CMSC 471

Intelligent Agents

Russell & Norvig
Chapter 2

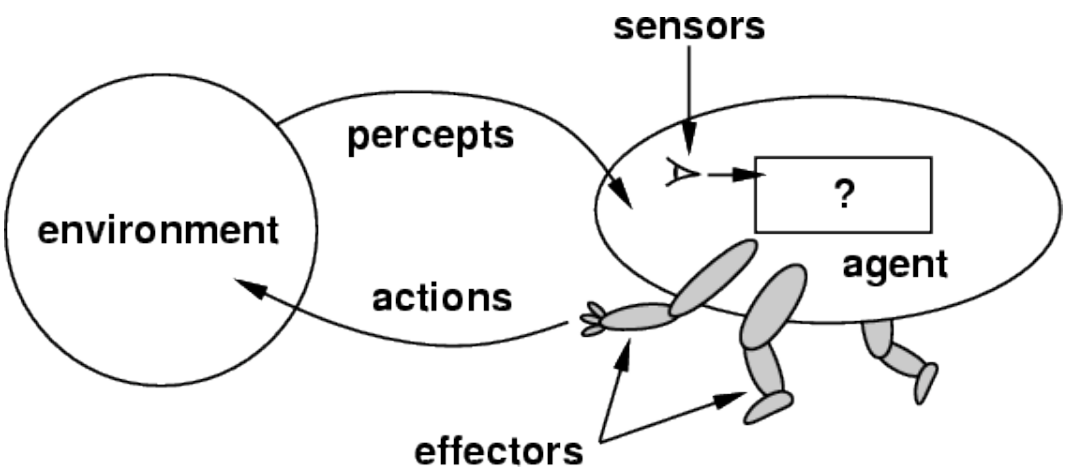
Today’s class

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

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**.

- Properties
  - –Autonomous
  - –Reactive to the environment
  - –Pro-active (goal-directed)
  - –Interacts with other agents via the environment



Examples of sensors/percepts and effectors/actions?

- Humans
  - Sensors: Eyes (vision), ears (hearing), skin (touch), tongue (gustation), nose (olfaction)
  - 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, …

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, …

Rationality

- An ideal **rational agent** should maximize its expected performance.

- Rationality includes information gathering (exploration).
- Rationality → Need a performance measure to say how well a task has been achieved (fitness).

Autonomy

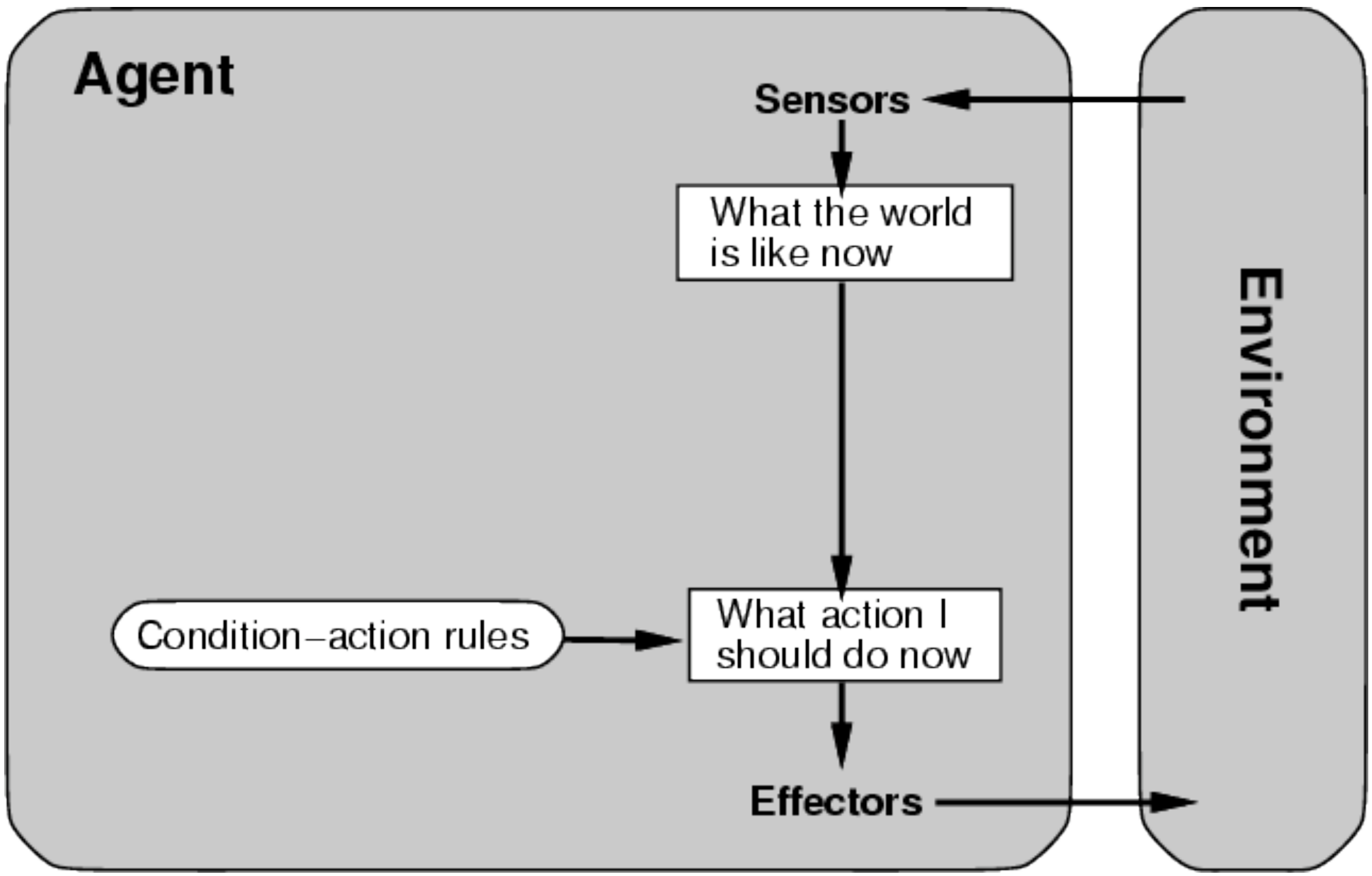
- A system is autonomous to the extent that its own behavior is determined by its own experience.

- A system is not autonomous if it is guided by its designer according to a priori decisions.

Examples of 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.
- **(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.

(0/1) Table-driven/reflex agent architecture



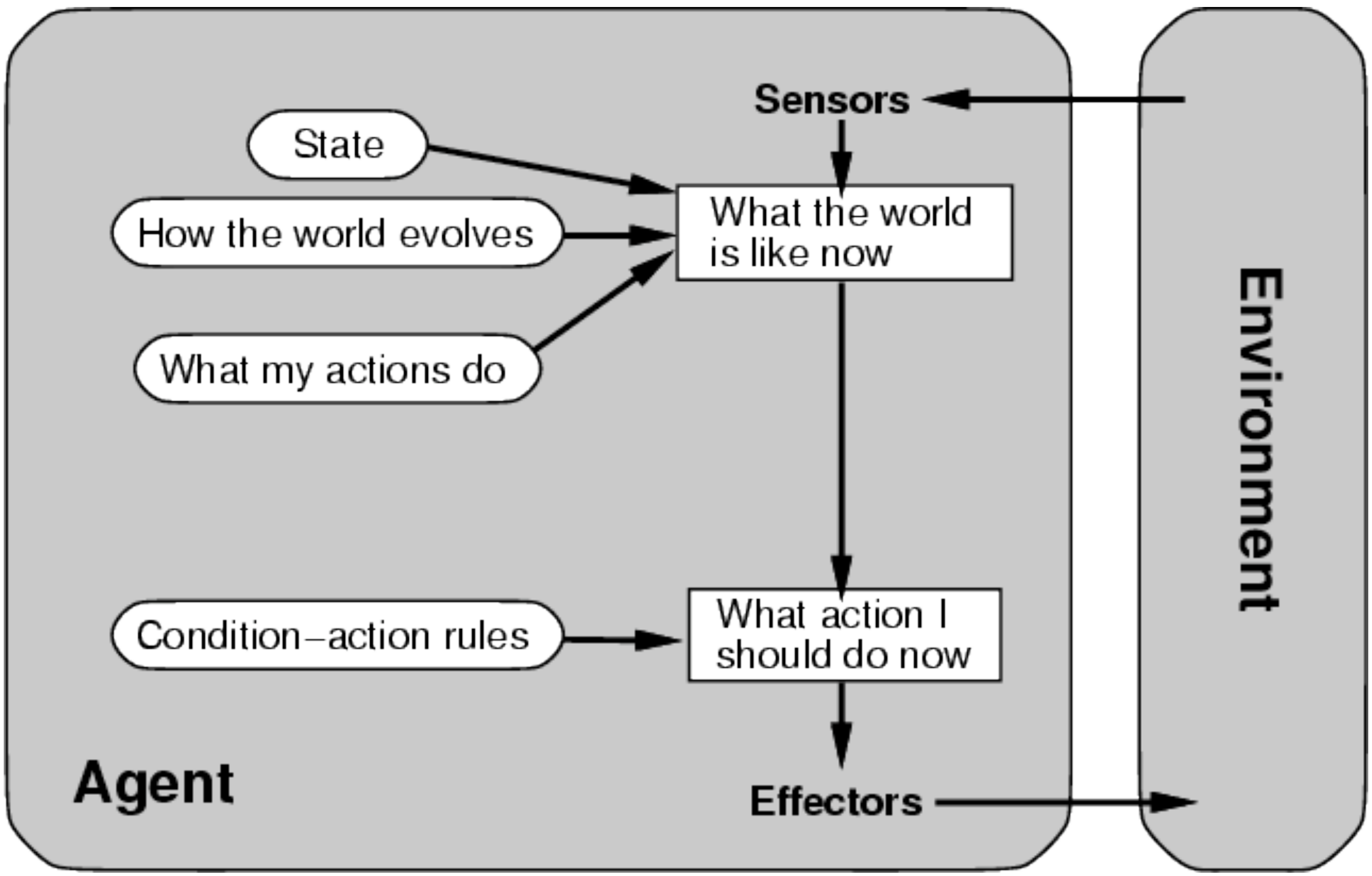
(0) Table-driven agents

- **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

(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

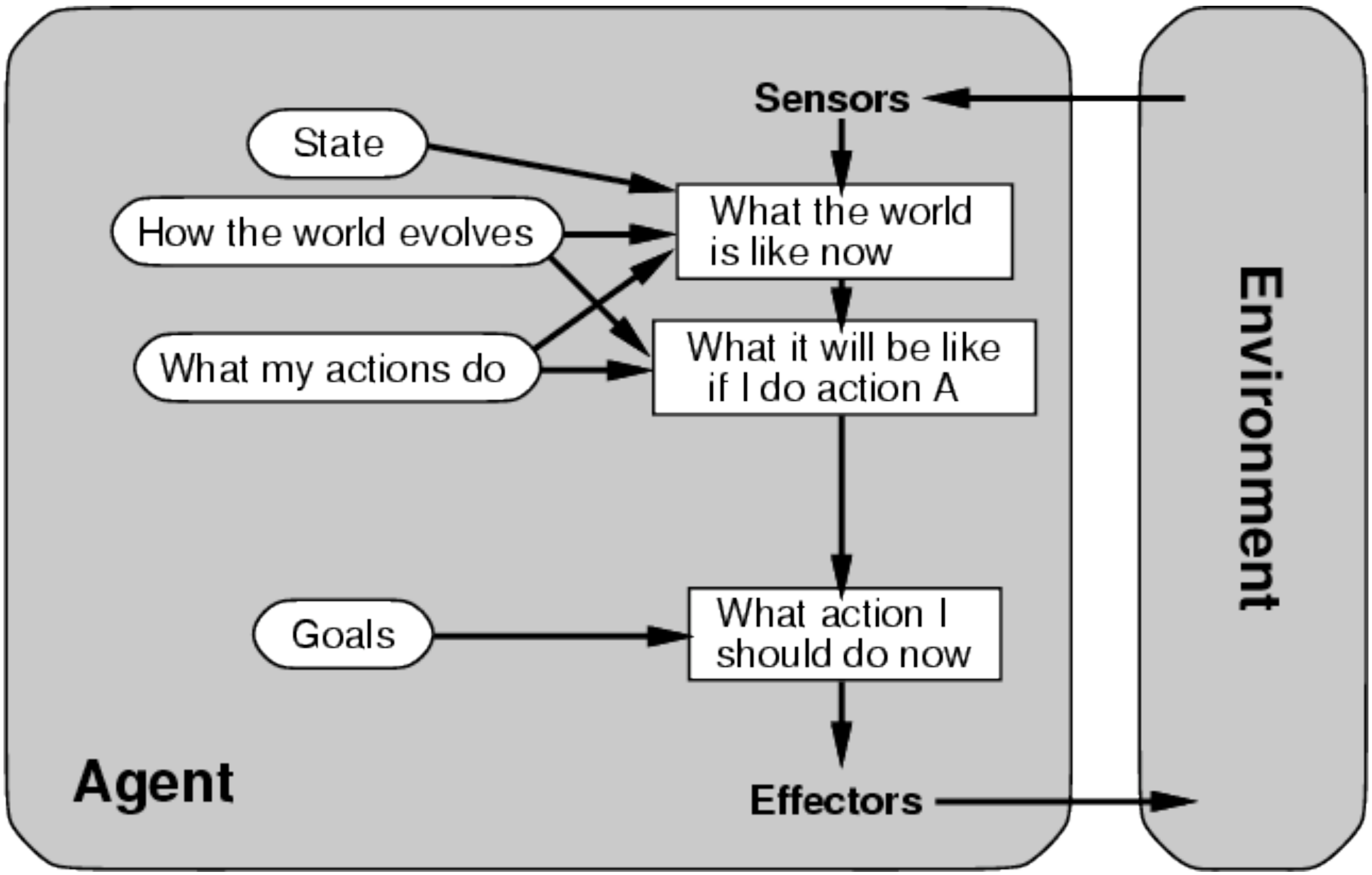
(2) Architecture for an agent with memory



(2) Agents with memory

- Encode “internal state” of the world.
- Needed because sensors only give a snapshot.
- Requires ability to represent change in the world; one possibility is to represent just the latest state, but then can’t reason about hypothetical courses of action.

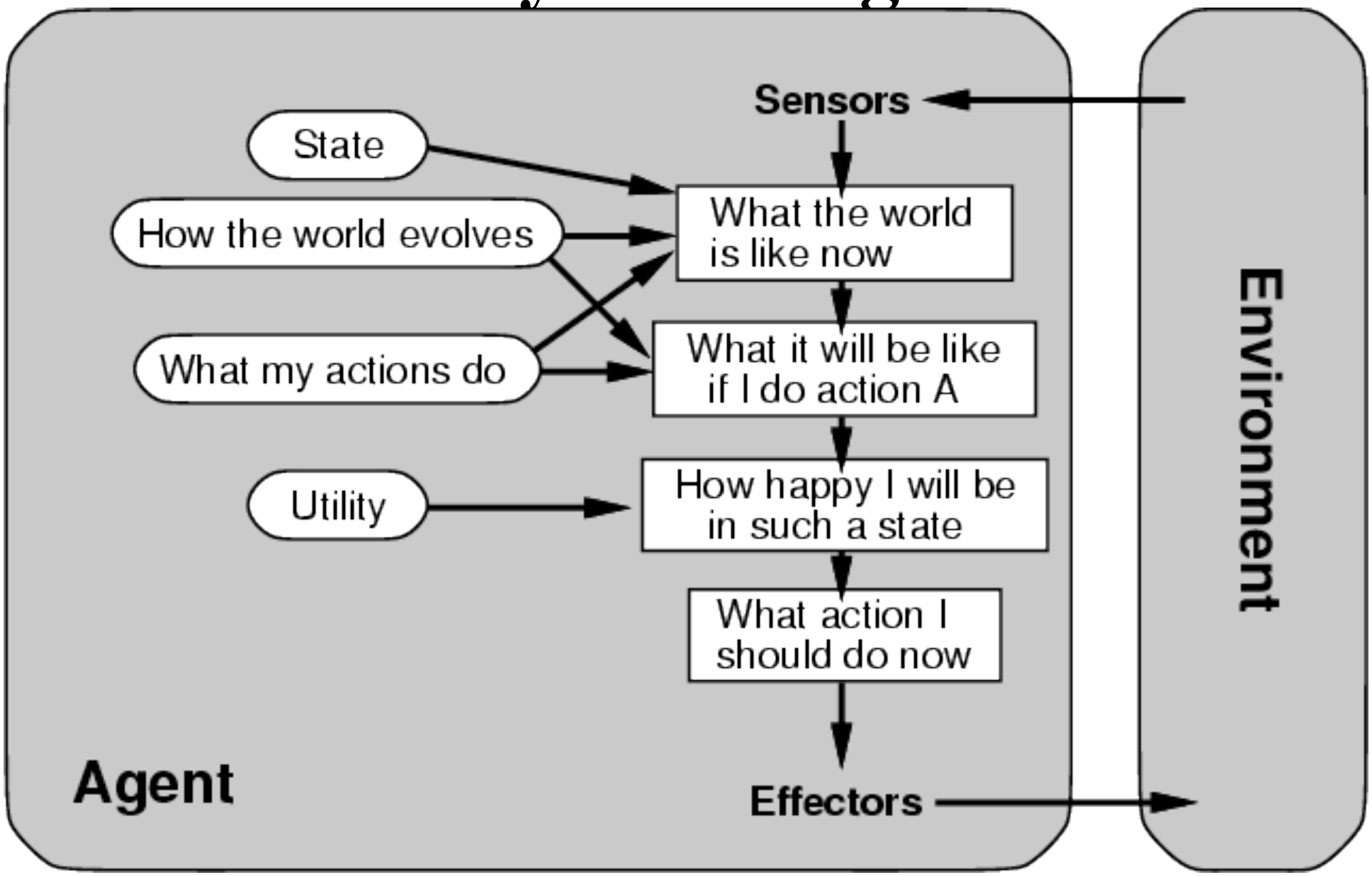
(3) Architecture for goal-based agent



(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.
- Utility function **U: State → Reals** indicating a measure of success or happiness when at a given state.

- Specifying utility as a function of the state of the environment (not the state of the agent) avoids loopholes in utility. (e.g., number of times dirt has been vacuumed vs. number of times dirt has been removed permanently)

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
  - In a fully observable, deterministic environment, the agent need not 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 (Star Trek).
  - In a **sequential** environment, the agent engages in a series of connected episodes (The Sopranos).
  - Such environments do not require the agent to plan ahead.

- **Static/Dynamic.**
  - A **static** environment does not change while the agent is thinking.

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)

Summary

- An **agent** perceives and acts in an environment, has an architecture, and is implemented by an agent program.
- 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.

Quick Overviews of Existing Agent Architectures

- SOAR
- BDI
- ACT-R
- Subsumption
- Society of Mind
- Memory Prediction Framework

SOAR

- Started 1983 by Laird, Newell, Rosenbloom.
- Based on a production system (roughly if-then rules)
- “An approximation of complete rationality”
- Used for cognitive modeling (modeling humans)
- Knowledge representation: procedural, declarative, episodic, and possibly iconic
- Representation of problem spaces, permanent knowledge, short term knowledge, generates sub-goals, learning via chunking, reinforcement learning, episodic learning, and semantic learning.
- Episodic and semantic memory.

http://en.wikipedia.org/wiki/SOAR

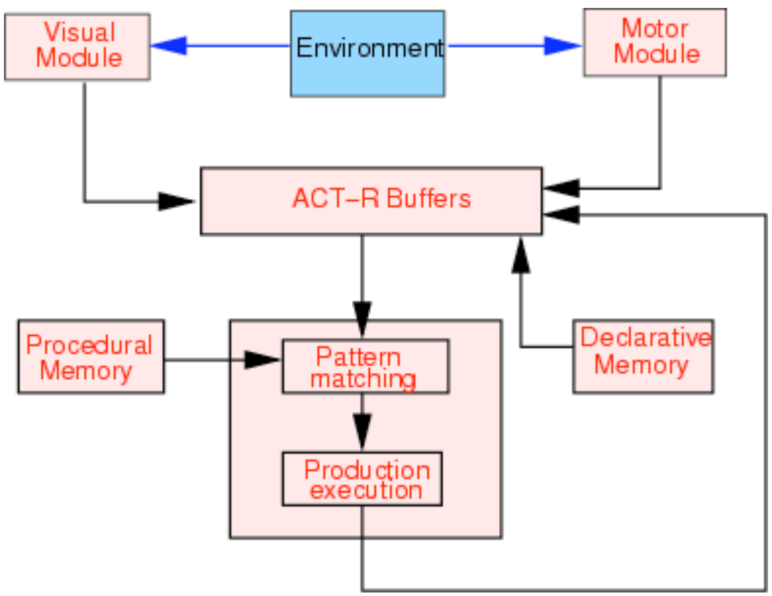
BDI

- Belief-Desire-Intention model by Bratman (Stanford)
- Beliefs: information state (knowledge and inferred beliefs through forward chaining)
- Desires: goals
- Intentions: what the agent has chosen to do
- Plans: sequences of actions (are nested and filled in as time progresses)

http://en.wikipedia.org/wiki/BDI_software_agent

ACT-R

- Theory of human cognition
- Knowledge is either declarative (facts) or procedural (how-to)
- Visual/motor modules are the interface to the real world
- Buffers access the modules
- Can only execute one production at a time
- Declarative is more of a parallel process



http://act-r.psy.cmu.edu/about/

Subsumption

- Rodney Brooks, 1986
- Break down complicated intelligent behavior into simple modules
- Organize these module into layers
- Low layers are fast acting reflexes while higher layers are goals

http://en.wikipedia.org/wiki/Subsumption_architecture

Society of Mind

- Written as a book, by Minsky
- Fundamental thinking entities as agents
- These agents form a society
- Agents have different processes and purposes
- Theory on how the mind works, not necessarily the brain.

http://en.wikipedia.org/wiki/Society_of_Mind_theory

Memory-prediction framework

- Jeff Hawkins (founder of palm) describes his framework in depth in the book On Intelligence
- Theory based on the human neocortex
- Explains how raw sensor input patters cause the brain to predict the future
- Attempts to be a unified theory on thinking (i.e., the same theory applies for vision, hearing) -- based on the observation that the neocortex is similar in structure everywhere
- Inputs start in a bottom up hierarchy and as higher nodes are activated, lower nodes are then activated. Input triggers low-level nodes, which trigger high-level nodes, which then trigger more low-level nodes to fill in a "picture" of the environment.

http://en.wikipedia.org/wiki/Memory-prediction_framework

Jeff Hawkins, Sandra Blakeslee (2004), *On Intelligence*

Other AGI systems

- OpenCog
- LIDA
- TexAI