AGENTS

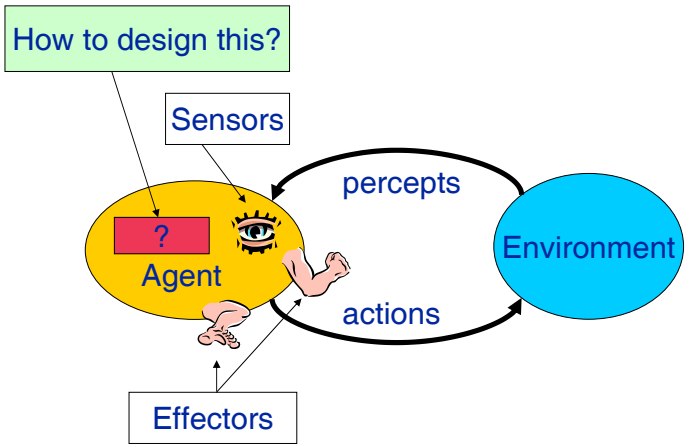
Outline

- PEAS (Performance, Environment, Actuators, Sensors)
- Environment Types
- Agent functions and properties
- Agent types

What is an Intelligent Agent?

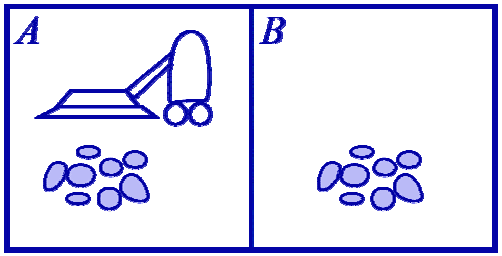
- **One definition:** An (intelligent) agent perceives it environment via sensors and acts rationally upon that environment with its effectors. Hence, an agent gets percepts one at a time, and maps this percept sequence to actions.
- **Another definition:** An agent is a computer software system whose main characteristics are situatedness, autonomy, adaptivity, and sociability.

Agents



Agents include humans, robots, softbots, thermostats, etc.

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

```
function REFLEX-VACUUM-AGENT([location,status]) returns an action
  if status = Dirty then return Suck
  else if location = A then return Right
  else if location = B then return Left
```

What is the **right** function?
Can it be implemented in a small agent program?

Agent Function and Program

- **Agent Function:** Mathematically speaking, we say that an agent's behavior is described by the agent function that maps any given percept sequence to an action.
- **Agent Program:** The implementation of the agent function for an artificial agent is called the agent program.

Rationality

- A rational agent is one that does the **right thing**.
- More precisely, what is rational at any given time depends on four things:
  - The performance measure that defines the criterion of success.
  - The agent's prior knowledge of the environment.
  - The actions that the agent can perform.
  - The agent's percept sequence to date.

Rational Agent

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

Performance Measure

- A performance measure embodies the criterion for success of an agent's behavior.

Task Environment

- **PEAS**
  - Performance,
  - Environment,
  - Actuators,
  - Sensors

PEAS for an Automated Taxi

- Performance Measure?
- Environment?
- Actuators?
- Sensors?

PEAS for an Automated Taxi

- Performance Measure? safety, destination, profits, legality, comfort, …
- Environment? streets/freeways, traffic, pedestrians, weather, …
- Actuators? steering, accelerator, brake, horn, speaker/display, …
- Sensors? video, accelerometers, gauges, engine sensors, keyboard, GPS, …

Agent Characteristics

- **Situatedness:** The agent receives some form of sensory input from its environment, and it performs some action that changes its environment in some way. Examples of environments: the physical world and the Internet.
- **Autonomy:** The agent can act without direct intervention by humans or other agents and that it has control over its own actions and internal state.

Agent Characteristics

- **Adaptivity :** The agent is capable of (1) reacting flexibly to changes in its environment; (2) taking goal-directed initiative (i.e., is pro-active), when appropriate; and (3) learning from its own experience, its environment, and interactions with others.
- **Sociability:** The agent is capable of interacting in a peer-to-peer manner with other agents or humans.

Environment Types

- **Fully observable** vs. **partially observable**.
  - If an agent's sensors give it access to the complete state of the environment at each point in time, then we say that the task environment is fully observable.
- **Deterministic** vs. **stochastic**.
  - If the next state of the environment is completely determined by the current state and the action executed by the agent, then we say the environment is deterministic; otherwise it is stochastic.

Environment Types

- Episodic vs. sequential.
  - In an episodic task environment, the agent's experience is divided into atomic "episodes."
  - Each episode consists of the agent perceiving and then performing a single action.
  - In sequential environments, the current decision may affect all future decisions.
- Static vs. dynamic.
  - If the environment can change while an agent is deliberating, then we say the environment is dynamic for that agent; otherwise it is static.
  - If the environment itself does not change with the passage of time but the agent's performance score does, then we say the environment is semidynamic.

Environment Types

- **Discrete** vs. **continuous**.
  - The discrete/continuous distinction can be applied to the *state* of the environment, to the way *time* is handled, and to the *percepts* and *actions* of the agent.
- **Single-agent** vs. **multi-agent**.
  - Competitive
  - Cooperative

Environment types

	Solitaire	Backgammon	Internet Shopping	Taxi
Observable??				
Deterministic??				
Episodic??				
Static??				
Discrete??				
Single-agent??				

Environment types

	Solitaire	Backgammon	Internet Shopping	Taxi
Observable??	Yes	Yes	No	No
Deterministic??				
Episodic??				
Static??				
Discrete??				
Single-agent??				

Environment types

	Solitaire	Backgammon	Internet Shopping	Taxi
Observable??	Yes	Yes	No	No
Deterministic??	Yes	No	Partly	No
Episodic??				
Static??				
Discrete??				
Single-agent??				

Environment types

	Solitaire	Backgammon	Internet Shopping	Taxi
Observable??	Yes	Yes	No	No
Deterministic??	Yes	No	Partly	No
Episodic??	No	No	No	No
Static??				
Discrete??				
Single-agent??				

Environment types

	Solitaire	Backgammon	Internet Shopping	Taxi
Observable??	Yes	Yes	No	No
Deterministic??	Yes	No	Partly	No
Episodic??	No	No	No	No
Static??	Yes	Semi	Semi	No
Discrete??				
Single-agent??				

Environment types

	Solitaire	Backgammon	Internet Shopping	Taxi
Observable??	Yes	Yes	No	No
Deterministic??	Yes	No	Partly	No
Episodic??	No	No	No	No
Static??	Yes	Semi	Semi	No
Discrete??	Yes	Yes	Yes	No
Single-agent??				

Environment types

	Solitaire	Backgammon	Internet Shopping	Taxi
Observable??	Yes	Yes	No	No
Deterministic??	Yes	No	Partly	No
Episodic??	No	No	No	No
Static??	Yes	Semi	Semi	No
Discrete??	Yes	Yes	Yes	No
Single-agent??	Yes	No	Yes (except auctions)	No

The environment type largely determines the agent design

The Structure of Agents

- *agent =architecture + program*

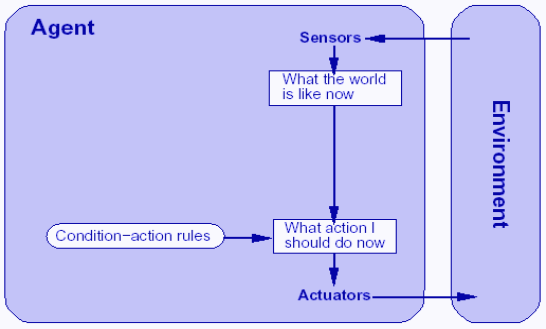
Agent Types

- Four basic types in order of increasing generality:
  - simple reflex agents
  - reflex agents with state
  - goal-based agents
  - utility-based agents
- All these can be turned into learning agents

Agent Functions and Programs

```
function TABLE-DRIVEN-AGENT( percept) returns an
  action
static:
  percepts, a sequence, initially empty
  table, a table of actions, indexed by percept
  sequences, initially fully specified
append percept to the end of percepts
action ←LOOKUP( percepts, table)
return action
```

Simple Reflex Agents

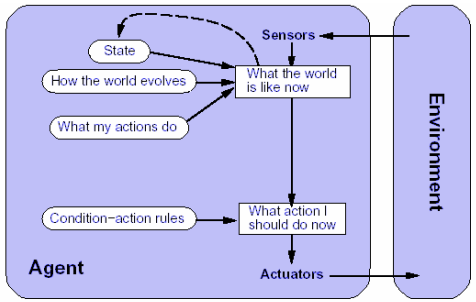


function SIMPLE-REFLEX-AGENT(*percept*) **returns** an action
  static: *rules*, a set of condition-action rules
  state ← INTERPRET-INPUT(*percept*)
  rule ← RULE-MATCH(*state, rules*)
  action ← RULE-ACTION[*rule*]
  return *action*

Simple Reflex Agent

- The simplest kind of agent is the **simple reflex agent**.
- These agents select actions based only on the *current* percept, ignoring the rest of the percept history.
- Table lookup of percept-action pairs defining all possible condition-action rules necessary to interact in an environment
- 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

Reflex agents with State



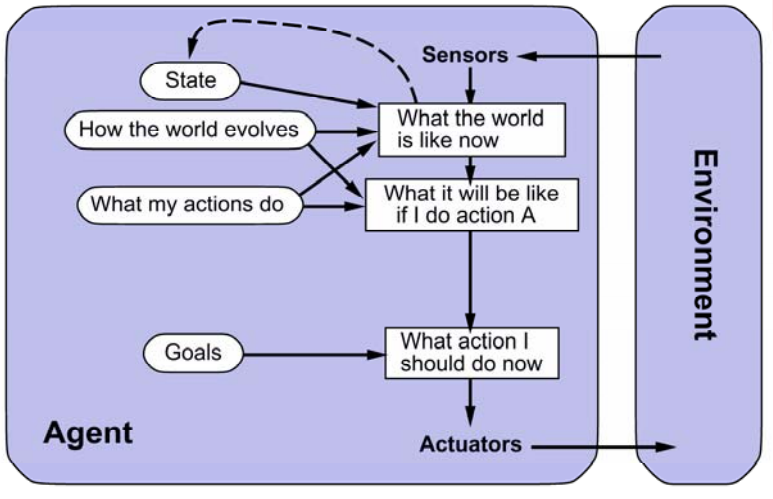
function REFLEX-AGENT-WITH-STATE(*percept*) **returns** an action
  static: *state*, a description of the current world state
    rules, a set of condition-action rules
    action, the most recent action, initially none

  state ← UPDATE-STATE(*state, action, percept*)
  rule ← RULE-MATCH(*state, rules*)
  action ← RULE-ACTION[*rule*]
  return *action*

Reflex Agent with State

- The knowledge about “how the world works” is called a model of the world.
- An agent that uses such a model is called a model-based agent.
- Encode "internal state" of the world to remember the past as contained in earlier percepts
- Needed because sensors do not usually give the entire state of the world at each input, so perception of the environment is captured over time. "State" used to encode different "world states" that generate the same immediate percept.
- 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

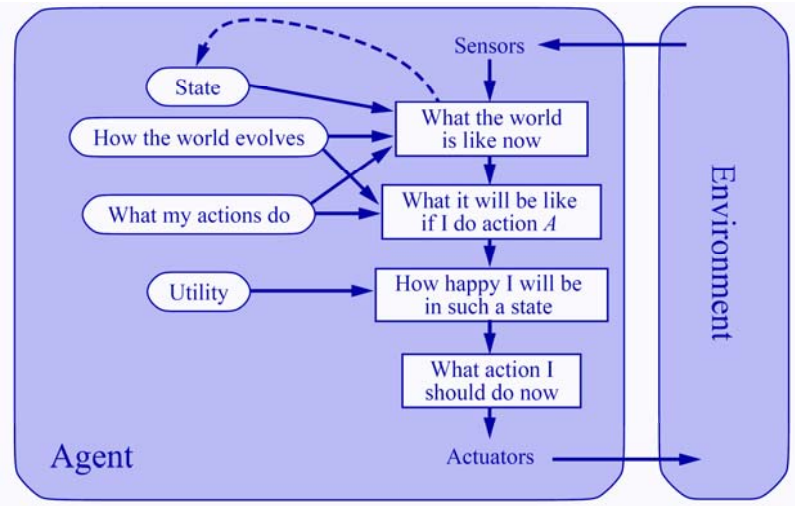
Goal Based Agents



Goal Based Agents

- Choose actions so as to achieve a (given or computed) goal= 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...?"

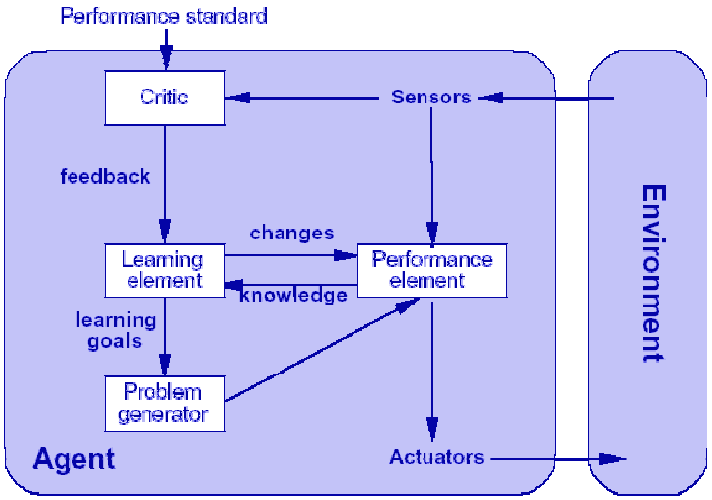
Utility Based Agents



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, but often need a more general performance measure that describes "degree of happiness"
- Utility function U: State --> Reals
  - indicates a measure of success or happiness when at a given state
  - Allows decisions comparing choice between conflicting goals, and choice between likelihood of success and importance of goal (if achievement is uncertain)

Learning agents



Components of a Learning Agent

- **learning element**, which is responsible for making improvements,
- **performance element**, which is responsible for selecting external actions. The performance element is the entire agent: it takes in percepts and decides on actions.
- **critic** gives feedback from the on how the agent is doing.
- **problem generator** is responsible for suggesting actions that will lead to new and informative experiences.