Artificial Intelligence 1

Roman Barták
Department of Theoretical Computer Science and Mathematical Logic

Artificial intelligence is the science of making machines do things that would require intelligence if done by men.

Marvin Minsky, 1967

Four Views to Artificial Intelligence

	human perfomance	*ideal performance (rationality)*
though processes	**Thinking humanly** cognitive science	**Thinking rationally** logic
behaviour	**Acting humanly** behaviorism	**Acting rationally** doing the right thing

Acting Humanly

Alan Turing (1950) provided an operational definition of intelligence.

– „Can machines think?“ *like a man*

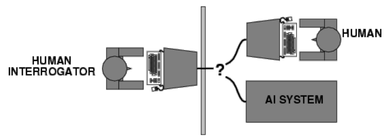
↳ „Can machines act ~~intelligently~~?“

– **Turing test**

A computer passes the test if a human interrogator, after posing some written questions, cannot tell whether the written responses come from a person or from a computer.

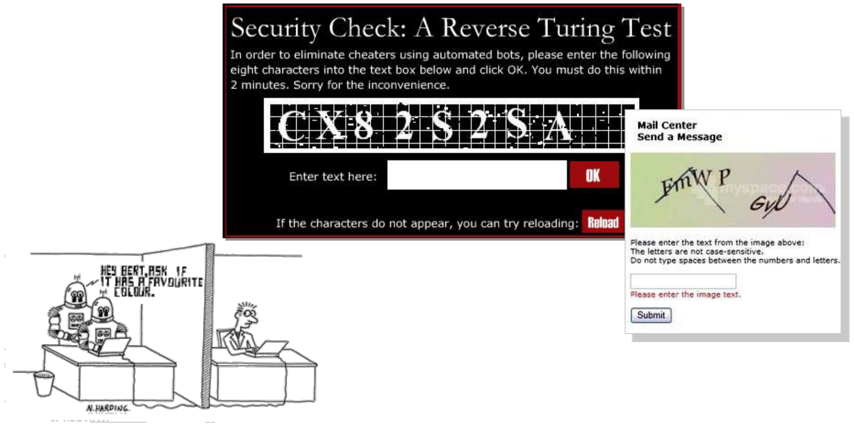
– **Required capabilities:**

- **natural language processing**
- **knowledge representation**
- **automated reasoning**
- **machine learning**
- computer vision
- robotics



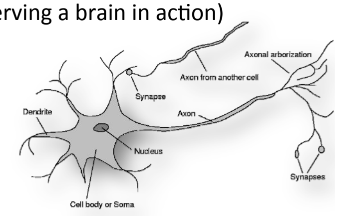
Reverse Turing test

computer attempts to recognize whether it communicates with a computer or a person



Cognitive Modelling

– modelling human mind
– we must have some way of determining how humans think
  - Top-down approach (**psychology**)
    – following human reasoning steps (found through introspection or through observing a person in action
    – GPS: General Problem Solver (Newell & Simon, 1957)
  - Bottom-up (**neuroscience**)
    – modelling the brain (through observing a brain in action)
    – connectionist models
    – „intelligent behaviour emerges by connecting a large number of simple units“



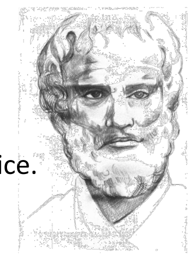
Since the time of **Aristotle** (384 – 322 BC) people attempted to codify „right thinking“

– *Syllogisms*
  - Patterns for argument structures that always yield correct conclusions when given correct premises
  - Socrates is a man, all men are mortal
    ⇒ Socrates is mortal

– This study initiated the field of **logic** (and mathematics)

Major obstacles:

– It is not easy to take informal knowledge and state it in the formal terms required by logical notation, particularly when the knowledge is less than 100% certain.
– There is a big difference between solving a problem „in principle“ and solving it in practice.



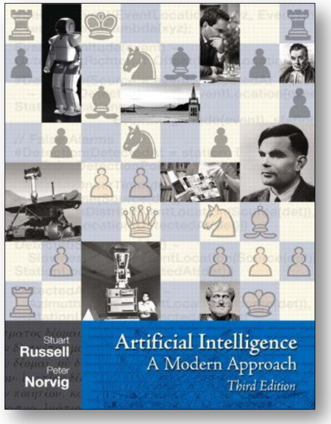
- **Rational behaviour** = doing „right things"
- **„right thing"** = achieving the best (expected) outcome even when there is uncertainty
- Making correct inferences (thinking rationally) is part of being a **rational agent**, but not exclusive.
  – In some situations, there is no provable correct thing to do, but something must still be done.
  – There are also ways of acting rationally that cannot be said to involve inference (for example, reflex actions).
  – **This course concentrates on general principles of rational agents and on components for constructing them.**

Course Structure

- **Introduction**
  - a bit of history, context, intelligent agents
- **Problem Solving**
  - search algorithms, constraint satisfaction
- **Knowledge and Reasoning**
  - logic and logical inference, knowledge representation
- **Planning**
  - composing actions to achieve goals

Resources

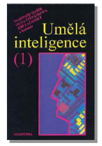
Artificial Intelligence: A Modern Approach

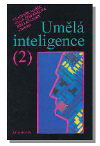


- S. Russell and P. Norvig
- Prentice Hall, 2010 (third ed.)
- http://aima.cs.berkeley.edu/

Umělá inteligence 1-6

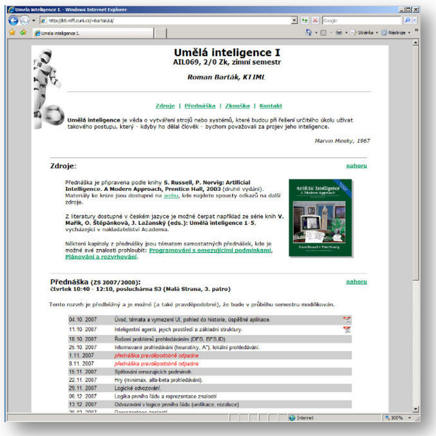
- Vladimír Mařík, Olga Štěpánková, Jiří Lažanský a kol.
- Academia





Course Web Page

`http://ktiml.mff.cuni.cz/~bartak/ui/`



You can find there:

- **slides**
- **links and resources**
- **contacts**
- **…**

Links to Other Courses

- **Seminar on Artificial Intelligence**
  - about theoretical and practical questions in a field of Artificial Intelligence
- **Constraint Programming**
  - about techniques of constraint satisfaction
- **Decision Procedures and Verification**
  - about logical inferences
- **Planning and Scheduling**
  - about automated construction of plans and schedules
- **Machine Learning**
  - about teaching computers to learn new things
- …

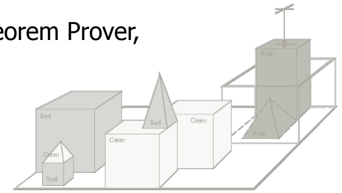
The Foundations of Artificial Intelligence

Artificial Intelligence draw ideas and techniques from many disciplines.

• **Philosophy** (428 BC -)	how does the mind arise from a brain? **logic**, reasoning techniques
• **Mathematics** (800 -)	what are the **formal** rules to draw valid conclusions? what can be computed?
• **Economics** (1776 -)	how to maximize payoff? **utility** theory, decision processes
• **Neuroscience** (1861 -)	how do **brains** process information? the physical seat of consciousness
• **Psychology** (1879 -)	how do humans think and act? **behaviourism**
• **Computer engineering** (1940 -)	how to build an efficient **computer**? machines for information processing
• **Control theory** (1948 -)	how can **artefacts** operate under their own control? systems maximizing an objective function over time
• **Linguistics** (1957 -)	how does **language** relate to thought? knowledge representation

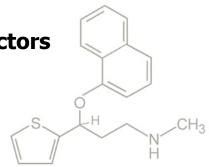
The History of Artificial Intelligence

- **The gestation of AI (1943-1955)**
  - W. McCulloch & W. Pitts: Boolean model of **neurons**
  - A. Turing: **„Computing Machinery and Intelligence"** the first complete vision of artificial intelligence
- **The birth of AI (1956)**
  - two-months workshop at **Dartmouth** College, NH
  - J. McCarthy gave the name **Artificial Intelligence**
  - A. Newell & H. Simon: software **Logic Theorist**
- **Great expectations (1952-1969)**
  - demonstrating one X after another from the list "a machine can never do X"
  - General Problem Solver, Geometry Theorem Prover, **Lisp (1958)**, Analogy, blockworld
  - J. McCarthy referred to this period as the **„Look, Ma, no hands!"** era.



The History of Artificial Intelligence

- **A dose of reality (1966-1973)**
  - „There are now machines that think, that learn and that create", but only on simple problems
  - Why?
    - the early programs **knew nothing of their subject matter**; they succeeded by means of simple syntactic manipulations
    - **intractability** of many problems that AI was attempting to solve (trying out different combinations of steps until the solution was found)
    - **fundament limitations** on the basic structures used (perceptron learns anything it can represent, but it could represent very little)
- **Knowledge-based systems (1969-1979)**
  - The alternative to „weak" general methods is to use more powerful, domain-specific knowledge.
  - expert (knowledge) systems:
    - **DENDRAL** (Buchanan) inferring molecular structure from the information provided by a mass spectrometer, introducing **rules** based on well-known patterns to reduce possible structures
    - **MYCIN** (Feigenbaum) diagnosing blood infections, introducing **certainty factors**
    - **PROLOG** (Colmerauer, 1972)
    - **frames** (Minsky, 1975) – motivations for current OOP



The History of Artificial Intelligence

- **AI becomes an industry (1980)**
  - commercial expert system **R1** for configuring computers DEC ($40 mil./year)
  - **Fifth Generation** of computers (Japan, 1981)
    - a 10-year plan to build intelligent computers running Prolog
  - **boom of AI industry** (billions of dollars in 1988)
  - and then the **„AI Winter"**
    - companies failed to deliver on extravagant promises (like the dot.com bubble)
- **The return of neural networks (1986)**
  - reinventing back-propagation learning algorithm
- **AI adopts the scientific method (1987)**
  - AI has come firmly under the **scientific method**, hypothesis must be subjected to rigorous empirical experiments, and the results must be analysed statistically for their importance; experiments can be replicated
  - novel approaches: hidden Markov models, Bayesian networks, data mining
  - formalisation and specialisation led to **fragmentation**
- **The emergence of intelligent agents (1995)**
  - encouraged by progress in solving the subproblems of AI researchers started to look at the "whole agent" problem again
  - SOAR (State, Operator and Result) – a complete agent architecture

Logistics


Gulf War 1991:

- Traditional approach:
  - hundreds of human planners
  - months to generate plans
- IP&S approach:
  - O-PLAN2 helps human planners
- **Savings:**
  - faster development of background
  - less cargo flights
  - return of investment >> **all AI research supported by US government**:
    - Since 1956
    - not only IP&S, but **all AI research!**

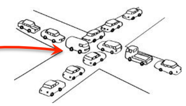
Deep Space 1

Launch: October 24, 1998

Target: Comet Borrelly

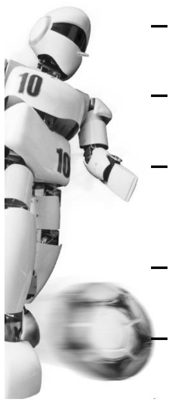
testing a payload of 12 advanced, high risk technologies

- **autonomous remote agent**
  - planning, execution, and monitoring spacecraft activities based on general commands from operators
  - three testing scenarios
    - 12 hours of low autonomy (execution and monitoring)
    - 6 days of high autonomy (operating camera, simulation of faults)
    - 2 days of high autonomy (keep direction)
      - » **beware of backtracking!**
      - » **beware of deadlock in plans!**

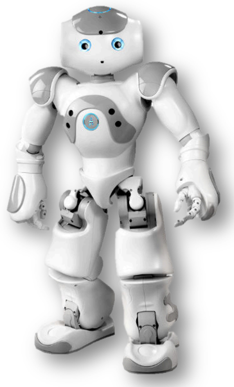


RoboCup

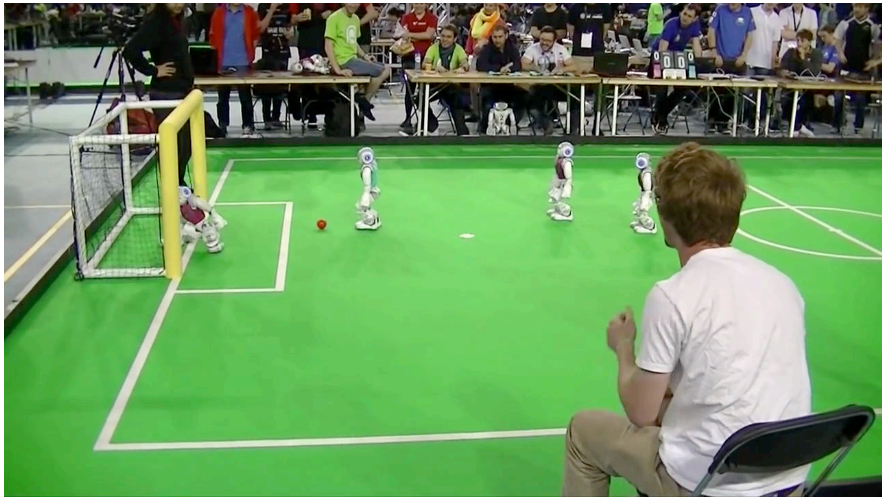
„By mid-21st century, a team of fully autonomous humanoid robot soccer players shall win the soccer game, complying with the official rule of the FIFA, against the winner of the most recent World Cup.“

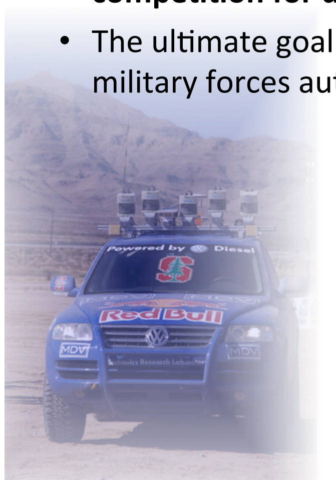


- **Simulation league**
  simulated games in computers
- **Small size league**
  robots limited to a 18 cm diameter
- **Middle size league**
  robots limited to a 50 cm diameter
  all sensors
- **Standard platform league**
  Sony Aibo, Nao
- **Humanoid league**
  penalty kicks and two-to-two game



RoboCup Emotions





- The Grand Challenge was the first **long distance competition for driverless cars** in the world.
- The ultimate goal was making one-third of ground military forces autonomous by 2015.
  - **2004 Grand Challenge**
    - Failure - None of the robot vehicles finished the route (max. 11,78 km, CMU)
  - **2005 Grand Challenge**
    - Done! Winner Stanley (212.4 km in about 7 hours, Stanford)
  - **2007 Urban Challenge**
    - Winner BOSS (CMU) driving in urban areas

Department of Theoretical Computer Science and Mathematical Logic
bartak@ktiml.mff.cuni.cz

Artificial Intelligence 1
Quiz #1 (history and terminology)

What is AI winter?
a) the time of year when AI researchers meet at a major conference (AAAI)
b) the period in last century when funding to AI in companies went down
c) a winter competition of AI systems
d) All Inclusive offer at hotels during winter time

Who is the author of term Artificial Intelligence?
a) Marvin Minski
b) Alan Turing
c) John McCarthy
d) Elon Musk

What is an imitation game?
a) an original name of Turing test
b) a special presentation when one researcher imitates another researcher
c) one of favorite games that AI scientists play
d) a computer program imitating humans

What is a syllogism?
a) an old form of philosophy
b) a specific religion of AI scientists
c) logical consequence
d) a pattern for correct deduction

Thinking Humanly is a view of Artificial Intelligence, where the system
a) behaves like a human
b) thinks in the best possible way
c) uses same cognitive processes as a human
d) must protect humans from any danger

Logic Theorist is
a) a system that can do automated theorem proving
b) a logical computer game
c) a pejorative denotation for people doing logic
d) a nickname of George Boole

Control theory is
a) a theory how to control humans
b) a theory how to control companies
c) a research area about automatic control of artifacts
d) a part of economy about running (controlling) companies effectively

Does rational behavior mean that an agent always achieves the best output?
a) yes
b) no

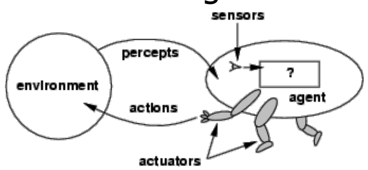
Artificial Intelligence 1

Roman Barták
Department of Theoretical Computer Science and Mathematical Logic

Intelligent agents, environment, structure of agents.

- Recall, that this course will focus on **rational behaviour**:
  – behaviour (not necessarily though processes),
  – ideal performance (not necessarily equal to human)
- We attempt to build **rational agents**
  – What is an **agent**?
  – What is a **rational agent**?
  – How does an **agent environment** look?
  – What **properties of environments** are important?
  – What are skeleton **agent designs**?

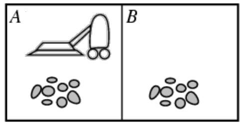
- An **agent** is anything that can be viewed as perceiving its **environment** through **sensors** and acting upon that environment through **actuators**.



- **Some examples:**
  – a human agent
    • eyes, ears, nose, … → hands, legs, mouth,…
  – a robotic agent
    • camera, infrared finder, … → arms, wheels, …
  – a software agent
    • keyboard, network packets… → screen, sending packets,…

- Agent perceives **percepts** and agent's behaviour is fully determined by the complete history of everything the agent has ever perceived.
- Formally, agent's choice of action can be described by an **agent function** (table):
  – $V^* \rightarrow A$, where V is a set of percepts, A is a set of actions
  – The agent function can be built by observing agent's behaviour for all possible percept sequences.
    • we need "restart" capability for the agent and "enough" time and space
  – The agent function is an **abstract mathematical description**.
- Internally, the agent function will be implemented by an **agent** program.

Example of an agent



Vacuum-cleaner
- percepts:
  - location (A,B)
  - property (clean, dirty)
- actions: suck up, move left, move right, do nothing

Agent function:

A sequence of percepts	**action**
(A,clean)	move right
(A,dirty)	suck up
(B,clean)	move left
(B,dirty)	suck up
(A,clean), (A,clean)	move right
…	

Agent program:

if property=dirty **then**
  suck up
else if location=A **then**
  move right
else if location=B **then**
  move left

Performance measure

- **What is the right way to fill out the table?** or **What makes an agent good or bad?**
- We consider consequences of agent's behaviour. The notion o desirability is captured by a **performance measure** that evaluates any given sequence of environment states.
- **Who do usually defines the performance measure?**
  - an agent designer
- **How to set the performance measure?**
  - It is better to design to design performance measures according to **what one actually wants in the environment**, rather than according to how one thinks the agent should behave.
- **Example** (vacuum cleaner)
  - performance measure: suck up as much dirt as possible
    - possible behaviour: suck up, dump, suck up, dump, …
  - better performance measure: have a clean floor

Rational agent

- **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.
- Beware, this is different from omniscience!
  - rational agents maximize **expected** performance measure
  - omniscience agents maximize the **actual** performance measure
- A rational agent should be **autonomous** – it should learn what it can to compensate for partial or incorrect prior knowledge.

Task environment

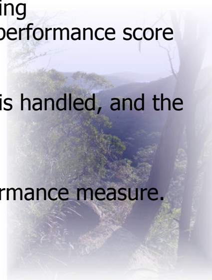
- In addition to **sensors**, **actuators**, and **performance measure** agents also need **environment** to affect (together this is called a **task environment**).

Example: automated taxi driver

Agent	**Perfomance measure**	**Environment**	**Actuators**	**Sensors**
Taxi drive	safe, fast, comfortable, profit	roads, other traffic, pedestrians, customer	steering accelerator, break signal, horn	camera sonar odometer GPS

Properties of task environments

- **Fully observable / partially observable**
  - agent‘s sensors give a access to the complete state of evironment
- **Deterministic / stochastic**
  - the next state of environment is fully determined by the current state and the action executed
  - strategic = only (other) agents can modify the environment
- **Episodic / sequential**
  - the agent’s experience is divided into atomic episodes (the next episode does not depend on actions taken in previous episodes)
- **Static / dynamic**
  - environment is not changing while an agent is deliberating
  - semidynamic = environment does not change, but the performance score does
- **Discrete / continuous**
  - depends of the state of the environment, the way time is handled, and the percepts and actions o the agent
- **Single agent / multi-agent**
  - Which entities must be viewed as agents?
    If their behaviour is best described as maximizing performance measure.
  - competitive vs. co-operative multi-agent environments



Examples of task environments

Environment	Observable	Determististic	Episodic	Static	Discrete	Agents
Crossword puzzle	fully	deterministic	sequential	static	discrete	single
Chess with clock	fully	strategic	sequential	semi	discrete	multi
Taxi	partialy	stochastic	sequential	dynamic	continuos	multi
Image analysis	fully	deterministic	episodic	semi	continuos	single

- **The simplest environment**
  - fully observable, deterministic, episodic, static, discrete with a single agent
- **The most challenging environment (real-life)**
  - partially observable, stochastic, sequential, dynamic, continuous, multi-agent

The structure of agents

agent = architecture + program

- **architecture** = a computing device with physical sensors and actuators
- **program** = implementation of the agent function
  - the **mapping from percepts to actions**
  - more precisely, the agent program takes the current percept as its input (because nothing more is available from the environment) and returns an action to actuators
  - if the agent’s actions depend on the entire percept sequence then the agent will have to remember the percepts
  - Obviously, the **program must be appropriate for the architecture**!

Table-driven agent

A straightforward agent that retains the complete percept sequence in memory and uses it as an index to the table with actions.

function TABLE-DRIVEN_AGENT(*percept*) **returns** an action

static: *percepts*, a sequence initially empty
table, a table of actions, indexed by percept sequence

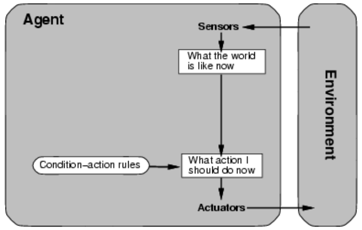
append *percept* to the end of *percepts*
action ← LOOKUP(*percepts*, *table*)
return *action*

Problems:

- the table is too large (even for agents working with the limited number of steps)
- the designer would not have enough time to create the table
- no agent could ever learn all the right table entries from experience
- the designer has no guidance about how to fill in the table entries

We need to find a different way!

Simple reflex agent



– the agent selects an action on the basis of the **current percept**
– implemented as condition-action rules (**if** property=dirty **then** suck up)
– significant reduction of the number of possibilities (to the number of percepts)

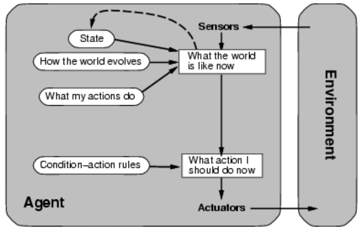
function SIMPLE-REFLEX-AGENT(*percept*) **returns** an action

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

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

- works for fully observable environments (otherwise may loop infinitely).
- randomisation of actions may help to escape from infinite loops.

Model-based reflex agent



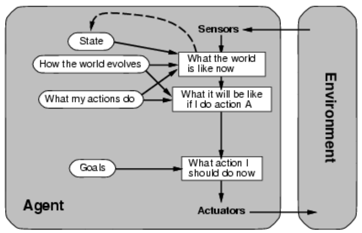
– Partial observability can be handled by keeping track of the part of world the agent cannot see now.
– Two kind of knowledge is necessary:
  - how the world evolves (independently of the agent)
  - how the agent's own actions affect the world
– **model of the world**

function MODEL-BASE-REFLEX-AGENT(*percept*) **returns** an action

static: *rules*, a set of condition-action rules
model, how the next state depends on the current state ad action
state, a description of the current world state
action, the most recent action.

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

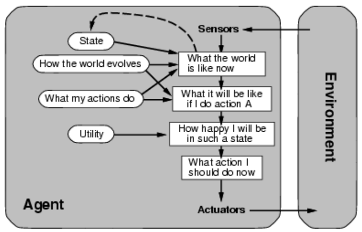
Goal-based agent



– Action selection is based not only on the state but on what the agent is trying to do.
– The agent needs some sort of **goal** information describing desirable situations.

- The major innovation is involving **consideration of the future**.
- **Search** and **planning** are devoted to finding action sequences that achieve the agent's goals.
- Goal-driven agent appears to be **less efficient** (than simple reflex agent), but it is **more flexible**.

Utility-based agent



– Goals alone are not enough to generate high-quality behaviour in most environments (goal reached / not reached).
– A more general performance measure allows comparison of different world states.

- It is possible to map states (or their sequences) to **utility** describing the performance measure.
  – The utility function is an **internalization of the performance measure** (the agent chooses actions to maximize its utility which will be rational if it corresponds to the external performance measure).
  – The agent can perform even if there are **conflicting goals** or **chances to achieve different goals are not equal**.

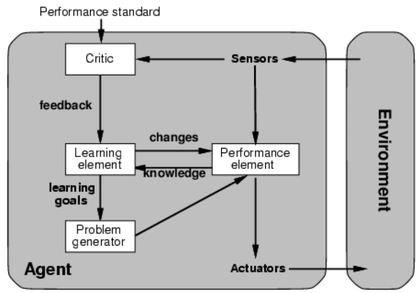
Detour - Representation of environment

How can we represent the environment that the agent inhibits?

- **atomic representation**
  - each state of the world is indivisible (blackbox)
  - used in search and game-playing algorithms, Markov Decision Processes
- **factored representation**
  - each state splits into a fixed set of variables (attributes), each of which can have a value
  - used in constraint satisfaction, propositional logic, and planning
- **structured representation**
  - each state consists of a set of objects (each may have attributes) with various and varying relationships
  - used in first-order logic

General learning agent

- So far we described how agents select actions, but **how do we obtain the programs for action selection**?
- One option is building **learning agents** and then to teach them instead of instructing them.
- The learning agent can operate in initially unknown environment and to become more competent that its initial knowledge alone might allow.
- We can extend any structure of agent to a learning agent by assuming:



- **Performance element**
  - the initial agent structure responsible for action selection
- **Learning element**
  - responsible for making improvements
- **Critic**
  - feedback on how the agent is doing (perceps themselves provide no indication of the agent's success)
- **Problem generator**
  - responsible for suggesting actions that will lead to new and informative experiences (exploratory actions)

A short summary

- An **agent** is something that perceives and acts in an environment.
- The **agent function** specifies the action taken by the agent in response to any percept sequence.
- The **performance measure** evaluates the behaviour of the agent in an environment.
- A **rational agent** acts so as to maximize the expected value of the performance measure.

A short summary

- The **agent program** implements the agent function.
  - *Simple reflex agents* respond directly to percepts
  - *Model-based reflex agents* maintain internal state to track aspects of the world that are not evident in the current percept.
  - *Goal-based agents* act to achieve their goals.
  - *Utility-based agents* try to maximize their own expected "happiness"
- All agents can improve their performance through **learning**.
- Different components of the agent structure answer questions such as:
  - *What is the world like now?*
  - *What action should I do now?*
  - *What do my actions do?*

Artificial Intelligence 1
Quiz #2 (agents)

What is an agent?

Can an agent exist without some environment?

What is the difference between sensors and actuators?

What is the relation between an agent function and an agent program?

What is an agent architecture and does it relate to an agent program?

What is the difference between rational and omniscient behavior?

How do we distinguish between good and bad behaviors of an agent?

Give an example of a game that is partially observable and explain why.

Do the Mars rovers Spirit and Opportunity operate in a multi-agent environment and why?

What is the difference between simple reflex agents and model-based reflex agents?

What is the major difference between reflex agents and goal-based agents?

What is critic for the learning agent?

Describe three possible representations of environment (world states) and give some examples.