Artificial Intelligence
(Introduction to)

2003-2004

Instructor

Dr Sergio Tessaris

- Researcher, faculty of Computer Science
- Contact
  - web page: tina.inf.unibz.it/~tessaris
  - email: tessaris@inf.unibz.it
  - phone: 0471 315 652
  - room 229 (2nd floor, left wing)
- Research interests
  - Knowledge Representation
  - Knowledge Representation and Databases
  - Semantic Web

Introduction

What is AI?

- **Turing, A.M.** (1950). *Computing machinery and intelligence. Mind, 59, 433-460*.
  - I propose to consider the question, “Can machines think?” This should begin with definitions of the meaning of the terms “machine” and “think”.
- “Can machines behave intelligently?”
  - *Turing Test* : an operational definition
- “AI is the science and engineering of making intelligent machines which can perform tasks that require intelligence when performed by humans”

Why study AI?

- scientific curiosity
  - try to understand entities that exhibit intelligence
- engineering challenges
  - building systems that exhibit intelligence
- some tasks that seem to require intelligence can be solved by computers
  - e.g. playing chess
- progress in computer performance and computational methods enables the solution of complex problems by computers
- humans may be relieved from tedious or dangerous tasks
  - e.g. demining or cleaning the swimming pool

What is AI?

Systems that think like humans	**Systems that think rationally**
Systems that act like humans	**Systems that act rationally**

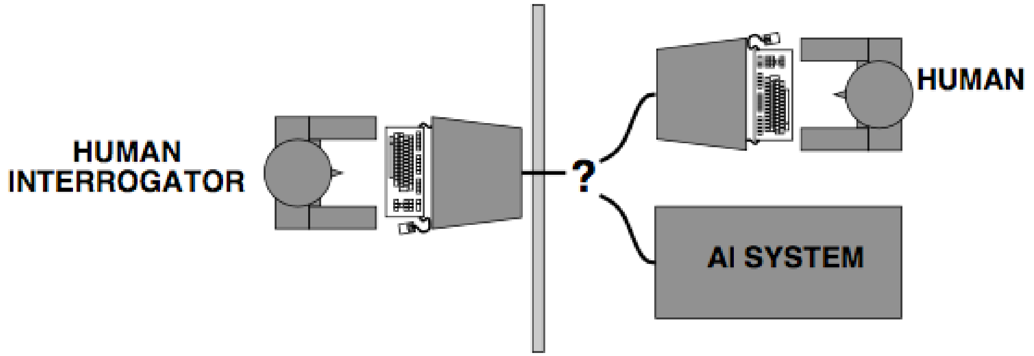
"The exciting new effort to make computers think… machines with minds, in the full and literal sense" [Haugeland, 1985] "[The automation of] activities that we associate with human thinking, activities such as decision-making, problem solving, learning …" [Bellman, 1978]	"The study of mental faculties through the use of computational models" [Charniak and McDermott, 1985] "The study of the computations that make it possible to perceive, reason, and act" [Winston, 1992]
"The art of creating machines that perform functions that require intelligence when performed by people" [Kurzweil, 1990] "The study of how to make computers do things at which, at the moment, people are better" [Rich and Knight, 1991]	"A field of study that seeks to explain and emulate intelligent behavior in terms of computational processes" [Schalkhoff, 1990] "The branch of computer science that is concerned with the automation of intelligent behavior" [Luger and Stubblefield, 1993]

Thinking humanly: Cognitive Science

- tries to construct theories of how the human mind works
- uses computer models from AI and experimental techniques from psychology
- most AI approaches are not directly based on cognitive models
  - often difficult to translate into computer programs
  - performance problems
- Cognitive Science is mainly distinct from AI

Acting humanly: The Turing test

- Operational test for intelligent behaviour: the Imitation Game



- Anticipated all major arguments against AI in following 50 years
- Suggested major components of AI: knowledge, reasoning, language understanding, learning

The Turing test

- not much work on systems that pass the test

  Problem: Turing test is not reproducible, constructive, or amenable to mathematical analysis

- Loebner Prize

  www.loebner.net/Prizef/loebner-prize.html

- Total Turing Test
  - includes video interface and a “hatch” for physical objects
  - requires computer vision and robotics as additional capabilities

Thinking Rationally: Laws of Thought

- mathematical logic as tool: notation plus derivation rules
- problems and knowledge must be translated into formal descriptions
- the system uses an abstract reasoning mechanism to derive a solution
- Problems:
  - Not all intelligent behaviour is mediated by logical deliberation
  - Resource limitations: There is a difference between solving a problem in principle and solving it in practice under various resource limitations such as time, computation, accuracy

Acting rationally

- rational behaviour: doing the right thing
- The right thing: that which is expected to maximize goal achievement, given the available information
- Doesn't necessarily involve thinking (e.g., blinking reflex) but thinking should be in the service of rational action
- Advantages:
  - More general
  - Its goal of rationality is well defined

Short history of AI (late 40s, 50s)

- artificial neurons (McCulloch and Pitts, 1943)
- learning in neurons (Hebb, 1949)
- chess programs (Shannon, 1950; Turing, 1953)
- neural computer (Minsky and Edmonds, 1951)
- official birth in summer 1956
  - gathering of a group of scientists with an interest in computers and intelligence during a two-month workshop in Dartmouth, NH
  - “naming” of the field by John McCarthy
  - many of the participants became influential people in the field of AI

Short history of AI (late 50s, 60s)

- Early successes
  - Logic Theorist (Newell and Simon, 1957)
    - able to proof most of the theorems in Ch2 of *Principia Mathematica*
  - General Problem Solver (Newell and Simon, 1961)
    - imitate human problem-solving methods (thinking humanly)
  - Shakey the robot (SRI)
    - logical reasoning and physical activity
  - Microworlds
    - ANALOGY: geometric analogies (Evans, 1968)
    - STUDENT: algebraic problems (Bobrow, 1967)
    - blocks world (Winston, 1970; Huffman, 1971; Fahlman, 1974; Waltz, 1975)
  - neural networks (Widrow and Hoff, 1960; Rosenblatt, 1962; Winograd and Cowan, 1963)
  - machine evolution/genetic algorithms (Friedberg, 1958)

Short history of AI (late 60s, 70s)

- AI and reality
  - lacks of "common sense" (e.g. ELIZA)
  - microworlds aren't the real thing: scalability and intractability problems (**NP-completeness**)
  - neural networks can learn, but not very much (Minsky and Papert, 1969)
- Knowledge-based systems: **knowledge is separate from reasoning**
  - expert systems
  - frames
  - logic based knowledge representation systems (80s-90s)
- knowledge representation schemes become useful

Short history of AI (80s)

- AI becomes an industry
  - Expert systems: Digital Equipment, Teknowledge, Intellicorp
  - Lisp machines: LMI, Symbolics
  - Constraint programming: ILOG
  - Robotics: Machine Intelligence Corporation, Adept, ABB
  - Speech understanding
- the return of neural networks
  - genetic algorithms and artificial life
- falling of Expert systems (late 80s)
  - feeding rules into a reasoning system is not enough
  - knowledge acquisition is a bottleneck

Short history of AI (last decade)

- AI becomes less philosophical, more technical and mathematically oriented
  - grounded on formal proofs or experimental evidence (vs intuition)
  - e.g. speech recognition, planning, Knowledge Representation
- **Agents** everywhere
  - agent architectures (e.g. SOAR)
  - agent perspective glues various AI fields
- Information management
  - to help humans in dealing with information
  - data mining (e.g. on the Web)
  - question answering

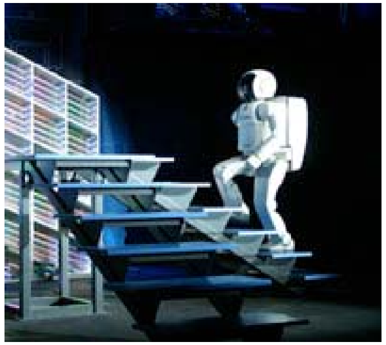
Applications of AI

- Deep Blue
  - Defeats Kasparov, Chess Grand Master - IBM 1997
  - www.research.ibm.com/deepblue
- PEGASUS (Speech understanding for ticketing)
  - www.sls.lcs.mit.edu/sls/applications
- AI in computer games
  - ai.eecs.umich.edu/people/laird/Game-AI-Resources.htm
- information agents
  - question answering (e.g. www.ai.mit.edu/projects/infolab)
  - The Text REtrieval Conference: trec.nist.gov

Applications of AI

- Honda ASIMO www.asimo.com
- unmanned vehicles
  - CMU Autonomous Helicopter (HELI)
- Mars PathFinder rover

  mars.jpl.nasa.gov/MPF/rover/about.html
- RoboCup www.robocup.org
  - robot teams playing football
  - RoboCup rescue
- Sony Aibo www.aibo.com



Course Overview

Objectives

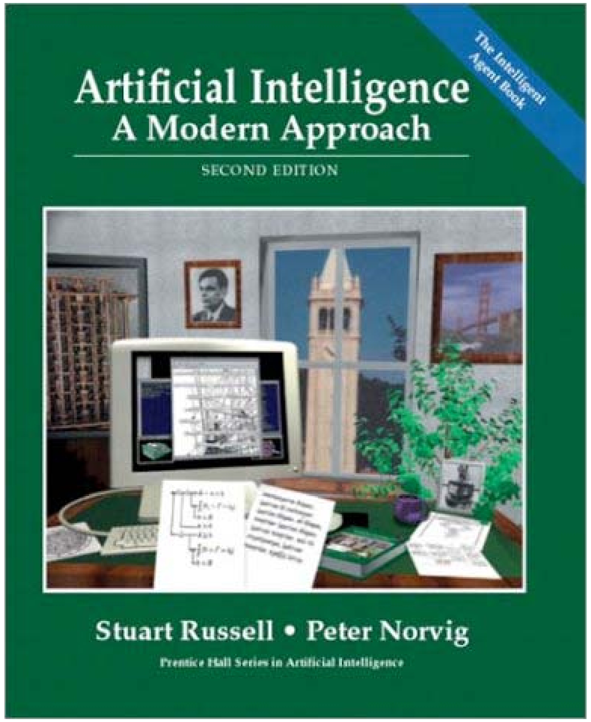
- provide an insight into the fundamental techniques used in AI
  - each topic would require a course by itself
- strong algorithmic perspective
  - you are expected to code
- grounded on mathematical tools
  - much less on cognitive science

Laboratories

- hands on the keyboard
  - implementing the algorithms and techniques discussed during the lectures
- programming language is Java
  - AI programming languages are usually Lisp and Prolog
  - but you can do everything in Java (you just need to be more disciplined)
  - nothing is preventing you to learn Lisp or Prolog
- outcomes of the labs will be part of the assessment
  - not essential, but you will be required more during the final exam

Textbook

Artificial Intelligence: A modern approach
by Stuart Russell and Peter Norvig



- aima.cs.berkeley.edu
- one of the leading books for undergraduate AI courses
- extensive material and source code available from the web site (several programming languages)
- 57th most cited computer science publication ever (source citeseer.nj.nec.com)

Prerequisites (modules)

- Required:
  - Introduction to Programming
  - Algorithms and Complexity

- Suggested:
  - Logic
  - Probability Theory and Statistics

  2nd year students should follow these courses

Prerequisites

to follow this course and pass the exam you need

- a good understanding of algorithms and algorithm design
- not to panic at the appearance of a mathematical formula
- avoid the episodic approach to lessons attendance

Practical issues

- Course slides:
  www.unibz.it/inf/acs/courses/all_03_04/ai
- Course timetable:
  - Thu 8:30-10:30 (E412)
  - Fri 8:30-9:30 (E412)
- Labs timetable:
  - Fri 9:30-11:30 (E431) starting from 17/10/2003
- next week (9,10 October) there are no AI lessons
- Thu 16 October there is no lesson (Industry day)

Agents

What is an Agent?

- an agent can be anything that
  - operates in an environment
  - perceives its environment through sensors
  - acts upon its environment through actuators
  - maximizes progress towards its goals
- conceptual tool to analyse systems:
  - robots, softbots, speed traffic lights, thermostats
- we are interested in Intelligent Agents
  - pursuit goals that require intelligence

Examples of Agents

- human agent
  - eyes, ears, skin, taste buds, etc. for sensors
  - hands, fingers, legs, mouth, etc. for actuators
- robot
  - camera, infrared, bumper, etc. for sensors
  - grippers, wheels, lights, speakers, etc. for actuators
- software agent (softbot)
  - functions as sensors
    - information provided as input to functions in the form of encoded bit strings or symbols
  - functions as actuators
    - results deliver the output

Agent or Program

- our criteria so far seem to apply equally well to software agents and to regular programs
- autonomy
  - agents solve tasks largely independently
  - programs depend on users or other programs for “guidance”
  - autonomous systems base their actions on their own experience and knowledge
  - requires initial knowledge together with the ability to learn
  - provides flexibility for more complex tasks

Agents and Environments

- an agent perceives its environment through sensors
  - the complete set of inputs at a given time is called a percept
  - the current percept, or a sequence of percepts may influence the actions of an agent
- it can change the environment through actuators
  - an operation involving an actuator is called an action
  - actions can be grouped into action sequences

Performance of Agents

- Behavior and performance of IAs in terms of agent function:
  - **Perception history** (sequence) to **Action Mapping:**

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

  - **Ideal mapping:** specifies which actions an agent ought to take at any point in time

- **Performance measure:** a *subjective* measure to characterize how successful an agent is (e.g., speed, power usage, accuracy, money, etc.)

Rationality: do the right thing

- **Rational Action:** The action that maximizes the expected value of the performance measure given the percept sequence to date
  - Rational = Best — Yes, to the best of its knowledge
  - Rational = Optimal — Yes, to the best of its abilities (and its constraints)
  - Rational ≠ Omniscience
  - Rational ≠ Successful
- problems:
  - what is “the right thing”
  - how do you measure the “best outcome”

Omniscience

- a rational agent is not omniscient
  - it doesn't know the actual outcome of its actions
  - it may not know certain aspects of its environment
- rationality takes into account the limitations of the agent
  - percept sequence, background knowledge, feasible actions
  - it deals with the expected outcome of actions

Look it up!

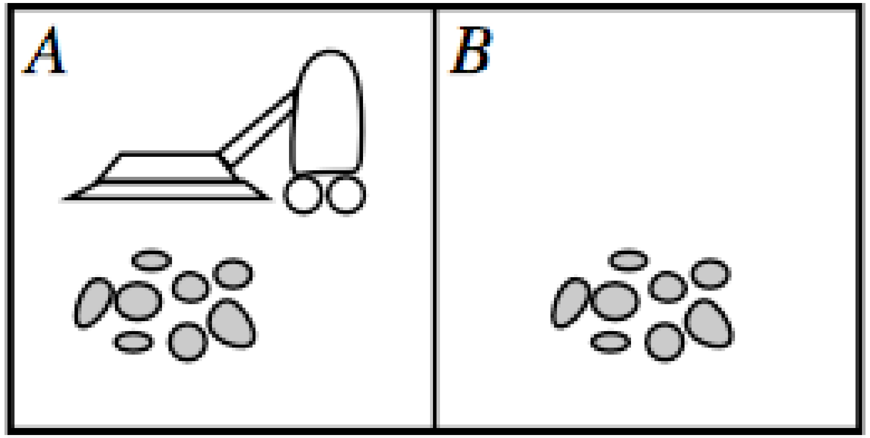
- a table is simple way to specify a mapping from percepts to actions
  - tables may become very large
  - all work done by the designer
  - no autonomy, all actions are predetermined
  - learning might take a very long time

- mapping is implicitly defined by a program
  - rule based
  - neural networks
  - algorithm

Structure of Intelligent Agents

- Agent = architecture + program
- **Agent program:** the implementation of agent's perception-action mapping
- **Architecture:** a device that can execute the agent program (e.g., general-purpose computer, specialized device, robot, etc.)

Vacuum-cleaner world



- Percepts: location+tile status *[A, Dirty], [A, Clean], [B, Clean], [B, Dirty]*
- Actions: *Left, Right, Suck, NoOp*
- Goal: clean the floor

Vacuum-cleaner agent: it sucks!

[A, Clean]	*Right*
[A, Dirty]	*Suck*
[B, Clean]	*Left*
[B, Dirty]	*Suck*
[A,Clean], [A,Clean]	*Right*
[A,Clean], [A,Dirty]	*Suck*
[A,Clean], [B,Clean]	*Left*
[A,Clean], [B,Dirty]	*Suck*
[A,Dirty], [A,Clean]	*Right*
[A,Dirty], [A,Dirty]	*Suck*
...	...
[A,Clean], [A,Clean], [A,Clean]	*Right*
[A,Clean], [A,Clean], [A,Dirty]	*Suck*
...	...

```
if status == Dirty
  then Suck
else
  if location == A
    then Right
  else Left
```

Performance Evaluation

vacuum agent

- number of tiles cleaned during a certain period
  - based on the agent's report, or validated by an objective authority
  - doesn't consider expenses of the agent, side effects
    - energy, noise, loss of useful objects, damaged furniture, scratched floor
  - might lead to unwanted activities
    - agent re-cleans clean tiles, covers only part of the room, drops dirt on tiles to have more tiles to clean, etc.

Cleaning Robots

- Cleaning Robot contest
  - http://www.service-robots.org/cleaningrobotscontest/

Software Agents

- also referred to as "softbots"
- live in artificial environments where computers and networks provide the infrastructure
- may be very complex with strong requirements on the agent
  - World Wide Web, real-time constraints,
- natural and artificial environments may be merged
  - user interaction
  - sensors and actuators in the real world
    - camera, temperature, arms, wheels, etc.

Mobile agents

- Programs that can migrate from one machine to another
- Execute in a platform-independent execution environment
- Require agent execution environment (places)
- Mobility not necessary or sufficient condition for agenthood
- Practical but non-functional advantages:
  - Reduced communication cost (eg, from PDA)
  - Asynchronous computing (when you are not connected)
- Applications:
  - Distributed information retrieval
  - Telecommunication network routing

Information agents

- Manage the explosive growth of information
- Manipulate or collate information from many distributed sources
- Information agents can be mobile or static
- information on the Web or in document corpora
  - ontologies for annotating Web pages (services)
  - data mining on unstructured data
  - question answering using knowledge intensive of statistical methods

Environments

- determine to a large degree the interaction between the “outside world” and the agent
  - the “outside world” is not necessarily the “real world” as we perceive it
- in many cases, environments are implemented within computers
  - they may or may not have a close correspondence to the “real world”

Environment Properties

- Fully observable vs. partially observable
  - Fully observable: sensors can detect all aspects of the environment
  - Effectively fully observable: relevant aspects
- Deterministic vs. stochastic
  - Deterministic: next state determined by current state and the agent' actions
  - Partial observable could be stochastic from the agent's view point
- Episodic vs. sequential
  - Agent's experience divided into episodes; subsequent episode do not depend on actions in previous episodes
- Static vs. dynamic
  - Dynamic: Environment changes while agent is deliberating
  - Semi-dynamic: environment static, performance scores dynamic
- Discrete vs. continuous
  - Discrete: Finite number of percepts and actions
- Single agent vs. multi-agent
  - Competitive, cooperative, and communication

Environment types

Environment	Observable	Deterministic	Episodic	Static	Discrete
Vacuum cleaner	Yes	Yes	Yes	Yes	Yes
Virtual Reality	Yes	Yes	Yes/No	No	Yes
Internet shopping	No	No	No	No	Yes

- agent design is mainly influenced by the environment
- often the abstraction influences the description of the environment
- Real world is
  - partially observable, stochastic, sequential, dynamic, continuous

Environment Programs

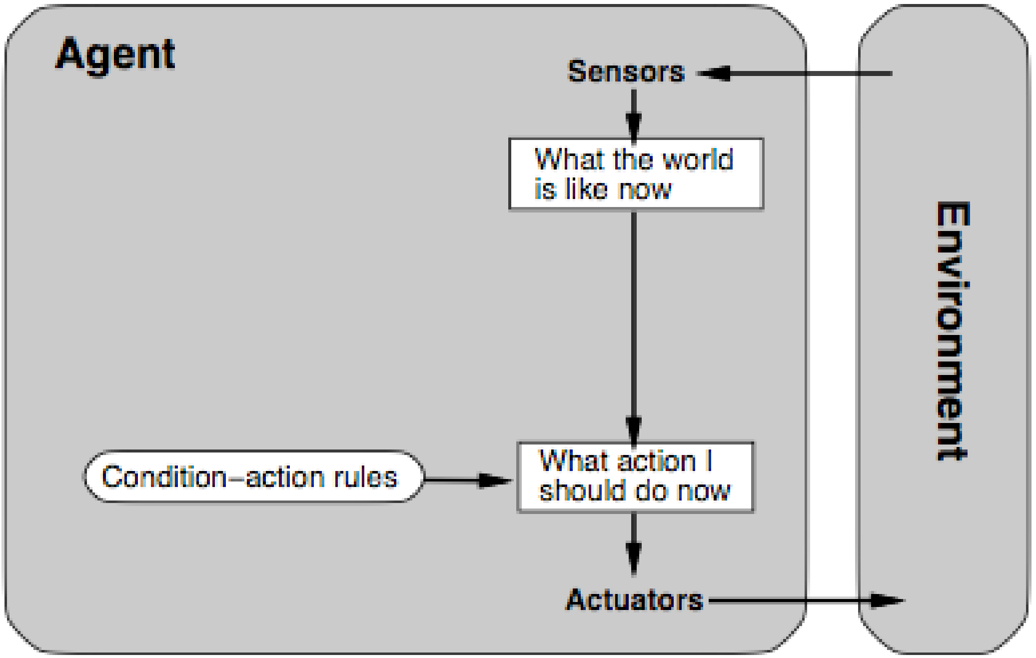
- environment simulators for experiments with agents
  - gives a percept to an agent
  - receives an action
  - updates the environment
- often divided into environment classes for related tasks or types of agents
- frequently provides mechanisms for measuring the performance of agents

Agent types

- Four basic types in order of increasing generality
  - simple reflex agents
  - model based reflex agents (with state)
  - goal-based agents
  - utility-based agents
- All these can be turned into learning agents

Simple reflex agents

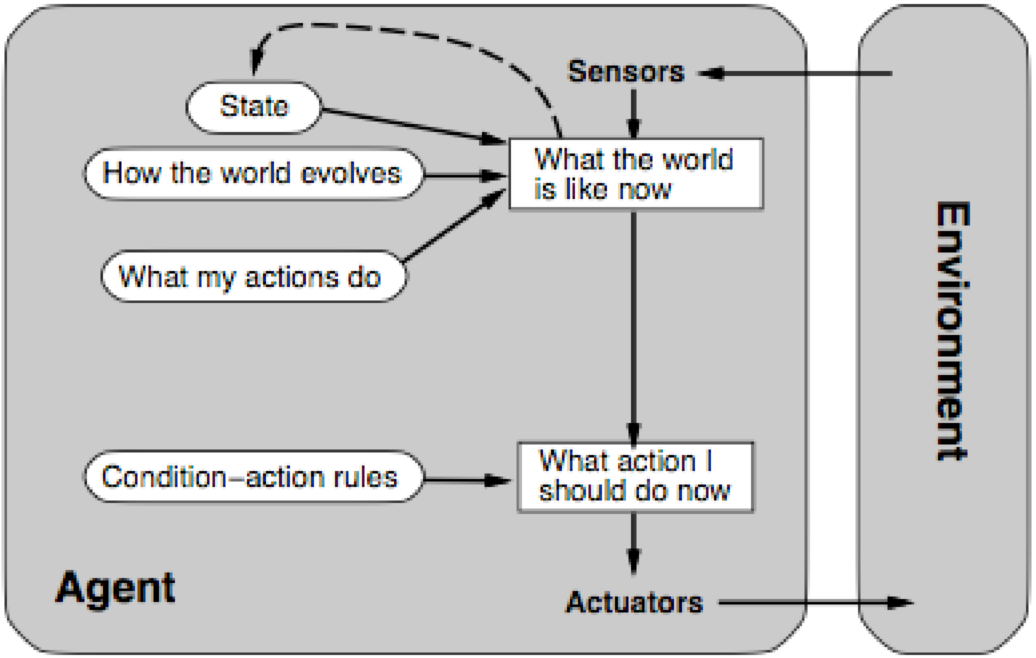
- Simple look-up table, mapping percepts to actions, is out of the question (too large, too expensive to build)
- Many situations can be summarized by condition-action rules (humans: learned responses, innate reflexes)



- Implementation: easy; Applicability: narrow

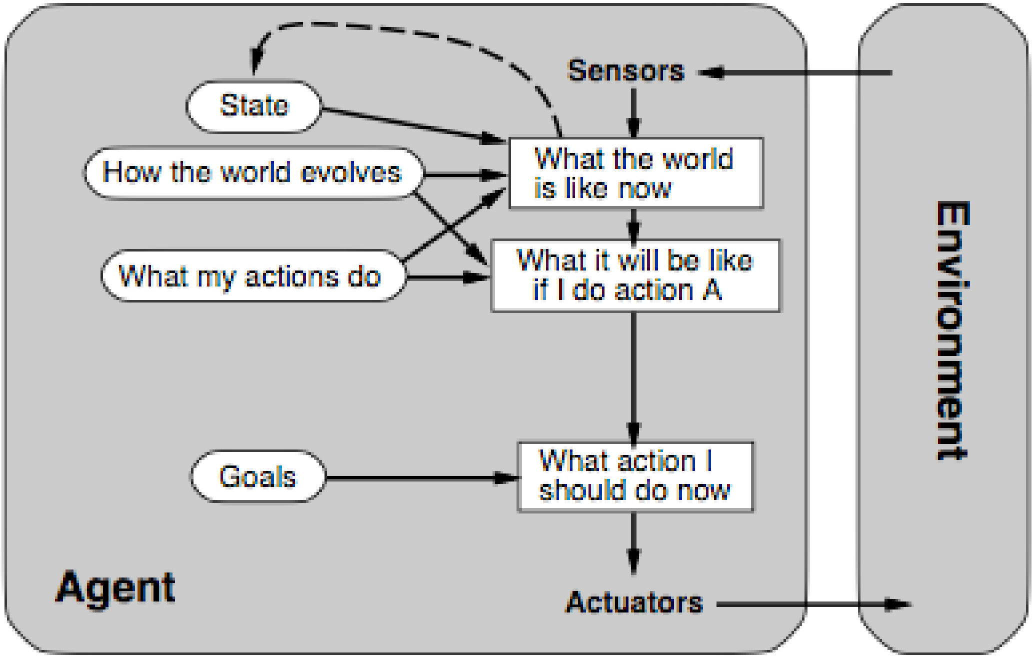
Model-based reflex agents (with state)

- Sensor information alone is not sufficient in case of partial observability
- Need to keep track of how the world evolves
  - Evolution: independently of the agent, or caused by the agent's action
  - Knowledge about how the world works – Model of the world



Goal-based agents

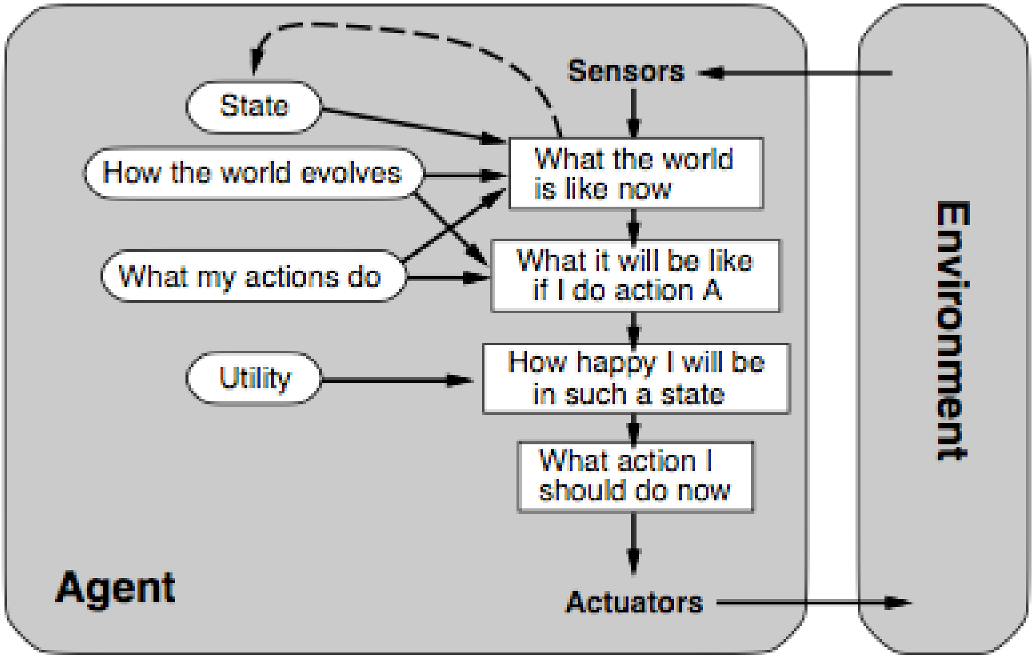
- State and actions don't tell where to go
- Need goals to build sequences of actions (planning)



- Goal-based: uses the same rules for different goals
- Reflex: will need a complete set of rules for each goal

Utility-based agents

- Several action sequences to achieve some goal (binary process)
- Need to select among actions and sequences (preferences)
- Utility: state → real number
  - express degree of satisfaction and specify trade-offs between conflicting goal



Learning agents

- Learning element: making improvements
- Performance element: selecting external actions (entire former agents)
- Critic: collecting feedback on how the agent is doing?
- Problem generator: suggesting (exploratory) actions (experiments)

Learning agents