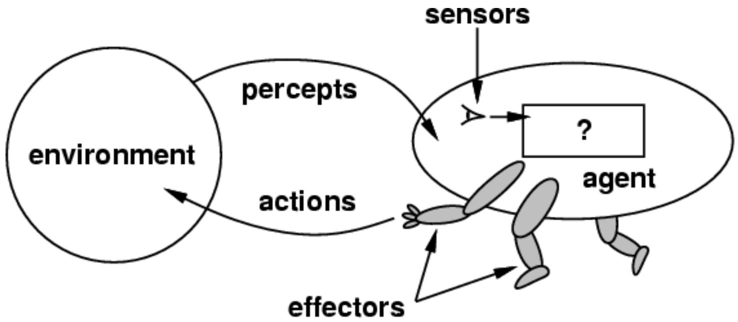
Intelligent Agents

- **Definition**: An Intelligent Agent perceives it environment via sensors and acts rationally upon that environment with its effectors (actuators).
- Hence, an agent gets percepts one at a time, and maps this percept sequence to actions.
- Properties
  - Autonomous
  - Interacts with other agents plus the environment
  - Reactive to the environment
  - Pro-active (goal- directed)



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

- Humans
  - Sensors: Eyes (vision), ears (hearing), skin (touch), tongue (gestation), 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, …
- 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!**

Challenge!!

Compare Software with an agent
Compare Human with an agent

Percept: The Agents perceptual inputs at any given instant.
Percept Sequence: The complete history of everything the agent has ever perceived.

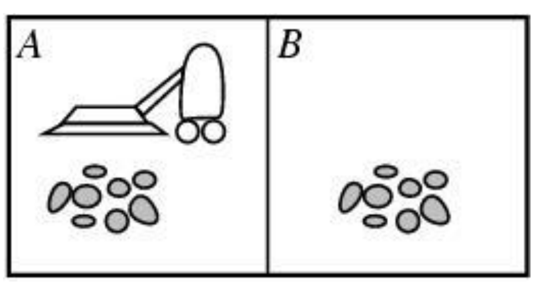
The *agent function* is mathematical concept that maps percept sequence to actions.

$$f : P^* \to A$$

The *agent function* will internally be represented by the *agent program.*

The agent program is concrete implementation of agent function it runs on the physical *architecture* to produce *f*.

The vacuum-cleaner world
Example of Agent



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

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**
…	**…**

The concept of rationality

A **rational agent** is one that does the right thing.

- Every entry in the table is filled out correctly.

What is the right thing?

- Right action is the one that will cause the agent to be most successful.

Therefore we need some way to measure success of an agent.

Performance measures are the criterion for success of an agent behavior. For the Vacuum cleaner world performance measure may be

- The amount of dirt cleaned within a certain time or
- How clean the floor is.

It is better to design Performance measure according to what is wanted in the environment instead of how the agents should behave.

It is not easy task to choose the performance measure of an agent. For example if the performance measure for automated vacuum cleaner is "The amount of dirt cleaned within a certain time" Then a ration agent can maximize this performance by cleaning up the dirt , then dumping it all on the floor, then cleaning it up again , and so on. Therefore "How clean the floor is" is better choice for performance measure of vacuum cleaner?

What is rational at a given time depends on four things:

- Performance measure,
- Prior environment knowledge,
- Actions,
- Percept sequence to date (sensors).

DEF^n: *A rational agent chooses whichever action maximizes the expected value of the performance measure given the percept sequence to date and prior environment knowledge.*

Rationality ≠ omniscience

- An omniscient agent knows the actual outcome of its actions.

Rationality ≠ perfection

– Rationality maximizes *expected* performance, while perfection maximizes *actual* performance.

Environments

To design a rational agent we must specify its task environment. Task environment means: PEAS description of the environment:

– Performance
– Environment
– Actuators
– Sensors

E.g. Fully automated taxi:

– PEAS description of the environment:
  Performance
    » Safety, destination, profits, legality, comfort
  Environment
    » Streets/freeways, other traffic, pedestrians, weather,, …
  Actuators
    » Steering, accelerating, brake, horn, speaker/display,…
  Sensors
    » Video, sonar, speedometer, engine sensors, keyboard, GPS, …

Structure of intelligent agent

Agent Type	Percepts	Actions	Goals	Environment
Medical diagnosis system	Symptoms, findings, patient's answers	Questions, tests, treatments	Healthy patient, minimize costs	Patient, hospital
Satellite image analysis system	Pixels of varying intensity, color	Print a categorization of scene	Correct categorization	Images from orbiting satellite
Part-picking robot	Pixels of varying intensity	Pick up parts and sort into bins	Place parts in correct bins	Conveyor belt with parts
Refinery controller	Temperature, pressure readings	Open, close valves; adjust temperature	Maximize purity, yield, safety	Refinery
Interactive English tutor	Typed words	Print exercises, suggestions, corrections	Maximize student's score on test	Set of students

Figure 2.3 Examples of agent types and their PAGE descriptions.

Environment types

Accessible/ Inaccessible (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 accessible.

– Such environments are convenient, since the agent is freed from the task of keeping track of the changes in the environment.

• **Deterministic/ Non-deterministi**c

– 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.

• **Episodic/ Nonepisodi**c.

– An episodic environment means that subsequent episodes do not depend on what actions occurred in previous episodes.

– Such environments do not require the agent to plan ahead.

Static/ Dynamic.

– A static environment does not change while the agent is thinking.

– In a static environment the agent need not worry about the passage of time while he is thinking, nor does he have to observe the world while he is thinking.

– In static environments the time it takes to compute a good strategy does not matter.

• **Discrete/ Continuous.**

– If the number of distinct percepts and actions is limited the environment is discrete, otherwise it is continuous.

With/ Without rational adversaries (Single Agent/ Multi-Agent)

– If an environment does not contain other rationally thinking, adversary agents, the agent need not worry about strategic, game theoretic aspects of the environment

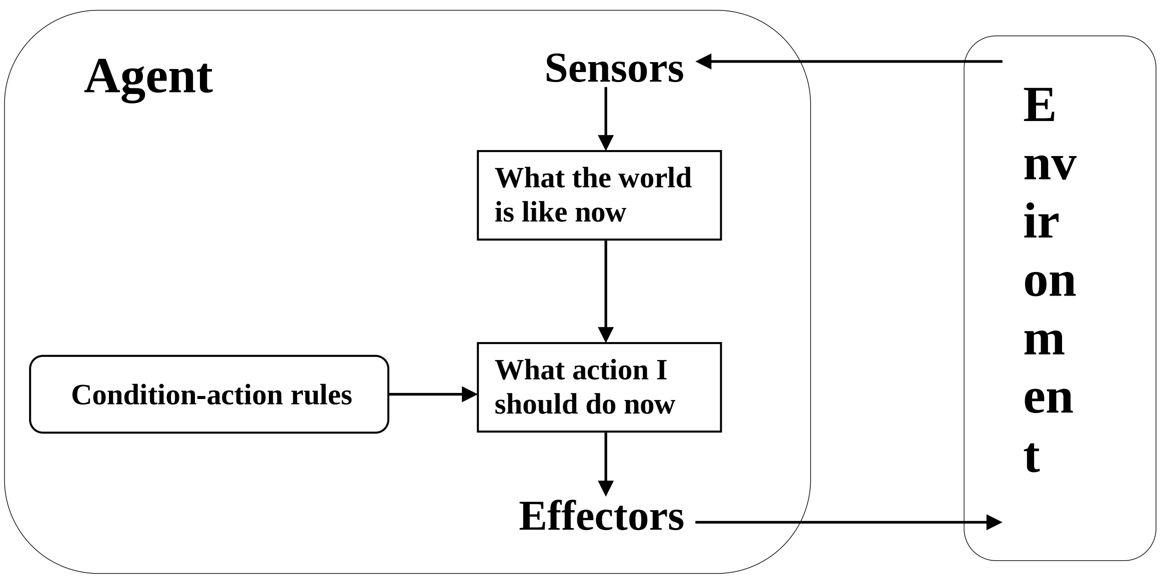
– Most engineering environments are without rational adversaries, whereas most social and economic systems get their complexity from the interactions of (more or less) rational agents.

–

Types of Agent

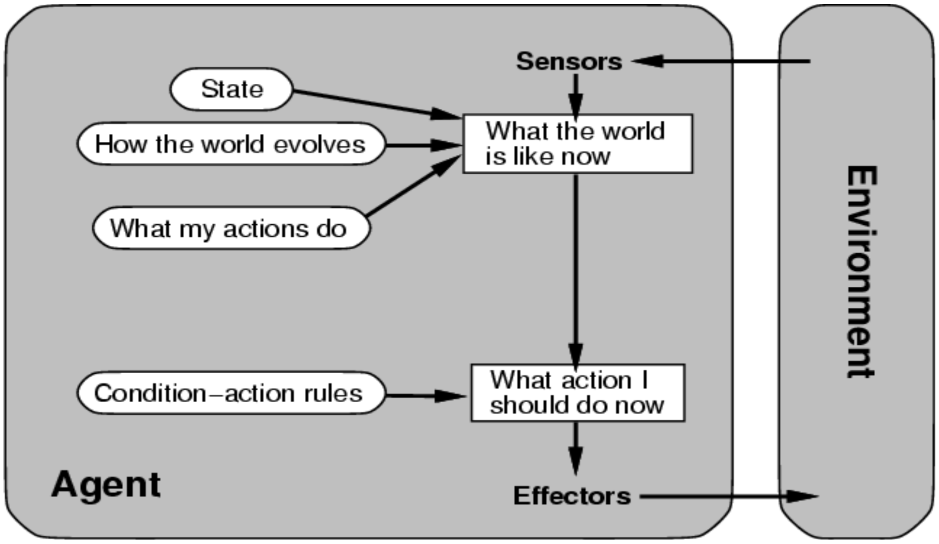
Simple Reflex Agent

- **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
- Use *condition-action* rules to summarize portions of the table



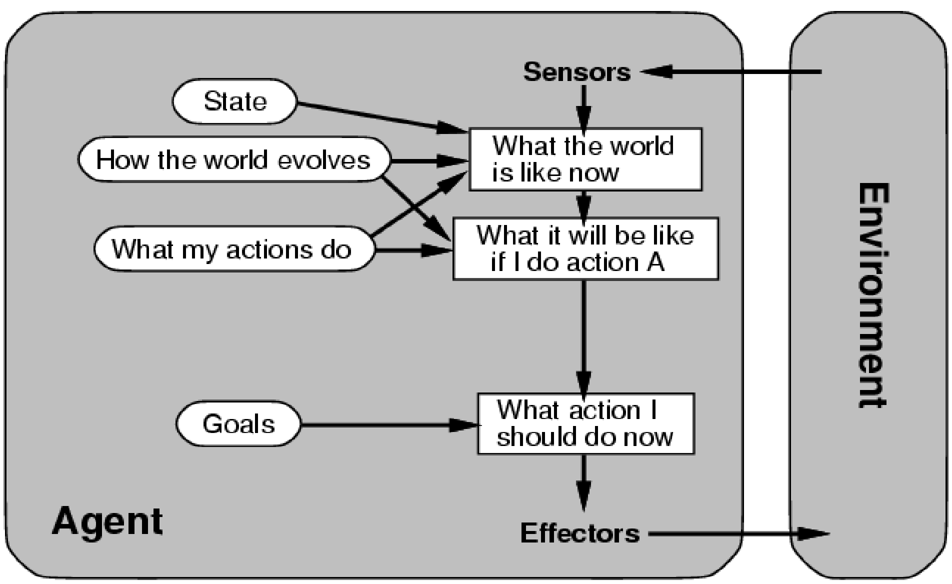
Model Based Agents (Reflex Agent with Internal State)

- 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 Agent

• 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...?"*



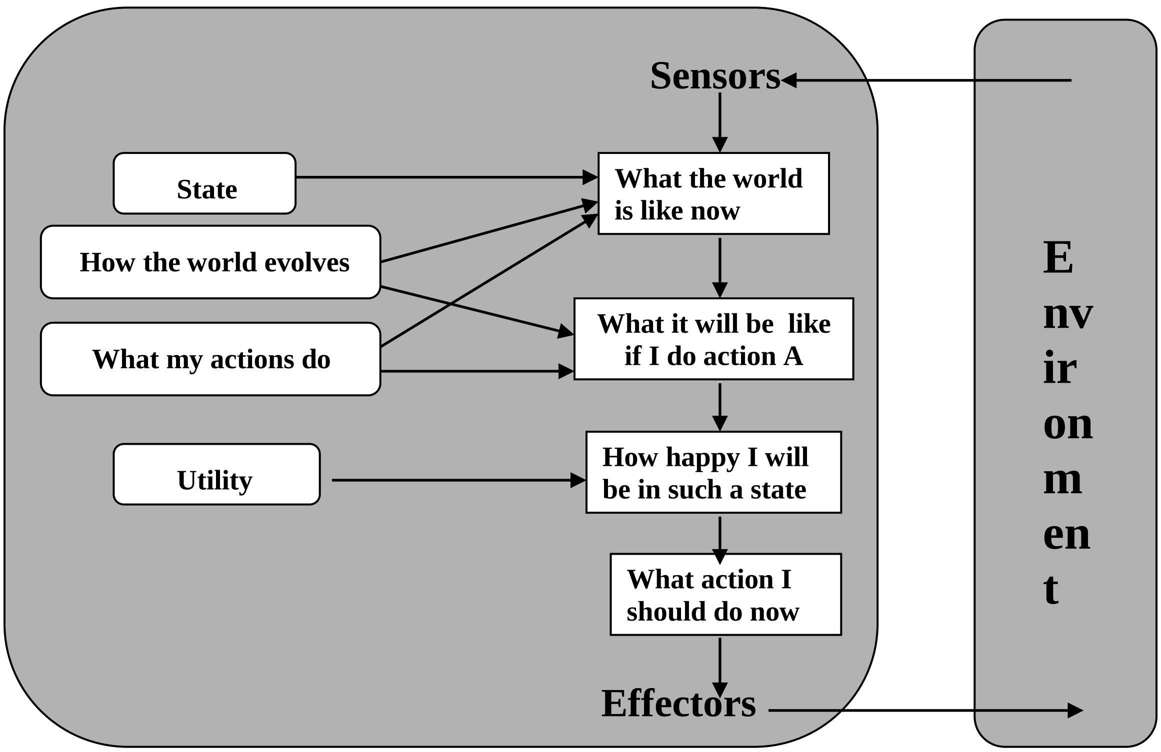
Utility- Based Agent

• 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: States --> 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)



===Simple reflex agents===

Simple reflex agents act only on the basis of the current percept, ignoring the rest of the percept history. The agent function is based on the ''condition-action rule'': if condition then action.

This agent function only succeeds when the environment is fully observable. Some reflex agents can also contain information on their current state which allows them to disregard conditions whose actuators are already triggered.

===Model-based reflex agents===

A model-based agent can handle a partially observable environment. Its current state is stored inside the agent maintaining some kind of structure which describes the part of the world which cannot be seen. This knowledge about "how the world works" is called a model of the world, hence the name "model-based agent".

A model-based reflex agent should maintain some sort of [[Mental model |internal model]] that depends on the percept history and thereby reflects at least some of the unobserved aspects of the current state. It then chooses an action in the same way as the reflex agent.

===Goal-based agents===

Goal-based agents further expand on the capabilities of the model-based agents, by using "goal" information. Goal information describes situations that are desirable. This allows the agent a way to choose among multiple possibilities, selecting the one which reaches a goal state. Search and planning are the subfields of artificial intelligence devoted to finding action sequences that achieve the agent's goals.

In some instances the goal-based agent appears to be less efficient; it is more flexible because the knowledge that supports its decisions is represented explicitly and can be modified.

===Utility-based agents===

Goal-based agents only distinguish between goal states and non-goal states. It is possible to define a measure of how desirable a particular state is. This measure can be obtained through the use of a ''utility function'' which maps a state to a measure of the utility of the state. A more general performance measure should allow a comparison of different world states according to exactly how happy they would make the agent. The term utility, can be used to describe how "happy" the agent is.

A rational utility-based agent chooses the action that maximizes the expected utility of the action outcomes- that is, the agent expects to derive, on average, given the probabilities and utilities of each outcome. A utility-based agent has to model and keep track of its environment, tasks that have involved a great deal of research on perception, representation, reasoning, and learning.

===Learning agents===

Learning has an advantage that it allows the agents to initially operate in unknown environments and to become more competent than its initial knowledge alone might allow. The most important distinction is between the "learning element", which is responsible for making improvements, and the "performance element", which is responsible for selecting external actions.

The learning element uses feedback from the "critic" on how the agent is doing and determines how the performance element should be modified to do better in the future.

The performance element is what we have previously considered to be the entire agent: it takes in percepts and decides on actions.

The last component of the learning agent is the "problem generator". It is responsible for suggesting actions that will lead to new and informative experiences.