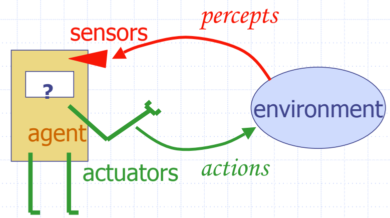
Intelligent Agents

Russell and Norvig: Chapter 2

CMSC421 – Fall 2003

Intelligent Agent

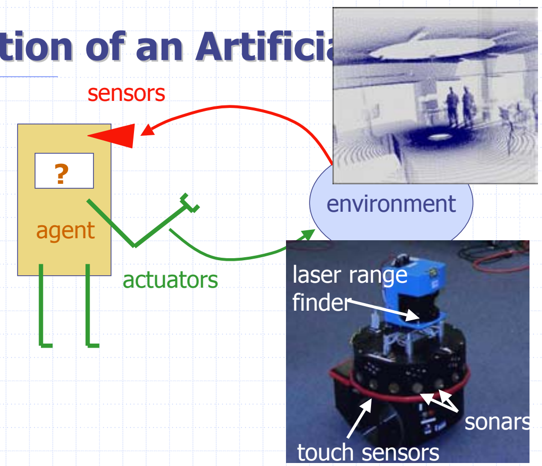


- Definition: An **intelligent agent** perceives its environment via **sensors** and acts rationally upon that environment with its **actuators.**

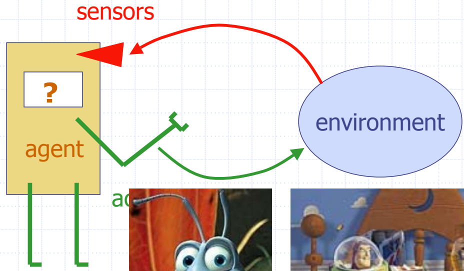
e.g., Humans

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

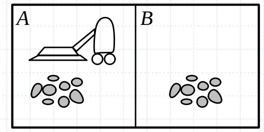
Notion of an Artifici



Notion of an Artificial Agent



Vacuum Cleaner World



Percepts: location and contents, e.g. [A, Dirty]
Actions: Left, Right, Suck, NoOp

Vacuum Agent Function

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
…	

Rational Agent

- What is rational depends on:
  - **P**erformance measure - The performance measure that defines the criterion of success
  - **E**nvironment - The agents prior knowledge of the environment
  - **A**ctuators - The actions that the agent can perform
  - **S**ensors - The agent's percept sequence to date
- We'll call all this the Task Environment (PEAS)

Vacuum Agent PEAS

- **Performance Measure**: minimize energy consumption, maximize dirt pick up. Making this precise: one point for each clean square over lifetime of 1000 steps.
- **Environment**: two squares, dirt distribution unknown, assume actions are deterministic and environment is static (clean squares stay clean)
- **Actuators**: Left, Right, Suck, NoOp
- **Sensors**: agent can perceive it's location and whether location is dirty

Automated taxi driving system

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

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

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 an accessible and deterministic environment, the agent need not deal with uncertainty.

Properties of Environments

- **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.
- **Discrete/Continuous.**
  - If the number of distinct percepts and actions is limited, the environment is **discrete**, otherwise it is **continuous**.

Environment Characteristics

	Fully Observable	Deterministic	Static	Discrete
Solitaire	No	Yes	Yes	Yes
Backgammon	Yes	No	Yes	Yes
Taxi driving	No	No	No	No
Internet shopping	No	No	No	No
Medical diagnosis	**No**	**No**	**No**	**No**

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

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) Model-based reflex agents**
  - have **internal state**, which is used to keep track of past states of the world.
- **(3) Goal-based agents**
  - 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.

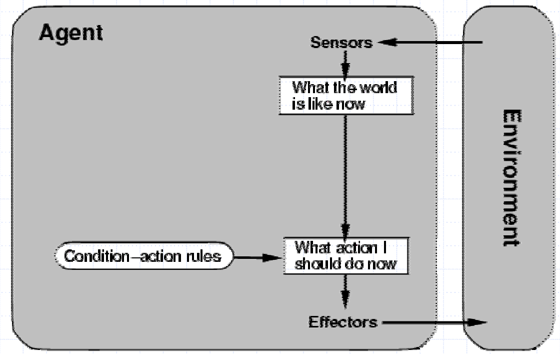
(0) Table-driven agents

- **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

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



Simple Vacuum Reflex Agent

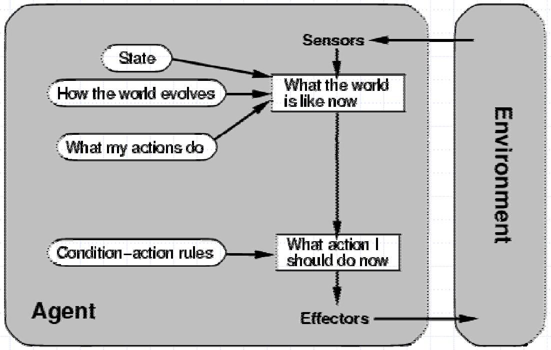
function Vacuum-Agent([location,status])
returns **Action**

if status = *Dirty* then return **Suck**
else if location = *A* then return **Right**
else if location = *B* then return **Left**

(2) Model-based reflex agents

- 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" is used to encode different "world states" that generate the same immediate percept.

(2)Model-based agent architecture

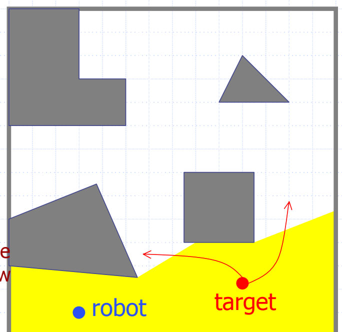


(3) Goal-based agents

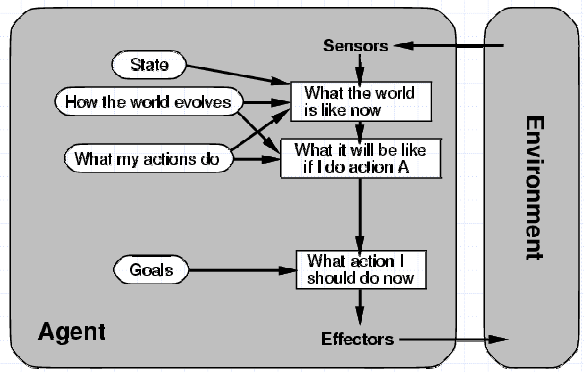
- Choose actions so as to achieve a (given or computed) goal.
- A goal is a description of a desirable situation.
- Keeping track of the current state is often not enough – need to add goals to decide which situations are good
- **Deliberative** instead of **reactive**.
- May have to consider long sequences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

Example: Tracking a Target

- The robot must keep the target in view
- The target's trajectory is not known in advance
- The robot may not know all the obstacles in advance
- Fast decision is required



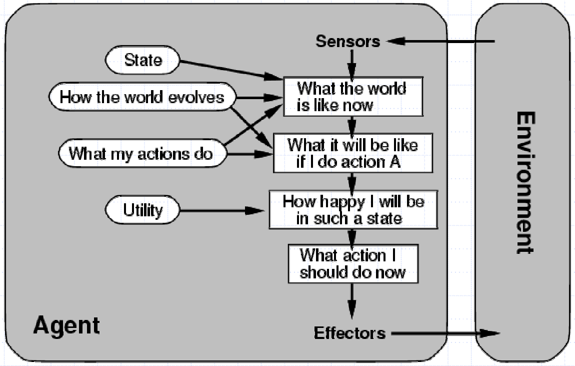
(3) Architecture for goal-based agent



(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, but often need a more general performance measure that describes "degree of happiness."
- Utility function **U: State → Reals** indicating 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).

(4) Architecture for a complete utility-based agent



Summary: Agents

- An **agent** perceives and acts in an environment, has an architecture, and is implemented by an agent program.
- Task environment – **PEAS (P**erformance**, E**nvironment, **A**ctuators, **S**ensors**)**
- 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 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 inaccessible, nondeterministic, dynamic, and continuous