U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

IART - Artificial Intelligence
Lecture 1: Intelligent Agents

Luís Paulo Reis

lpreis@fe.up.pt

Director of LIACC – Artificial Intelligence and Computer Science Lab.
Associate Professor at DEI/FEUP – Informatics Engineering Department,
Faculty of Engineering of the University of Porto, Portugal
President of APPIA – Portuguese Association for Artificial Intelligence

Presentation Structure

- Agent Concept
- Intelligent Agents - Rational Agents
- Properties of Environments
- Intelligent Agents Architectures
  - Simple Reflex Agents
  - Agents with Representation of the World
  - Objective-Based Agents
  - Utility-Based Agents
  - Learning Agents
  - BDI Agents
- Multi-Agent Systems

Inteligent Agents

- An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators

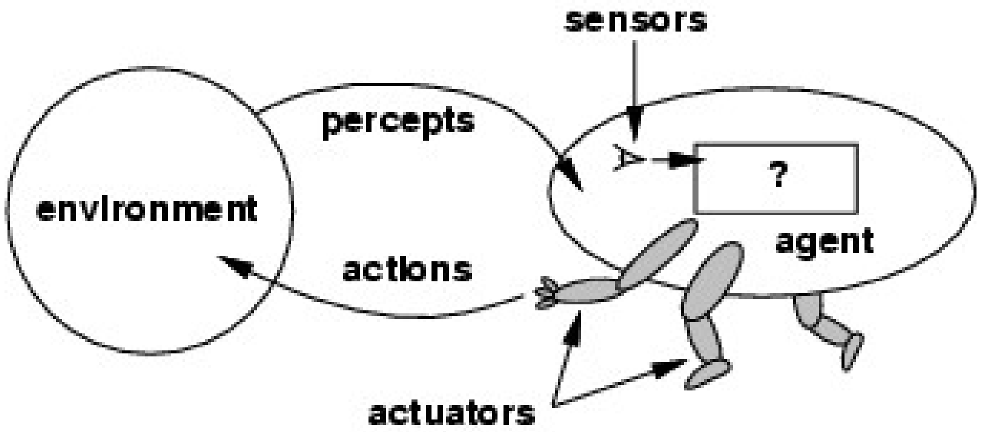
- Human:
  - Sensors: Eyes, ears, nose, touch, taste, others
  - Actuators: Legs, arms, hands, other body parts
- Robotic agent:
  - Sensors: cameras, infrared sensors, range finders, microphone, etc.
  - Actuators: motors, wheels, speaker, etc.

Any given instant can depend on the entire percept sequence observed to date, but not on anything it hasn't perceived

Intelligent Agents - Rational Agents

- **Rational Agent** is the one who does the right/correct thing!
- What's the correct thing/correct action?
  - The one that makes the agent more successful! E.g.
    - Robot Driver?
    - Chess Playing Program?
    - Spam email classifier?
- How and when to **evaluate this success**? (measure of success)
- Ideal Rational Agent:
  - "For each sequence of perceptions, it does the action that is expected to maximize its measure of performance (success), given the knowledge it has!"
  - Given the percept sequence
  - Given the agent's built in knowledge
- Mapping between perceptions and actions!
- **Rationality is distinct from omniscience** (all-knowing with infinite knowledge

Intelligent Agent



The agent function maps the history of perceptions into actions:

$$[f: \mathcal{P}^* \rightarrow \mathcal{A}]$$

The agent program runs on the physical architecture to produce f

Agent = Architecture + Program

The agent function is an abstract mathematical description

The agent program is a specify implementation, running within some physical system

Structure of Intelligent Agents

- Agent displays a behaviour - action that is performed after a given sequence of perceptions!
- AI Task: Design the Program and Architecture for the Agent
- What's an agent?

  Agent = Architecture + Program
- Software Agents vs Physical Agents

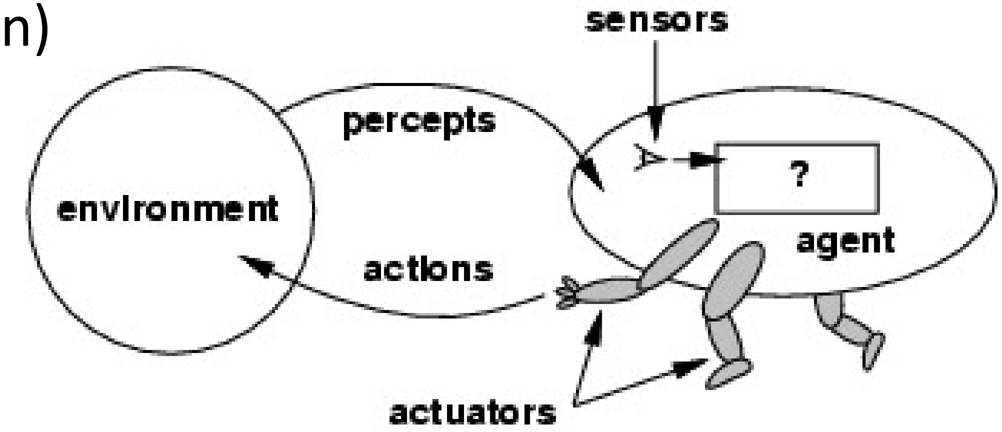
Agent Requisites

- Traditional definition include to much or leaves “holes”!
- Requisites:
  - Perceive its environment (sensors)
  - Decide actions to execute (“think”)
  - Execute actions in environment using its actuators
  - Communicate?
  - Perform a complex function?
- Agents vs Objects:
  - Agents decide what to do
  - Object methods are called externally
  - Agents react to sensors and control actuators

“Objects do it for free! Agents do it for money (or because they want to)!”

Agent Program / Agent Types

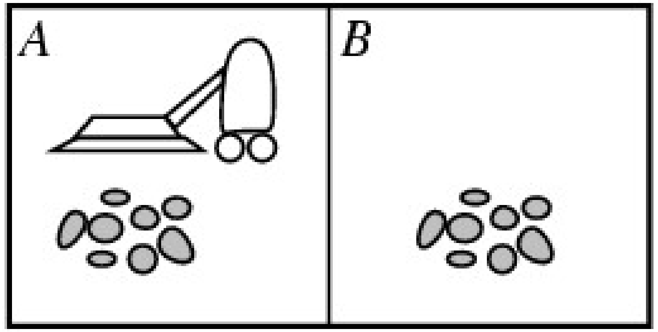
- Internal Data Structures are updated using perceptions and used to make the decision of the actions to be performed (best action)



Types of Agents (Russell and Norvig):

- Simple reflex agents
- Agents representing the world
- Objective-based agents
- Utility-based agents
- Learning Agents

Example: Vacuum cleaner World



- Perceptions: Place and content
  - Example: [A, Dirty]
- Actions: Left, Right, Suck, NoOp
- Simple Agent function:
  - If the current state is Dirty, then suck otherwise move to the other square

The Nature of Environments: PEAS

- When designing an agent, the first task is to characterize its task environment and the task
- PEAS: Performance measure, Environment, Actuators, Sensors
- Must first specify the setting for intelligent agent design
- Consider the task of designing an agent for a specific task:
  - Performance Measure
  - Environment
  - Actuators
  - Sensors

PEAS – Taxi Driver Agent

Agent: Taxi Driver Agent

- **Performance measure:** Safe, fast, legal, comfortable trip, maximize profits
- **Environment:** Roads, other traffic, pedestrians, customers
- **Actuators:** Steering wheel, accelerator, brake, signal, horn
- **Sensors:** Cameras, sonar, speedometer, GPS, odometer, engine sensors, keyboard

PEAS – Medical Diagnosis System

Agent: Medical diagnosis system

- **Performance measure:** Healthy patient, minimize costs, lawsuits
- **Environment:** Patient, hospital, staff
- **Actuators:** Screen display (questions, tests, diagnoses, treatments, referrals)
- **Sensors:** Keyboard (entry of symptoms, findings, patient's answers)

PEAS – Part Picking Robot

Agent: Part-picking robot

- **Performance measure:** Percentage of parts in correct bins
- **Environment:** Conveyor belt with parts, bins
- **Actuators:** Jointed arm and hand
- **Sensors:** Camera, joint angle sensors

Properties of Environments

- **Accessible vs Inaccessible**
  - Accessible if the agent sensors detect everything that is relevant in the environment!
- **Deterministic vs Non Deterministic**
  - Deterministic if the next state is determined by the previous state and the agent's actions!
- **Episodic vs Non Episodic**
  - Divided into episodes! Subsequent episodes do not rely on actions in previous episodes!
- **Static vs Dinamic**
  - Dynamic if it changes while the agent is thinking!
- **Discrete vs Continuos**
  - Discrete if there is a finite number of perceptions and actions!
- **Single Agent (versus multi-agent)**
  - A single agent operating alone in the environment vs multi-agents in the environment (cooperative or competitive)

Environment Types

- The environment type has a large influence on the agent design
- The simplest environment is
  - Fully observable, deterministic, episodic, static, discrete and single-agent.
- Most real situations are:
  - Partially observable, non-deterministic (stochastic), sequential, dynamic, continuous and multi-agent

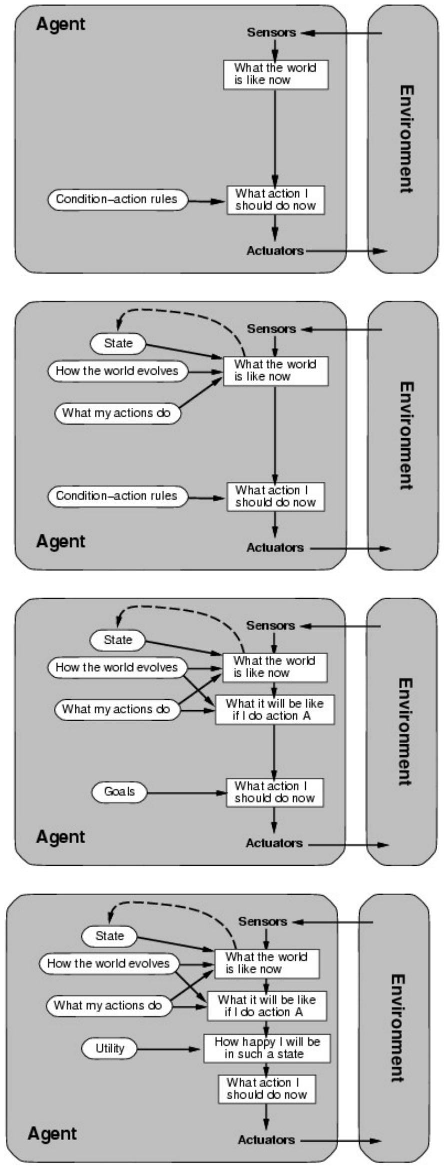
Properties of Environments

	Chess with Clock	Chess Without Clock	Taxi Driver
Completely observable	Yes	Yes	No
Deterministic	Yes	Yes	No
Episodic	No	No	No
Static	Semi	Yes	No
Discrete	Yes	Yes	No
SingleAgent	No	No	No

- The task environment type largely determines the agent project
- The real world is partially observable, stochastic, sequential, dynamic, continuous, multi-agent

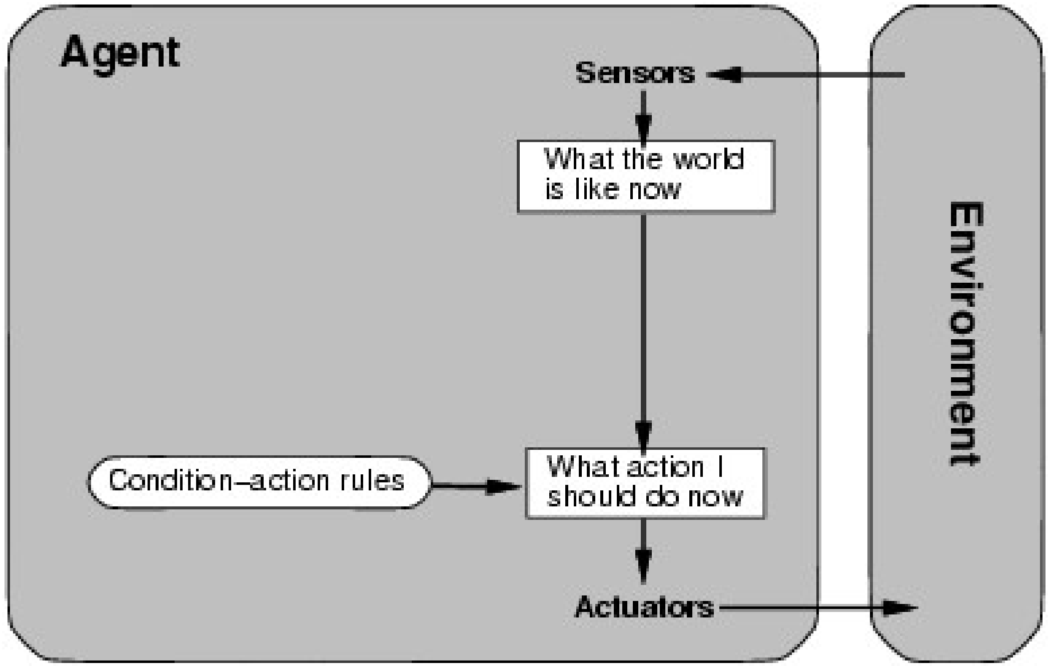
Basic Types of Agents

- Some of the basic types, from the simplest to the most complex
  - Simple reactive/reflex agents
  - Agents with world representation
  - Objective-based agents
  - Utility-based agents
  - Learning Agents
  - BDI Agents



Simple Reflex Agents

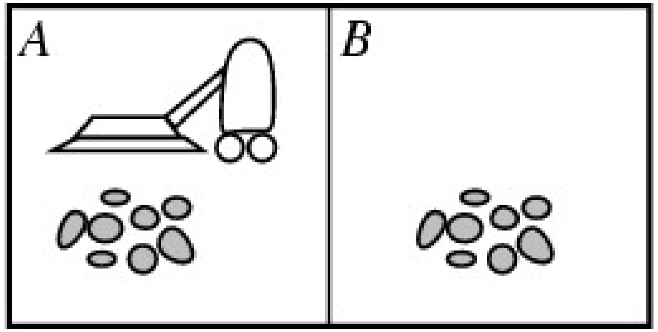
- Based on condition-action rules table (if-then rules)



Example: Simple Reactive Agent

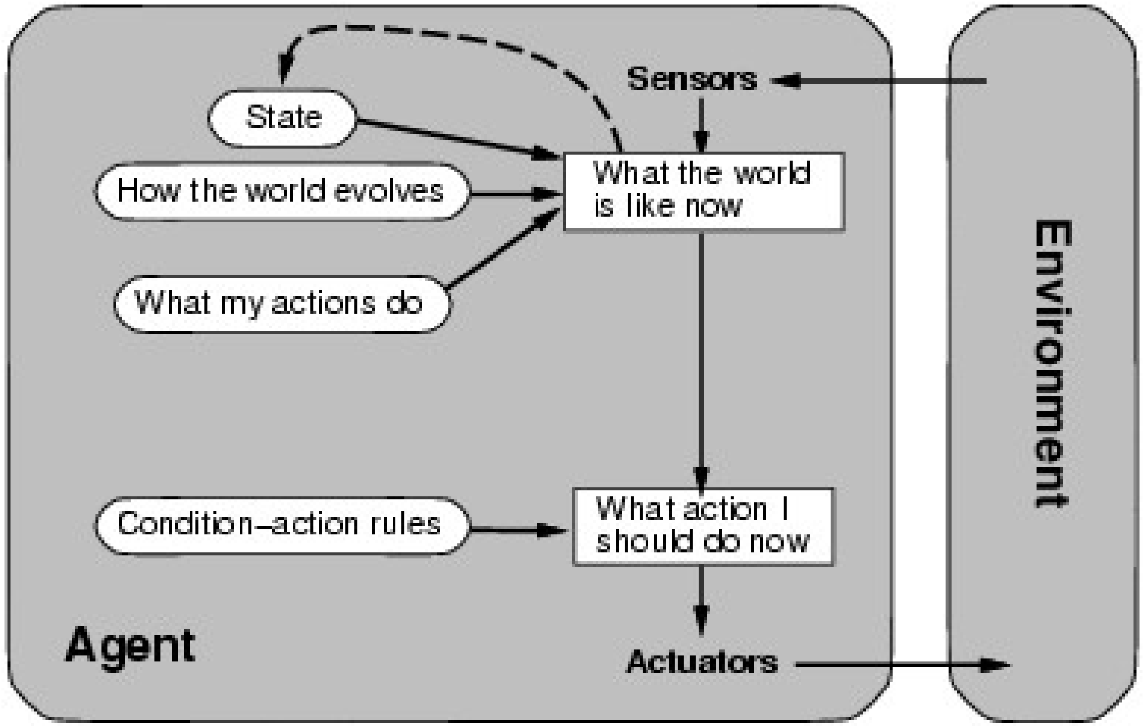
```
AGENT-VACUUM-REACTIVE function (position, state)
  returns an action
      if state = Dirty then returns Suck
      if not if position = A then returns Right
      if not if position = B then returns Left
```

- Condition-Action rules (if-then rules) make a direct link between current perception and action
- The agent works only if the environment is fully observable and the correct decision can be made based only on the current perception



Agents with World Representation

- Model-based reflex agents
- Maintain an internal state (representation of the world)



Agents with World Representation

Function AGENT-REACTIVE-With_World_Rep (*Perception*) **returns** an *Action*

Static Variables:

State, A description of the current world state

Rules, a set of condition-action rules

Action, the most recent action, inicially empty

State ← UPDATES_STATE(*State*, *Action*, *Perception*)

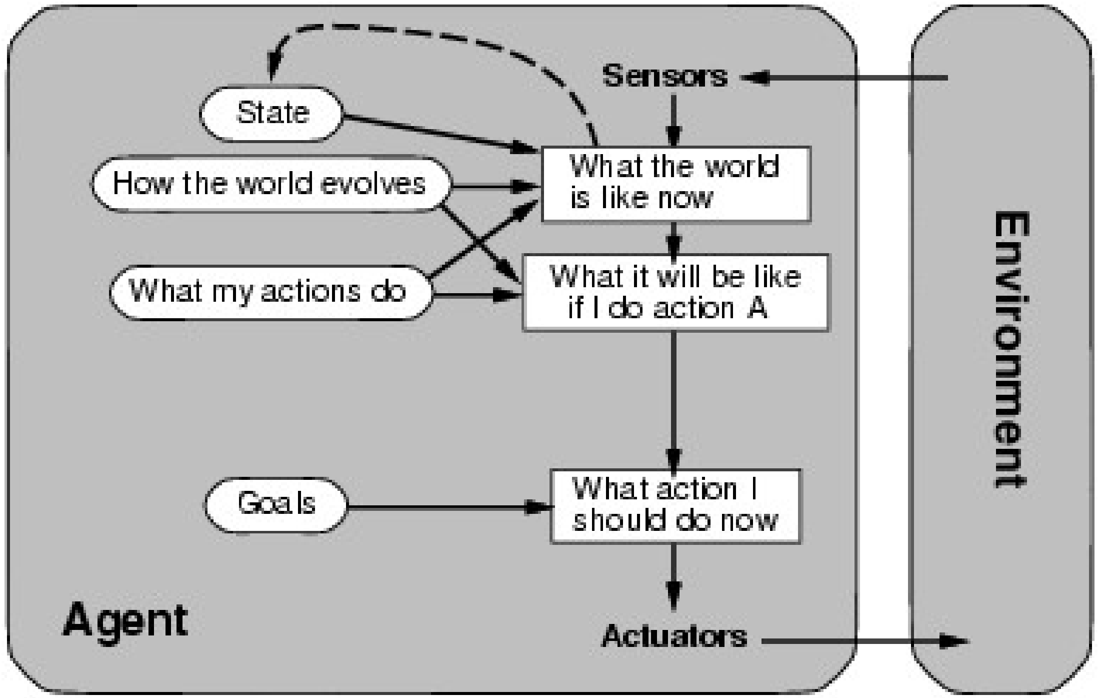
Rule ← DETERMINES_RULES(*State*, *Rules*)

Action ← RULE_ACTION(Rule)

returns *Action*

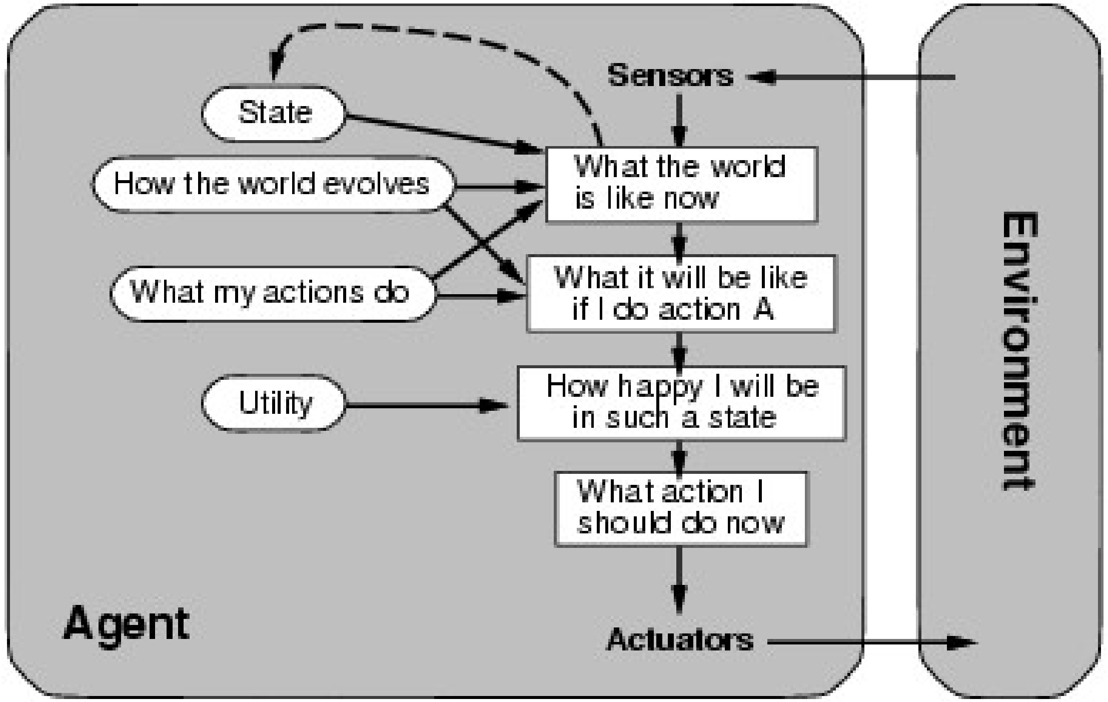
Objective Based Agents

- Description of the state of the world and the objective to be achieved
- Example of an Objective: Getting from Porto to Lisbon
- Problem Solving by Search, Planning



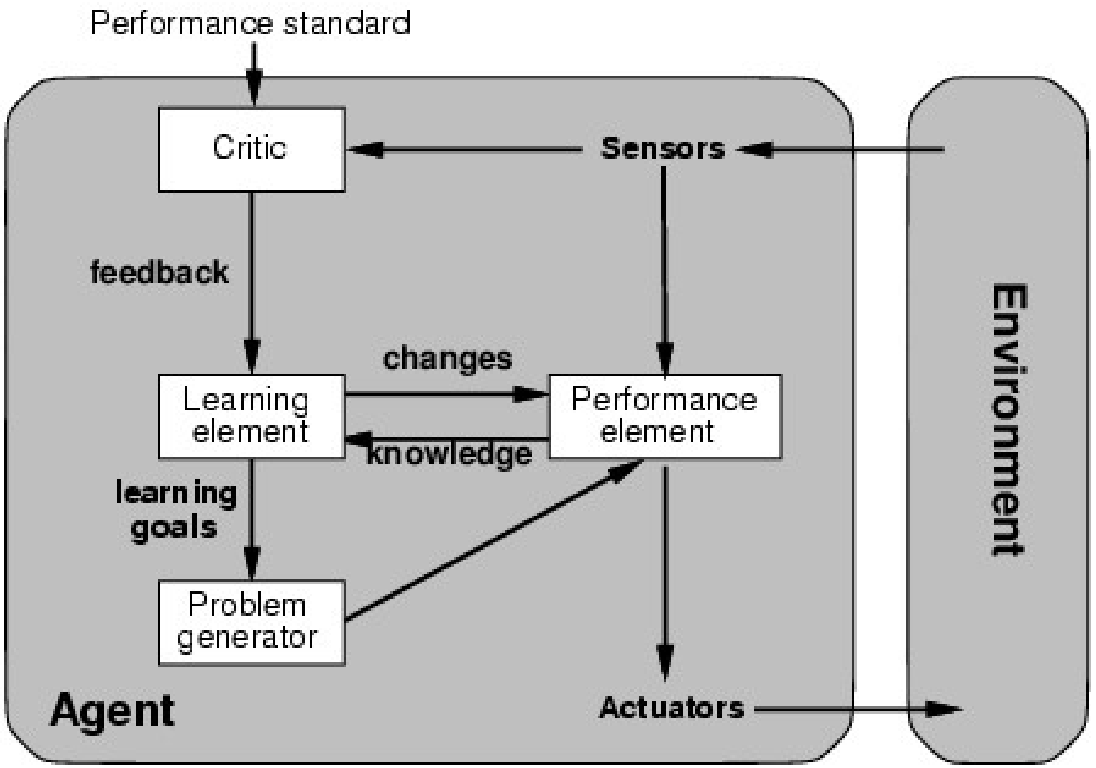
Utility Based Agents

- Utility: Performance measure
- Kind of agent's degree of happiness!
- Maps the current state to a value!



Learning Agents

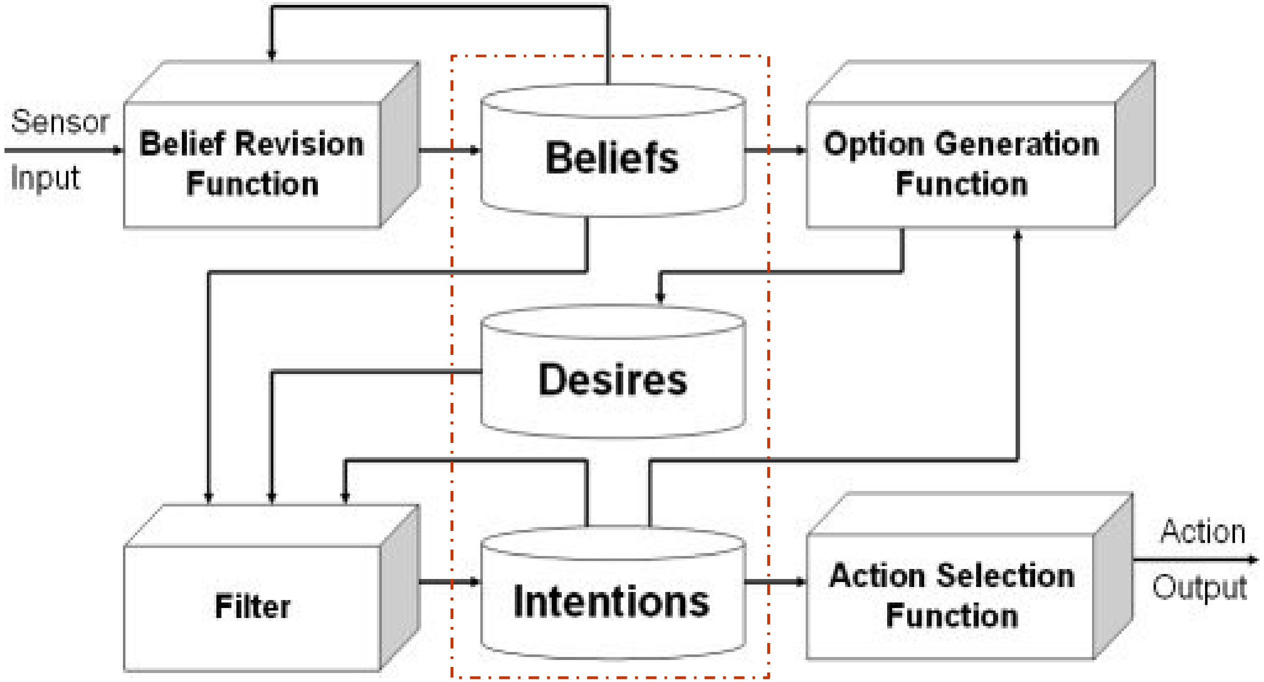
- Learning element and performance evaluation element
- Performance increases over time by using machine learning



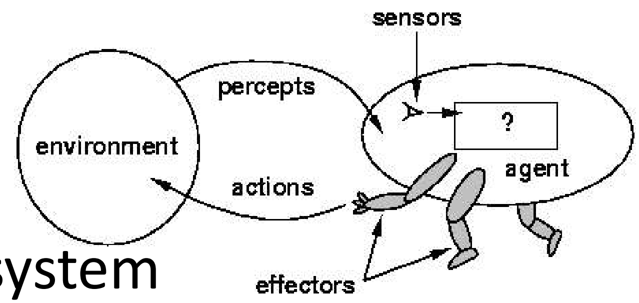
BDI Agents

Three "mental attitudes"

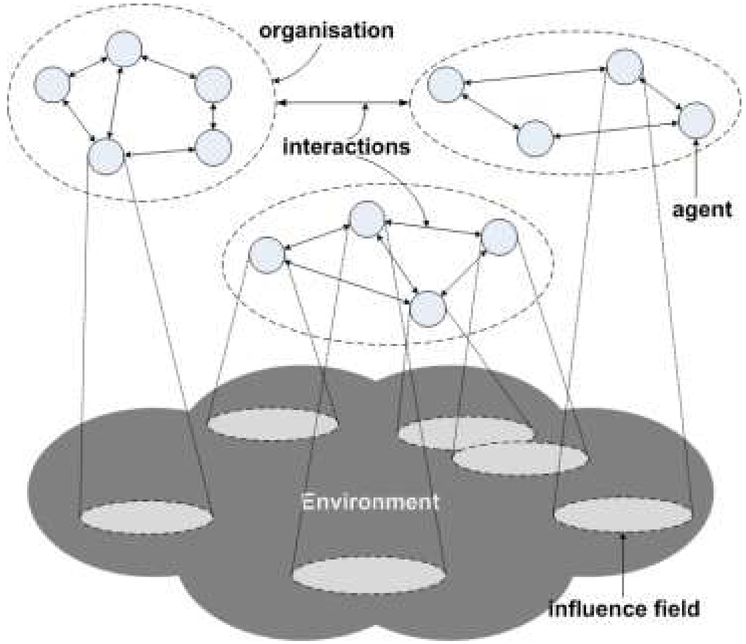
- **(B)eliefs** are information the agent has about the world – *information*
- **(D)esires** are all the possible states of affairs that the agent might like to accomplish – *motivation*
- **(I)ntentions** are the states of affairs that the agent has decided to work towards – *deliberation*



Multi-Agent System (MAS)



- Composed by **multiple agents** that:
  - Exhibit **autonomous behavior**
  - **Interact** with the other agents in the system
- **MAS Motivation:**
  - Natural Solution (distributed problems)
  - Distributed knowledge or information
  - Problem Dimensions
  - Human-machine interface
  - Project clarity and simplicity
  - Legacy Systems
  - Efficiency
  - Robustness and Scalability
  - Problem division
  - Information privacy



Exercício 1 – CONTROLEX Agent

CONTROLEX: Agente para Controlar Temperatura de uma Sala

Apresente um diagrama e o pseudo-código para um agente simples reflexo – CONTROLEX - para controlar a temperatura de uma sala. Suponha que dispõe das perceções T1 e T2 correspondentes à temperatura da sala e à temperatura exterior e as ações:

- AQ ligar o aquecedor, NAQ – Desligar o aquecedor, AC – ligar o ar frio, NAC – Desligar o ar frio, AJ – abrir as janelas, NAJ – fechar as janelas.

Pretende-se que a temperatura da sala esteja entre os 22 e os 24 graus. Sempre que seja possível usar as janelas para controlar a temperatura (não desperdiçando energia), tal deve ser efetuado. Sempre que a temperatura esteja mais de 2 graus afastada da banda desejada (ou seja se a temperatura for inferior a 20 ou superior a 26 graus), deve-se fechar as janelas e em vez disso, usar o aquecedor ou ar frio para repor a temperatura dentro da banda desejada.

Como poderia construir um agente um pouco mais inteligente para este problema (que tipo de agente, perceções, estado do mundo, etc., usar)?

Exercise 1 – CONTROLEX Agent

CONTROLEX: Agent to Control the Temperature of a Room

Present a diagram and the pseudo-code for a simple reflex agent - CONTROLEX - to control the temperature in a room. Suppose you have the T1 and T2 perceptions corresponding to the room temperature and the outside temperature and the actions:

- AQ turn on the heater, NAQ - Turn off the heater, AC - turn on the cold air, NAC - Turn off the cold air, AJ - open the windows , NAJ - close the windows.

It is intended that the temperature of the room is between 22 and 24 degrees. Whenever it is possible the agent should use the windows to control the temperature (not wasting energy). Whenever the temperature is more than 2 degrees away from the desired band (that is, if the temperature is below 20 or above 26 degrees), the windows must be closed and instead use the heater or cold air to reset the temperature. temperature within the desired band.

How could you build a slightly more intelligent agent for this problem (what kind of agent, perceptions, state of the world, etc., to use)?

Exercício 1: Tópicos de Resolução

- **Perceções:**
  - T1 – Temperatura Interior
  - T2 – Temperatura Exterior
- **Ações:**
  - AQ ligar o aquecedor
  - NAQ – Desligar o aquecedor
  - AC – ligar o ar frio
  - NAC – Desligar o ar frio
  - AJ – abrir as janelas
  - NAJ – fechar as janelas
- **Objetivo:**
  - Manter a temperatura da sala entre os 22 e os 24 graus

- **Agente mais inteligente:**
  - Câmaras para analisar quantas e quais as pessoas no interior da sala
  - Ajuste da temperatura em função dos gostos das pessoas
  - Utilização de previsões meteorológicas da Internet

Interpretação da Perceção:

M_QUENTE = T1 > 26

QUENTE = T1 > 24 e T1<= 26

NORMAL = T1 >= 22 e T1<= 24

FRIO = T1 >= 20 e T1< 22

M_FRIO = T1< 20

FORA_UTIL = T2 < 24 e QUENTE ou
T2 > 22 e FRIO

Regras Condição-Ação:

SE NORMAL
Então NAQ; NAC; NAJ

Se (QUENTE ou FRIO) e FORA_UTIL
Então NAQ; AJ; NAC

Se QUENTE e não(FORA_UTIL) ou M_QUENTE
Então NAQ; NAJ; AC

Se FRIO e não(FORA_UTIL) ou M_FRIO
Então AQ; NAJ; NAC

Exercise 1: Solution Topics

Perceptions:

- T1 - Indoor Temperature
- T2 - Outside Temperature

Actions:

- AQ turn on the heater
- NAQ - Turn off the heater
- AC - turn on cold air
- NAC - Turn off cold air
- AJ - open the windows
- NAJ - close the windows

Goal:

- Keep the room temperature between 22 and 24 degrees

More Intelligent agent:

- Cameras to analyse how many and which people are inside the room
- Temperature adjustment according to people's tastes
- Use of Internet weather forecasts

Perception Interpretation:

V_HOT = T1 > 26

HOT = T1 > 24 AND T1<= 26

NORMAL = T1 >= 22 AND T1<= 24

COLD = T1 >= 20 AND T1< 22

V_COLD = T1< 20

OUTSIDE_USEFUL =
  T2 < 24 AND HOT OR T2 > 22 AND COLD

Condition-Action Rules:

IF NORMAL
  THEN NAQ; NAC; NAJ

IF (HOT OR COLD) AND OUTSIDE_USEFUL
  THEN NAQ; AJ; NAC

IF HOT AND NOT (OUTSIDE_USEFUL) OR V_HOT
  THEN NAQ; NAJ; AC

IF COLD AND NOT(OUTSIDE_USEFUL) OR V_COLD
  THEN AQ; NAJ; NAC

Exercício 2 – Agente PISCINEX

PISCINEX: Agente para Controlar uma Piscina

Apresente um diagrama e o pseudo-código para um agente simples reflexo – PISCINEX - para controlar o nível de água e a temperatura de uma piscina. Suponha que dispõe das perceções TEMP correspondentes à temperatura da piscina e ALT correspondente à altura de água da piscina. Dispõe das ações: AS – abrir saída de água, FS – fechar saída da água, AEQ – Abrir entrada de água quente, FEQ – fechar entrada de água quente, AEF – Abrir entrada de água fria e FEF – fechar entrada de água fria.

Pretende-se que a temperatura da piscina esteja entre os 25 e os 27 graus e que o nível de água esteja entre 1.3 e 1.5 metros. É possível ligar simultaneamente uma das entradas e a saída da água mas não existe nenhuma garantia que o nível permaneça o mesmo. Não se pode manter ligada a entrada de água quando o nível da água esteja superior a 1.45 metros.

Exercise 2 – POOLEX Agent

POOLEX: Agent to Control a Pool

Present a diagram and the pseudo-code for a simple reflex agent - POOLEX - to control the water level and temperature of a swimming pool. Suppose you have the TEMP perceptions corresponding to the pool temperature and ALT corresponding to the pool's water height. It has the following actions: AS - open water outlet, FS - close water outlet, AEQ - Open hot water inlet, FEQ - close hot water inlet, AEF - Open cold-water inlet and FEF - close cold-water inlet.

It is intended that the temperature of the pool is between 25 and 27 degrees and that the water level is between 1.3 and 1.5 meters. It is possible to connect one of the inlets and the water outlet simultaneously, but there is no guarantee that the level will remain the same. The water inlet cannot be connected when the water level is above 1.45 meters

Exercício 3 – Agente RATEX

RATEX: Agente para Resolver um Labirinto Simples

Suponha um Robot autónomo com 2 rodas motrizes (MEsq, MDir), 3 sensores de proximidade (SEsq, SCentro e SDir), 1 sensor de chão (SChao) e 1 sensor de farol (SFarol) que se move num labirinto povoado por outros robots, tentando atingir a zona do mesmo onde se encontra o farol! Responda às seguintes questões:

a) Apresente uma descrição PEAS do Agente e classifique o seu ambiente

b) Será possível resolver todos os tipos de labirintos com um Agente puramente reativo? Justifique, apresentando exemplos.

c) Apresente um algoritmo simples que permita resolver labirintos para este agente.

d) Supondo que o agente após chegar à área de chegada deveria regressar ao ponto de partida. Explique como adaptaria o algoritmo desenvolvido para lidar com esta situação

e) Suponha que o agente quer unicamente mover-se no labirinto sem bater em nenhum outro robot! Indique uma arquitetura e implemente um agente (o mais simples possível) capaz de o fazer!

Exercise 3 – RATEX Agent

RATEX: Agent to Solve a Simple Maze

Suppose an autonomous Robot with 2 drive wheels (MLef, MRig), 3 proximity sensors (SLef, Scent and SRig), 1 floor sensor (SGround) and 1 headlight sensor (SLight) that moves in a maze populated by other robots , trying to reach the area where the lighthouse is! Answer the following questions:

a) Submit a PEAS description of the Agent and a complete classification of its environment

b) It will be possible to solve all types of mazes with a purely reactive Agent? Justify, giving examples.

c) Present a simple algorithm that allows solving mazes with this agent.

d) Assuming that the agent after arriving at the arrival area should return to the starting point. Explain how you would adapt the algorithm developed to deal with this situation

e) Suppose the agent only wants to move in the maze without hitting any other robot! Indicate an architecture and implement an agent (as simple as possible) capable of doing it!

Summary: Intelligent Agents

- An agent perceives and acts in an environment, has an architecture, and is implemented by an agent program
- PEAS (Performance, Environment, Actuators, Sensors)
- The most challenging environments are inaccessible, nondeterministic, dynamic, continuous and multi-agent
- Rational agent chooses the action which maximizes its expected performance, given its percept sequence so far
- An agent program maps from percept history to action and updates internal state.
  - **Reflex agents** respond immediately to percepts (simple reflex agents, model-based reflex agents)
  - **Goal-based agents** act in order to achieve their goal(s)
  - **Utility-based agents** maximize their own utility function
- Agents can improve their performance through learning

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

IART - Artificial Intelligence
Lecture 1: Intelligent Agents

Luís Paulo Reis

lpreis@fe.up.pt

Director of LIACC – Artificial Intelligence and Computer Science Lab.
Associate Professor at DEI/FEUP – Informatics Engineering Department,
Faculty of Engineering of the University of Porto, Portugal
President of APPIA – Portuguese Association for Artificial Intelligence