Tecnológico de Monterrey

CS4000 Intelligent Systems

Intelligent Agents

Content

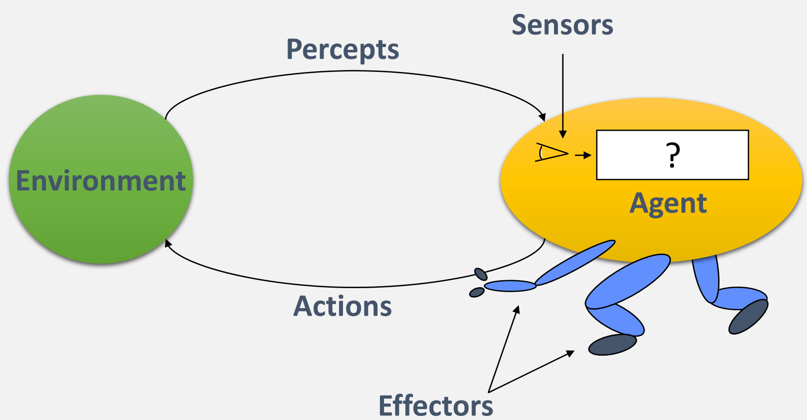
- What is an intelligent agent?
- Agent types
- Task environments
- Structure of an intelligent agent
- Conclusions

What is an intelligent agent?

What is an agent?

- An **agent** is something that perceives and acts.
- Agents have:
  - **Sensors** with which they perceive **percepts**.
  - **Effectors** with which they perform **actions**.
- Rational agents act in a **correct way**.
  - They infer correctly.
  - They act reflexively.

What is an agent?



Agent Behavior

- Described externally through the **agent function.**
  - The agent function is an abstract mathematical description.
  - **Input**: Percepts sequence.
  - **Output**: Action to execute.

- Described internally through the **agent program.**
  - The agent program is a concrete implementation running within some physical system.
  - **Input**: Current perception.
  - **Output**: Action to execute.

Rational Agent

- **Definition** of a rational agent (*Russel and Norvig*):

*"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."*

- **Rationality** 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 Behavior

- Rational vs Omniscient
  - Expected performance vs perfect performance

- Rational actions:
  - Obtain information before acting.
  - Learn from what is perceived.

- A rational agent should be autonomous.
  - It should learn to compensate for failures in its prior knowledge (missing or incorrect).

Task Environments

- "Problems" for which rational agents are the "solution".
- Affect the design of the agent program.
- **PEAS** Specification:
  - **P**erformance measure
  - **E**nvironment
  - **A**ctuators
  - **S**ensors

Agent Types

Agent Type	Performance Measure	Environment	Actuators	Sensors	
Medical diagnosis systems	Healthy patient, reduced costs	Patient, hospital, staff	Display of questions, tests, diagnoses, treatments, referrals	Keyboard entry of symptoms, findings, patient's answers	
Satellite image analysis system	Correct image categorization	Downlink from orbiting satellite	Display of scene categorization	Color pixel arrays	
Part-picking robot	Percentage of parts in correct bins	Conveyor belt with parts; bins	Jointed arm and hand.	Camera, joint angle sensors	
Refinery controller	Purity, yield safety	Refinery, operators	Valves, pumps, heaters, displays	Temperature, pressure, chemical sensors	
Interactive English tutor	Student's score on test	Set of students, testing agency	Display of exercises, suggestions, corrections	Keyboard entry	

Task Environments

Task Environments

- The agents operate within an environment:
  - Robots feel the real world.
  - Simulators can provide environmental information to the sensors.
- Environment properties:
  - Fully vs. **Partially** Observable.
  - Deterministic vs. **Stochastic.**
  - Episodic vs. **Sequential.**
  - Static vs. **Dynamic.**
  - Discrete vs. **Continuous.**
  - Single agent vs. **Multi-agent.**

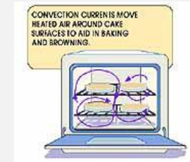
Fully vs. Partially Observable

- **Fully observable** means that the agent sensors give all relevant information to select the action.
- Fully observable environments in games.
- **Partially observable** environments in process control systems.

Deterministic vs. Stochastic

- An environment is **deterministic** if the next state of the environment is 100% determined by the current state and the selected action, otherwise it is **stochastic.**
- An environment is **strategic** when it is deterministic except for the actions of other agents.
- The environments in games like the chess are strategic, not so in games of chance.

Episodic vs. Sequential



- An episode is a perception-action pair.
- An environment is **episodic** when the quality of what happens in subsequent episodes does not depend on what happened in past episodes; otherwise it is considered **sequential**.
- Usually, games do not have episodic environments, but a robot that separates bad parts from good ones through computer vision has it.

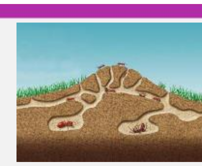
Static vs. Dynamic

- A **static** environment is one that does not change while the agent determines the action to be performed.
- If the environment does not change but does the performance of the agent as a function of time, we speak of a **semi-dynamic** environment.
- The taxi driver is not driven in a static environment. On the contrary, an agent that plays chess does so.

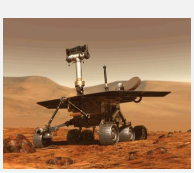
Discrete vs. Continuous

- Applied to the state, to the way in which time is handled and to the percepts and actions of the agent.
- For example, if the number of percepts and actions is finite then the environment is **discrete**; if not, it is **continuous**.
- The game of chess is given in a discrete environment (states, perceptions and actions); on the other hand, the taxi driver works in a continuous environment.

Single Agent vs. Multi-agent



- It depends on how the problem is conceptualized and on the number of agents involved in such conceptualization (if successes in performance measures are affected by other agents).
- The taxi driver includes multiple agents. A system for assigning classrooms to groups could have only one agent.
- **Concepts involved**: competition, negotiation, cooperation, coordination.



Examples of environments

Environment	Observable?	Deterministic?	Episodic?	Static?	Discrete?	Agents
Crossword	Fully	Deterministic	Sequential	Static	Discrete	Simple
Chess with a clock	Fully	Deterministic	Sequential	Semi	Discrete	Multi
Poker	Partially	Stochastic	Sequential	Static	Discrete	Multi
Backgammon	Fully	Stochastic	Sequential	Static	Discrete	Multi
Taxi driver	Partially	Stochastic	Sequential	Dynamic	Continuous	Multi
Medical diagnostic	Partially	Stochastic	Sequential	Dynamic	Continuous	Simple
Image Analyzer	Fully	Deterministic	Episodic	Semi	Continuous	Simple
Robot picker parts	Partially	Stochastic	Episodic	Dynamic	Continuous	Simple
Driver refinery	Partially	Stochastic	Sequential	Dynamic	Continuous	Simple
Interactive English tutor	Partially	Stochastic	Sequential	Dynamic	Discrete	Multi

Structure of Agents

Structure of Intelligent Agents

- **Agent = Architecture + Program.**
- **Architecture:** Computational device running the program and implementing the interface between the program and the environment.
- ROBOT: has sensors and actuators.
- Softbot: has sensors (simulated, artificial, extended) and actuators.
- The main feature is that the agent is located in an environment where it can detect events and produce changes.

A Table-driven Agent

- Given a perception, it simply looks for the answer.
- It seems simple, but there are some problems such as:
  - Combinatorial explosion.
  - At the beginning, it is difficult to build a table.
- Learning based in the Atkeson Memory can help through the use of incomplete tables and interpolation.

Table-driven Agent

function TABLE-DRIVEN-AGENT (*percept*) returns *action*
persistent: *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*

Basic Agent Types

- Simple reflex agents.
- Model-based reflex agents.
- Goal-based agents.
- Utility-based agents.
- Learning agents.

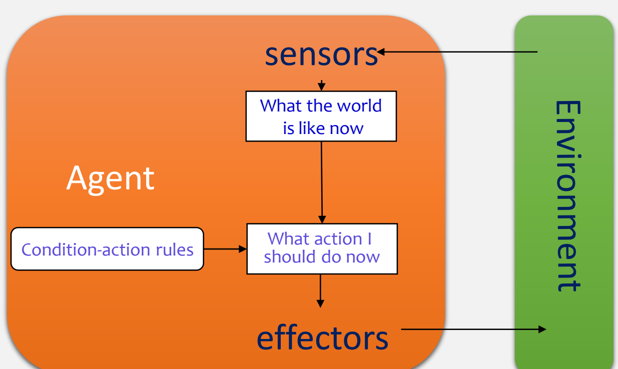
Purely Reflex Agents

- They do not refer to its history: $action: S \rightarrow A$

Examples:
ants, bees, etc.
simple robots
thermostat

action (s) = heater on if s.temperatura = COLD
heater off otherwise

Simple Reflex Agent



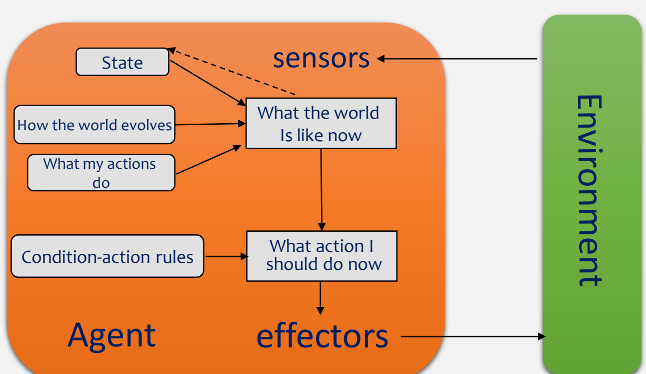
Simple Reflex Agent

function SIMPLE-REFLEX-AGENT (percept) **returns** *action*

persistent: *rules*, a set of condition-action rules

state ← INTERPRET-INPUT (*percept*)
rule ← RULE-MATCH (*state, rules*)
action ← *rule*.ACTION
return *action*

Model-based Reflex Agents



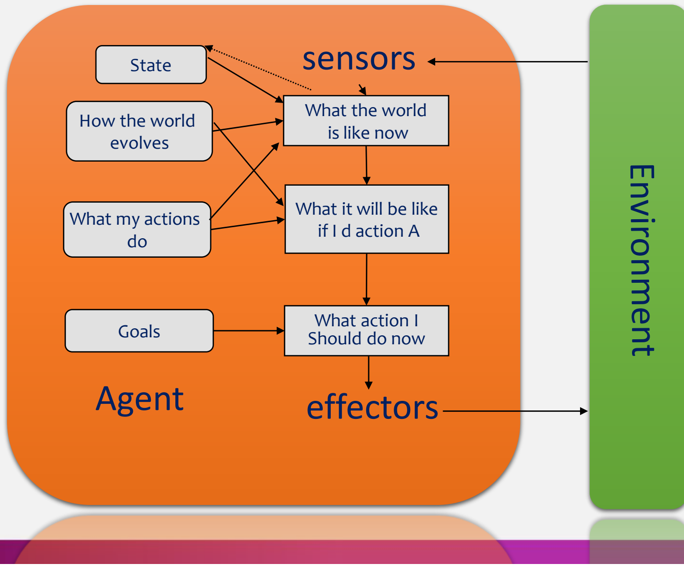
Model-based Reflex Agent

Function MODEL-BASED-REFLEX-AGENT (percept) **returns** action
  persistent: *state,* the agent's current conception of the world state
    model, a description of how the next extate dependes on current state and action
    rules, s set of condition-action rules
    action, the most recent action, initially none
  state <- UPDATE-STATE (state, action, percept, model)
  rule <- RULE-MATCH (state, rules)
  action <- rule.ACTION
  return action

Goal-based Agent

- The goal reflects the wishes of the agent.
- It may involve a projection of actions to determine consistency with the goals.
- Search and problem solvers can take this form.

Goal-based Agent



Utility-based Agent

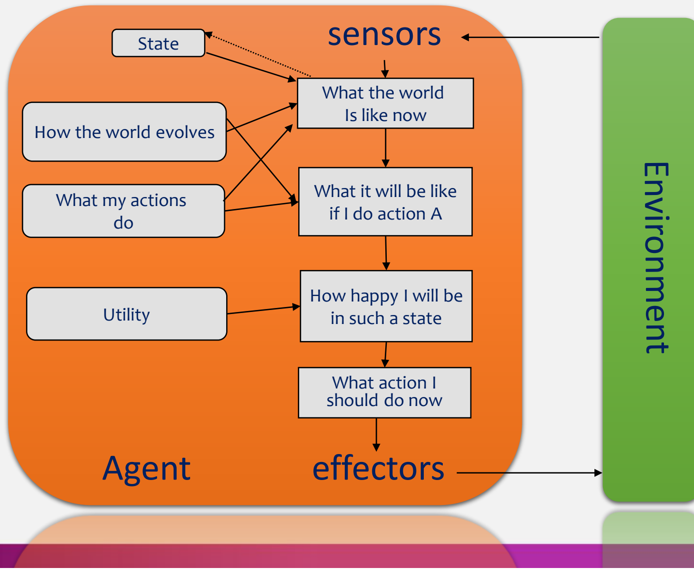
- The evaluation function is used to measure utility.

$$f\,(\text{state}) \longrightarrow \text{real number}$$

- It is useful to evaluate competing goals and to guide search.

- Game programs fall into this category.

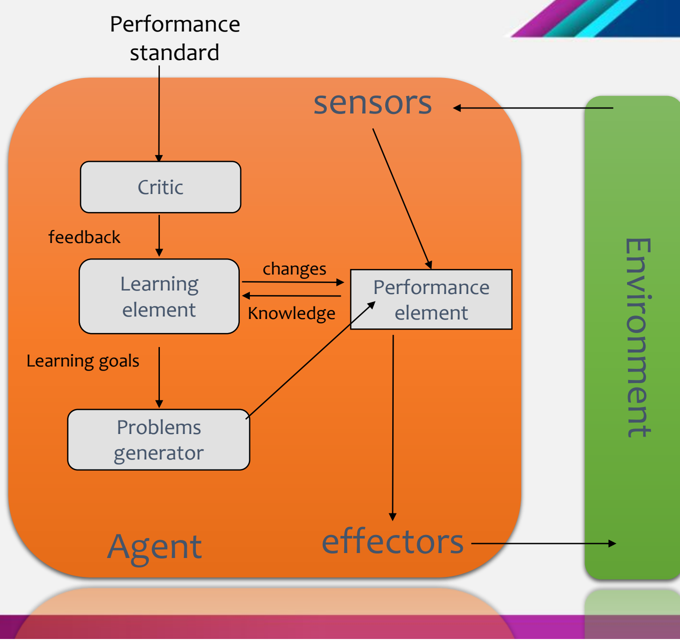
Utility-based Agent



Learning Agents

- Performance element
  - Decide what actions will be carried out.
- Critic
  - It provides feedback on how the agent is doing.
- Learning element
  - Uses critical feedback and modifies the performance element to take better decisions in the future.
- Problem generator
  - Suggests actions to guide to new informative experiences.

Learning Agent



Example of Learning Agent

- Taxi Agent
  - The agent leaves to the road and drives using its **performance element**.
  - The **critic** observes the world and passes information to the learning element, for example, expressions of disgust when the driver changes lanes without warning.
  - The **learning element** creates new rules, for example to avoid changing lanes without warning.
  - The **problem generator** identifies potential areas for improvement and suggests experiments, such as testing the brakes on different surfaces under different conditions.

Tecnológico de Monterrey