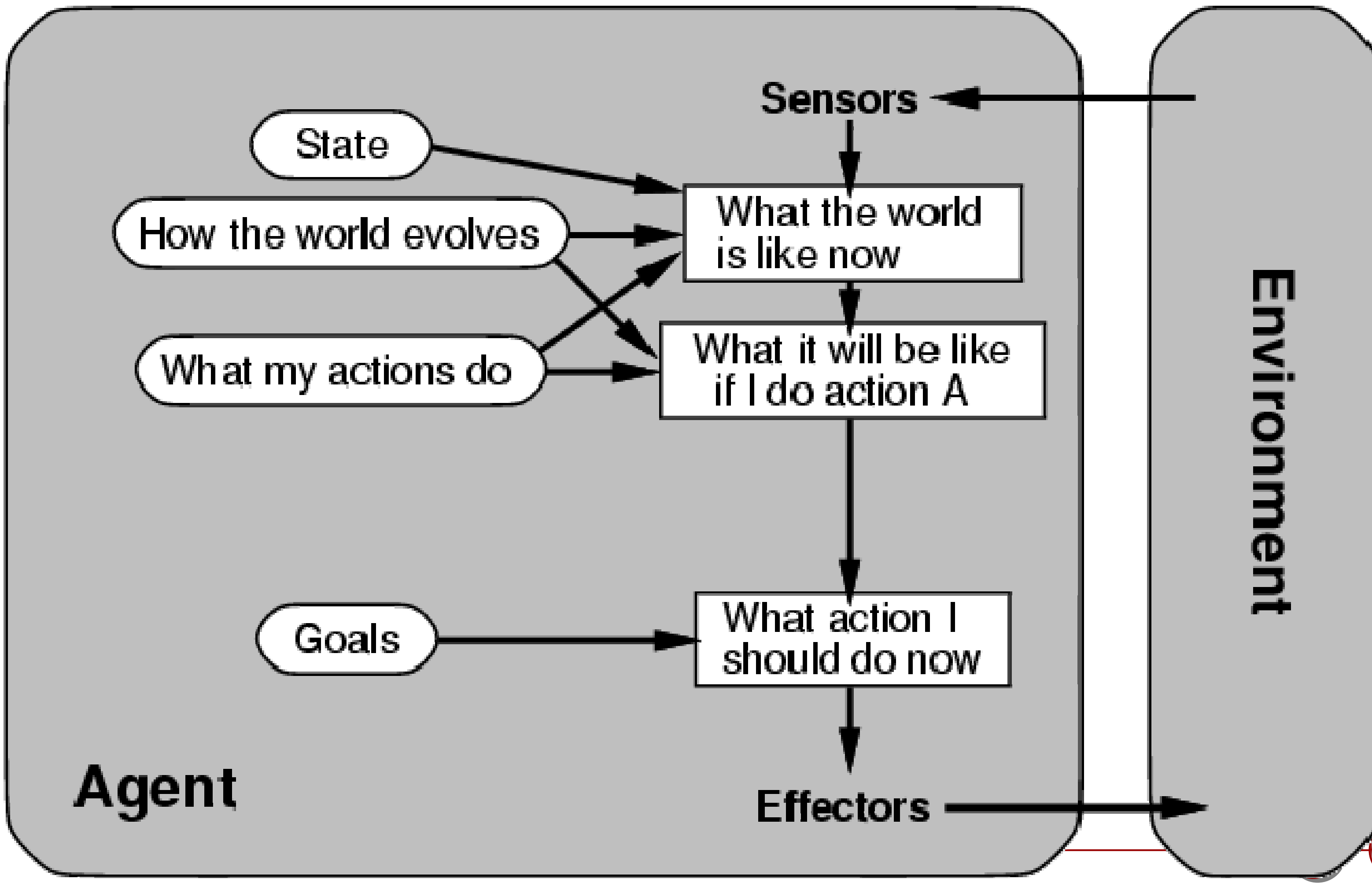
(2) Agents with memory

- Encode “internal state” of the world to remember the past as contained in earlier percepts.
- *Why is that even needed?*
- Sensors do not usually give the entire state of the world at each input, so perception of the environment is captured over time.
- “State” is used to encode different "world states" that generate the same immediate percept.

An example: Brooks's Subsumption Architecture

- Main idea: build complex, intelligent robots by decomposing behaviors into a hierarchy of skills, each completely defining a complete percept-action cycle for one very specific task.
- Examples: avoiding contact, wandering, exploring, recognizing doorways, etc.
- Each behavior is modeled by a finite-state machine with a few states (though each state may correspond to a complex function or module).
- Behaviors are loosely coupled, asynchronous interactions.

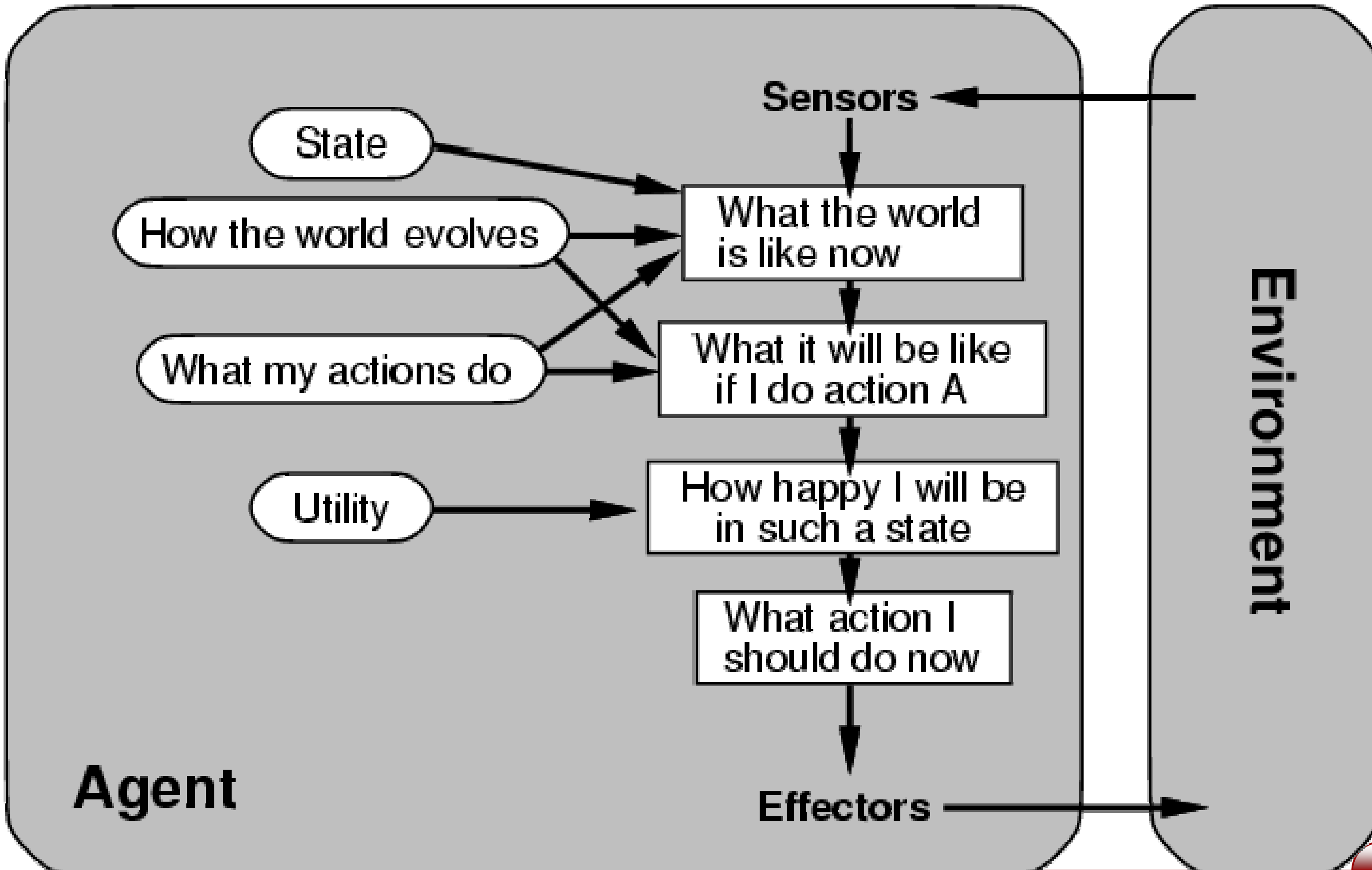
(3) Architecture for goal-based agent



(3) Goal-based agents

- Choose actions so as to achieve a (given or computed) goal.
- A goal is a description of a desirable situation.
- Keeping track of the current state is often not enough – need to add goals to decide which situations are good
- **Deliberative** instead of **reactive**.
- May have to consider long sequences of possible actions before deciding if goal is achieved – involves consideration of the future, *"what will happen if I do...?"*

(4) Architecture for a complete utility-based agent



(4) Utility-based agents

- When there are multiple possible alternatives, how to decide which one is best?
- A goal specifies a crude distinction between a happy and unhappy state
  - often need a more general performance measure that describes "degree of happiness."
  - Utility function **U: State → Real**

Properties of Environments

- **Fully observable/Partially observable.**
  - If an agent's sensors give it access to the complete state of the environment needed to choose an action, the environment is **fully observable**.
  - Such environments are convenient, since the agent is freed from the task of keeping track of the changes in the environment.
- **Deterministic/Stochastic**.
  - An environment is **deterministic** if the next state of the environment is completely determined by the current state of the environment and the action of the agent; in a **stochastic** environment, there are multiple, unpredictable outcomes

Fully observable + Deterministic ➔ no need to deal with uncertainty

Properties of Environments II

- **Episodic/Sequential**.
  - An **episodic** environment means that subsequent episodes do not depend on what actions occurred in previous episodes.
  - In a **sequential** environment, the agent engages in a series of connected episodes.
  - Such environments do not require the agent to plan ahead.
- **Static/Dynamic.**
  - A **static** environment does not change while the agent is thinking.
  - The passage of time as an agent deliberates is irrelevant.
  - The agent doesn’t need to observe the world during deliberation.

Properties of Environments III

- **Discrete/Continuous.**
  - If the number of distinct percepts and actions is limited, the environment is **discrete**, otherwise it is **continuous**.
- **Single agent/Multi-agent.**
  - If the environment contains other intelligent agents, the agent needs to be concerned about strategic, game-theoretic aspects of the environment (for either cooperative *or* competitive agents)
  - Most engineering environments don't have multi-agent properties, whereas most social and economic systems get their complexity from the interactions of (more or less) rational agents.

Characteristics of environments

	Fully observable?	Deterministic?	Episodic?	Static?	Discrete?	Single agent?
Solitaire						
Backgammon						
Taxi driving						
Internet shopping						
Medical diagnosis						

Characteristics of environments

	Fully observable?	Deterministic?	Episodic?	Static?	Discrete?	Single agent?
Solitaire	No					
Backgammon	Yes					
Taxi driving	No					
Internet shopping	No					
Medical diagnosis	**No**					

Characteristics of environments

	Fully observable?	Deterministic?	Episodic?	Static?	Discrete?	Single agent?
Solitaire	No	Yes				
Backgammon	Yes	No				
Taxi driving	No	No				
Internet shopping	No	No				
Medical diagnosis	**No**	**No**				

Characteristics of environments

	Fully observable?	Deterministic?	Episodic?	Static?	Discrete?	Single agent?
Solitaire	No	Yes	Yes			
Backgammon	Yes	No	No			
Taxi driving	No	No	No			
Internet shopping	No	No	No			
Medical diagnosis	**No**	**No**	**No**			

Characteristics of environments

	Fully observable?	Deterministic?	Episodic?	Static?	Discrete?	Single agent?
Solitaire	No	Yes	Yes	Yes		
Backgammon	Yes	No	No	Yes		
Taxi driving	No	No	No	No		
Internet shopping	No	No	No	No		
Medical diagnosis	**No**	**No**	**No**	**No**		

Characteristics of environments

	Fully observable?	Deterministic?	Episodic?	Static?	Discrete?	Single agent?
Solitaire	No	Yes	Yes	Yes	Yes	
Backgammon	Yes	No	No	Yes	Yes	
Taxi driving	No	No	No	No	No	
Internet shopping	No	No	No	No	Yes	
Medical diagnosis	**No**	**No**	**No**	**No**	**No**	

Characteristics of environments

	Fully observable?	Deterministic?	Episodic?	Static?	Discrete?	Single agent?
Solitaire	No	Yes	Yes	Yes	Yes	Yes
Backgammon	Yes	No	No	Yes	Yes	No
Taxi driving	No	No	No	No	No	No
Internet shopping	No	No	No	No	Yes	No
Medical diagnosis	**No**	**No**	**No**	**No**	**No**	**Yes**

→ Lots of real-world domains fall into the hardest case!

Summary

- An **agent** perceives and acts in an environment, has an architecture, and is implemented by an agent program.
- An **ideal agent** always chooses the action which maximizes its expected performance, given its percept sequence so far.
- An **autonomous agent** uses its own experience rather than built-in knowledge of the environment by the designer.
- An **agent program** maps from percept to action and updates its internal state.
  - **Reflex agents** respond immediately to percepts.
  - **Goal-based agents** act in order to achieve their goal(s).
  - **Utility-based agents** maximize their own utility function.
- **Representing knowledge** is important for successful agent design.
- The most challenging environments are partially observable, stochastic, sequential, dynamic, and continuous, and contain multiple intelligent agents.