Artificial Intelligence
CS 6364

Professor Dan Moldovan

Section 1

Introduction to AI

Outline

- Introduction to AI
- Intelligent Agents

Definition

AI is the study of computations that make it possible to perceive, reason and act. (Winston)

One may say that AI is: computational reasoning

Notice: 1) Definition implies computations, programs, computers (computer science and engineering)

2) Definition implies perception, reasoning, action—which are human qualities

A bad definition is: AI is the study of computers that perform human-like activities.

This includes emotions, feelings, smelling, etc. which are not part of AI (at least not now).

Other definitions of AI

- Systems that think like humans
  - Cognitive modeling approach
- Systems that act like humans
  - Activities that require human intelligence: communicate, reason, learn, plan, etc.
  - Turing test
- Systems that think rationally
  - Activities related to human thinking, arriving to logical conclusions
- Systems that act rationally
  - Activities related to rational human activities, that have a purpose

Some Task Domains of AI

I. **Mundane Tasks**

Perception
- Vision
- Speech

Natural Language
- Understanding
- Generation
- Translation

Planning

Commonsense reasoning

Robot control

II. **Formal Tasks**

Games
- Chess, backgammon, etc.

Mathematics
- Logic, geometry, calculus

III. **Expert Tasks**

Engineering
- Design
- Manufacturing planning

Scientific analysis

Medical diagnosis

Financial Analysis

Some successes and Failures of AI

- Successes
  - Expert systems
    - Nonlinear dynamics
    - Medicine
    - CAD
  - Learning from examples
    - Discovering a new star
  - Robot control, planning
    - Many robots are built using AI
  - Natural Language
    - Information retrieval
    - Translation systems
  - AI used in combination with Software Engineering in many applications
  - Military applications
- Failures
  - Most systems are toy systems working in some small domains
  - AI did not deliver in 1980's as it was expected. 5th Generation computers (project that started in Japan and picked up in US failed to build an AI computer—of good quality ⇒ It was an "AI winter".

Fundamental Hypothesis of AI

Intelligent systems can be constructed from explicit, declarative knowledge bases which in turn are operated on by general, formal reasoning mechanisms.

Thus, of central importance to the field of AI is:

- Knowledge representation and
- Reasoning

Intelligence is knowledge.

AI systems are synonymous of knowledge-based systems.

Leibnitz (1646-1716) wanted to mechanize intelligence. He dreamed of a calculus of ideas, where truth could be determined by manipulating an "alphabet of thoughts" (characteristica universalis). This was supposed to be similar with Newton's calculus which manipulates numbers to preserve value. Leibnitz' calculus was supposed to preserve the truth.

Examples of Computational Inferences

An inference is a (reasoning) process that provides new facts unstated in a knowledge base.

Example 1. Consider the sentences

S1: John hit the ball with a bat.

S2: It landed far away.

An inference system would provide first a correct interpretation of these two sentences. One interpretation (most plausible) is that a male named John held a club in his hands, and the club touched with some force a round object, the ball, which then moved through the air and finally fell to the ground due to gravitational force.

To build a machine to provide such interpretations (inferences) for broad English domain is a monumental task.

Examples of Computational Inferences

Example 2. (more modest)

Consider that we are building a KB system to assist in providing a medical treatment. We want to include which patients are allergic to which medications. A simple database (DB) may be:

S1: Ralph is allergic to sulfa.

S2: Trixie is allergic to penicillin.

S3: Alice is allergic to tetracycline, etc.

Simple retrieval operation will tell us whether or not to use a drug in a therapy. However, drugs are related to one another:

S4: All patients allergic to penicillin are also allergic to ampicillin.

From these statements we conclude that:

S5: Trixie is allergic to ampicillin.

This is a new fact not stated in the KB, and cannot be simply retrieved from KB.

The truth of a conclusion is implicit in the truth of the two premises. We say that conclusion is entailed by the premises.

Computing Entailments (Inferences)

$\models$ means entails

KB $\models$ α means that α is entailed from the KB.

α is a truth that may be derivable from the KB. We need rules of inference that allow us to extract truth from KB.

The central goal of AI is to build KB systems capable of determining truths about the world implied by the contents of the KB, that is go beyond explicitly stated facts.

Some differences between Databases and KB.

DB	KB
– Stores simple facts – In DB there is no notion of entailment (only complex queries)	– Stores complex facts – KB deals with incomplete info. There is an inference mechanism that extracts new facts not stated explicitly in KB.

Computational Effectiveness

- The field of AI is complex because intelligent behavior is complex. AI systems were built from limited domains. (Examples of systems in limited domains).
- It is not possible to expand from limited domains to larger, more general domains by simply adding more facts and rules.

  Intelligence is not a linear function. Solutions do not scale up.

- Computational effectiveness is how to interconnect many domains to make more complex domains.
- For a system to be intelligent, it must be capable of acting within some specified time. The required time is usually imposed by the environment. This is sometimes called computational effectiveness constraint. It is a basic property of intelligent systems.

Computational Effectiveness

- Parallel processing provides speed, and speed helps with computational effectiveness.

  Open question: Can speed (PP) provide intelligence?

- Problems with von Neumann architectures when used in AI—bottleneck. Only an insignificant portion of system's knowledge participates in processing. Intelligent behavior requires frequent interactions between many pieces of information. Memory cells should not only hold information, but be able to process, and exchange information.

Evidential Reasoning

- Often, information available is incomplete. The form of reasoning where inferences do not establish the truth of a proposition, but instead find most likely hypothesis among a finite set of alternatives is called evidential reasoning. This leads to non-monotonic, or probabilistic truth values.

- Important for evidential reasoning is the ability to handle best matches, or partial match operations. Also, treatment of exceptions, multiple inheritances or conflicting information may be necessary.

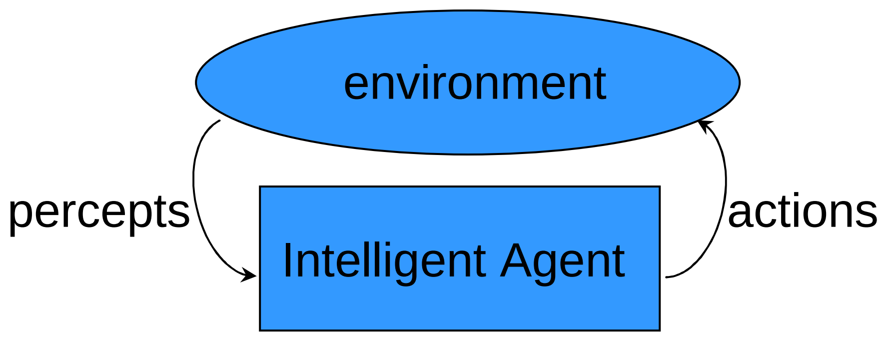
Intelligent Agents

An agent is a thing (e.g. program, or system) that can be viewed as perceiving its environment and acting upon that environment through effectors.

The notion of agent is useful because it provides a tool for analyzing AI systems.

What do we want from agents?

To act rationally.



Intelligent Agents

- Agents:
  - Perceive the environment through sensors.
  - Perform actions.

Intelligence is based on these two: perception and capability of actions.

Agents must perform actions to obtain useful information (acquire more percepts).

An agent may be regarded as a mapping of percepts into actions.

- A performance measure is needed to determine how successful an agent is.

An <u>ideal rational agent</u> provides an action that maximizes its performance measure, based on the percepts received and its built-in knowledge.

Autonomous agents

We want to build intelligent agents capable of reacting to the environment on their own. That is to say that we want agents to be able to learn from their experiences and face new situations.

This is different than programming extensively, hoping that the programmer foresaw all possible situations.

An agent is autonomous when its behavior is determined by its own experience. (autonomous from other agents or humans)

Example: a clock adjusting to time zones.

Intelligent Agents vs Other Software

- Agents are autonomous
- Agents contain some level of intelligence
- Agents react to environments, and can sometimes take proactive actions
- Agents may have social ability; ie communicate with user and other agents
- Agents may cooperate with other agents
- Agents may migrate from system to system

Structure of Intelligent Agents

Agents are built with programs running on some hardware. AI systems may be general-purpose computers, or special-purpose computers.

Agent programs are very complex, because intelligence is complex. Simple look up tables mapping percepts into actions will not work.

Example: A chess player would require 35^{100} entries.

Example: Automated Taxi Driver

Agent Type	Performance Measure	Environment	Actuators	Sensors
Taxi driver	Safe, fast, legal, comfortable trip, maximize profits	Roads, other traffic, pedestrians, customers	Steering, accelerator, brake, signal, horn, display	Cameras, sonar, speedometer, GPS, odometer, accelerometer, engine sensors, keyboard

Agent Types and their Descriptions

Agent Type	Performance Measure	Environment	Actuators	Sensors
Medical diagnosis system	Healthy patient, minimize costs, lawsuits	Patient, hospital, staff	Display 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 categorization of scene	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	Maximize purity, yield, safety	Refinery, operators	Valves, pumps, heaters, displays	Temperature, pressure, chemical sensors
Interactive English tutor	Maximize student's score on test	Set of students, testing agency	Display exercises, suggestions, corrections	Keyboard entry

Properties of environments

1. Accessible vs inaccessible

   Accessible when the sensors detect all aspects that are relevant to the choice of action.
2. Deterministic vs nondeterministic

   When the next state of environment is completely determined from the current state and the actions selected by agents.
3. Episodic vs nonepisodic

   Episodic environment is when agent's experience is divided into "episodes". Agent perceives an episode then acts, and is done with it. Subsequent episodes do not depend on the episode itself.
4. Static vs dynamic

   Dynamic is when environment changes while agent deliberates and acts.
5. Discrete vs continuous

   Descrete is when there is a limited number of distinct, clearly defined percepts and actions.

Note: The hardest are: inaccessible, nonepisodic, dynamic and continuous.

Properties of environments: examples

Task Environment	Observable	Deterministic	Episodic	Static	Discrete	Agents
Crossword puzzle	Fully	Deterministic	Sequential	Static	Discrete	Single
Chess with a clock	Fully	Strategic	Sequential	Semi	Discrete	Multi
Poker	Partially	Strategic	Sequential	Static	Discrete	Multi
Backgammon	Fully	Stochastic	Sequential	Static	Discrete	Multi
Taxi driving	Partially	Stochastic	Sequential	Dynamic	Continuous	Multi
Medical diagnosis	Partially	Stochastic	Sequential	Dynamic	Continuous	Single
Image-analysis	Fully	Deterministic	Episodic	Semi	Continuous	Single
Part-picking robot	Partially	Stochastic	Episodic	Dynamic	Continuous	Single
Refinery controller	Partially	Stochastic	Sequential	Dynamic	Continuous	Single
Interactive English tutor	Partially	Stochastic	Sequential	Dynamic	Discrete	Multi

The Structure of Intelligent Agents

The question facing the designer is: How to structure (architect) such an agent?

Four types of agents (in increasing complexity order)

- Simple reflex agents
- Agents that keep track of the world
- Goal-based agents
- Utility-based agents

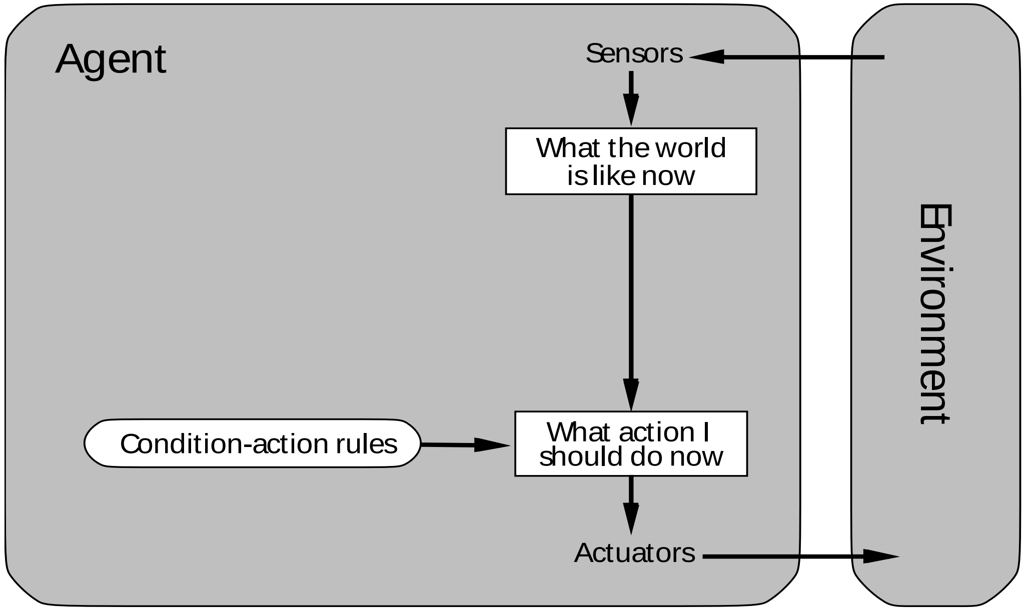
Simple reflex agents

- These are implemented with condition-action rules.

  <u>if</u> condition <u>then</u> action

  <u>if</u> car-in-front-is-braking <u>then</u> initiate-braking
- Rule conditions are compared with the current situation to determine which rule is applicable; once found, a rule is applied, meaning that some action is triggered.
- Environments need be fully observable

Simple reflex agents



Schematic diagram of a simple reflex agent.

function SIMPLE-REFLEX-AGENT(*percept*) **returns** an action
  static: *rules*, a set of condition–action rules

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

A simple reflex agent. It works by finding a rule whose condition matches the current situation (as defined by the percept) and then doing the action associated with that rule.

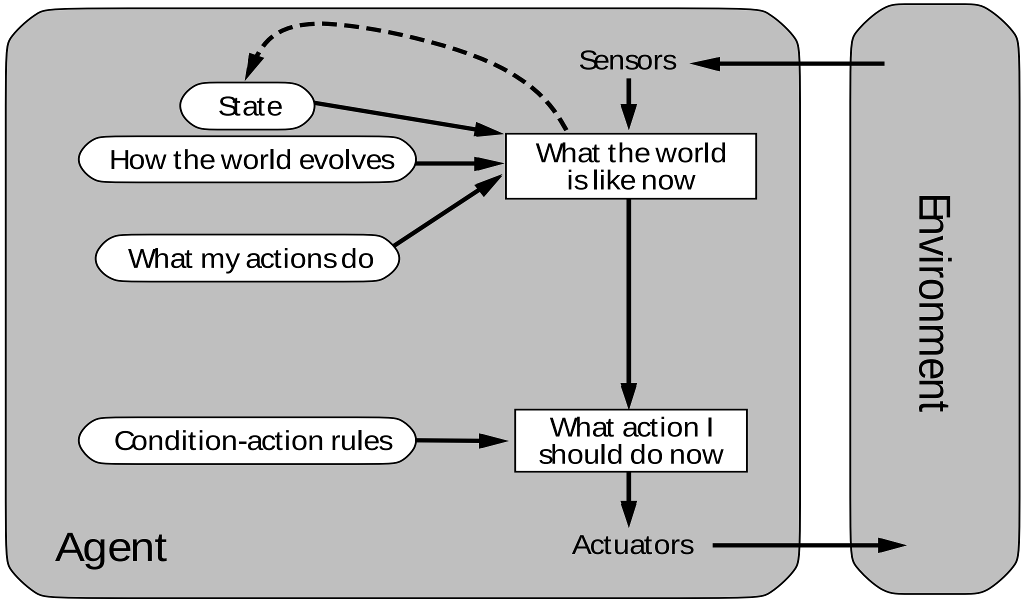
Agents that keep track of the world (agents with internal states)

- In this case, rules are more complex; rule conditions take into account internal states of agents. By keeping some internal states, agents enhance their ability to react to environment. (To distinguish between states is fundamental in nature; recall states in logic design) Internal states capture info about how the world evolves and agent acts.

- Also called model-based agents

- Environments can be partially observable

Agents that keep track of the world (agents with internal states)



A reflex agent with internal state.

function REFLEX-AGENT-WITH-STATE(*percept*) **returns** an action
  static: *state*, a description of the current world state
    rules, a set of condition–action rules
    action, the most recent action, initially none

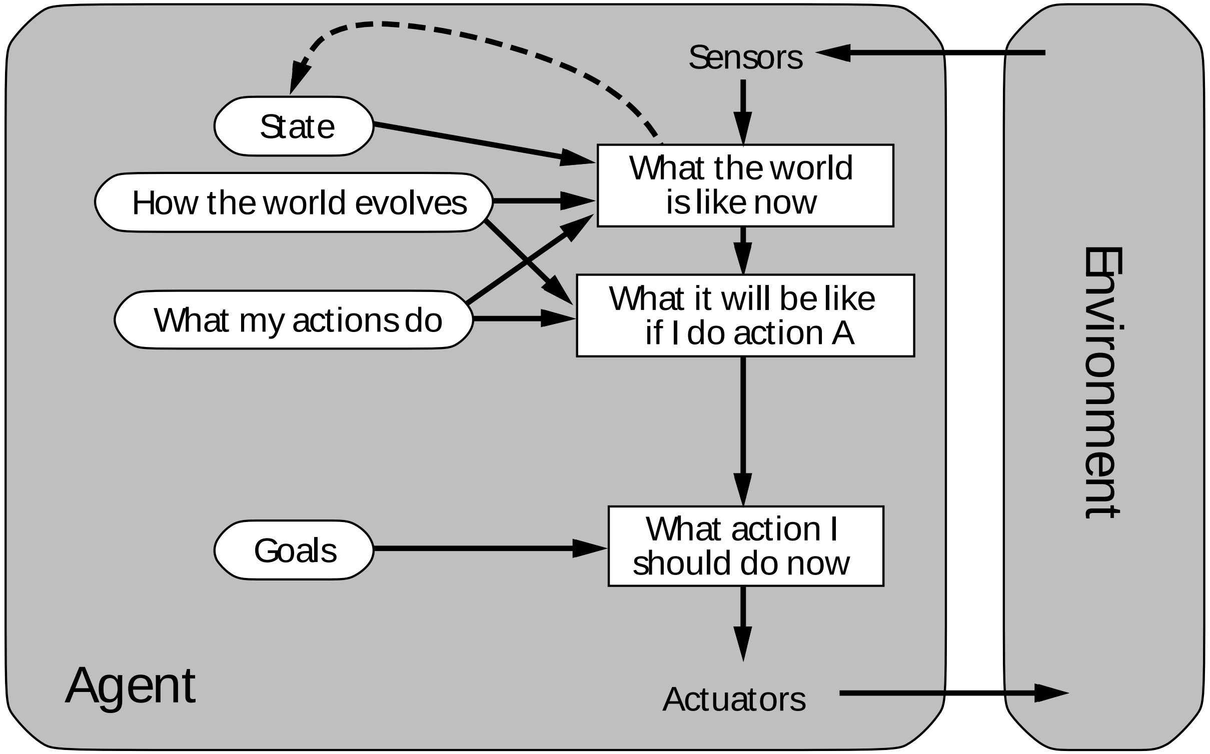
  state ← UPDATE-STATE(*state*, *action*, *percept*)
  rule ← RULE-MATCH(*state*, *rules*)
  action ← RULE-ACTION[*rule*]
  return *action*

A reflex agent with internal state. It works by finding a rule whose condition matches the current situation (as defined by the percept and the stored internal state) and then doing the action associated with that rule.

Goal-based agents

- Another enhancement is to give the agent a goal to look for. The agent actions constitute a sequence that leads to the goal.
  - Application examples: Searching, Planning.
- Goal information describes desirable situations.
- The actions now are not provided by if-then rules, but they are selected such that will bring the system closer to the goal.
- States are still necessary.
- Note that the way actions are selected to get closer to the goal, may still use if-then rules (but not exclusively).

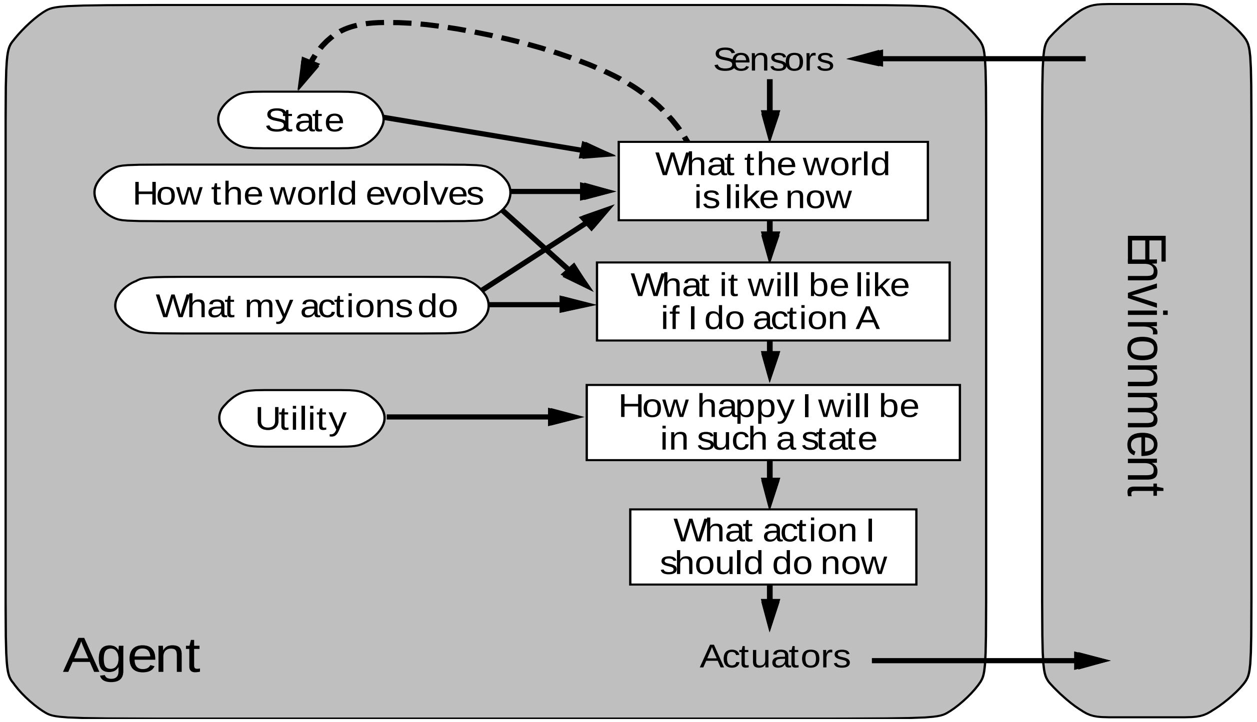
Goal-based agents



A complete utility-based agent.

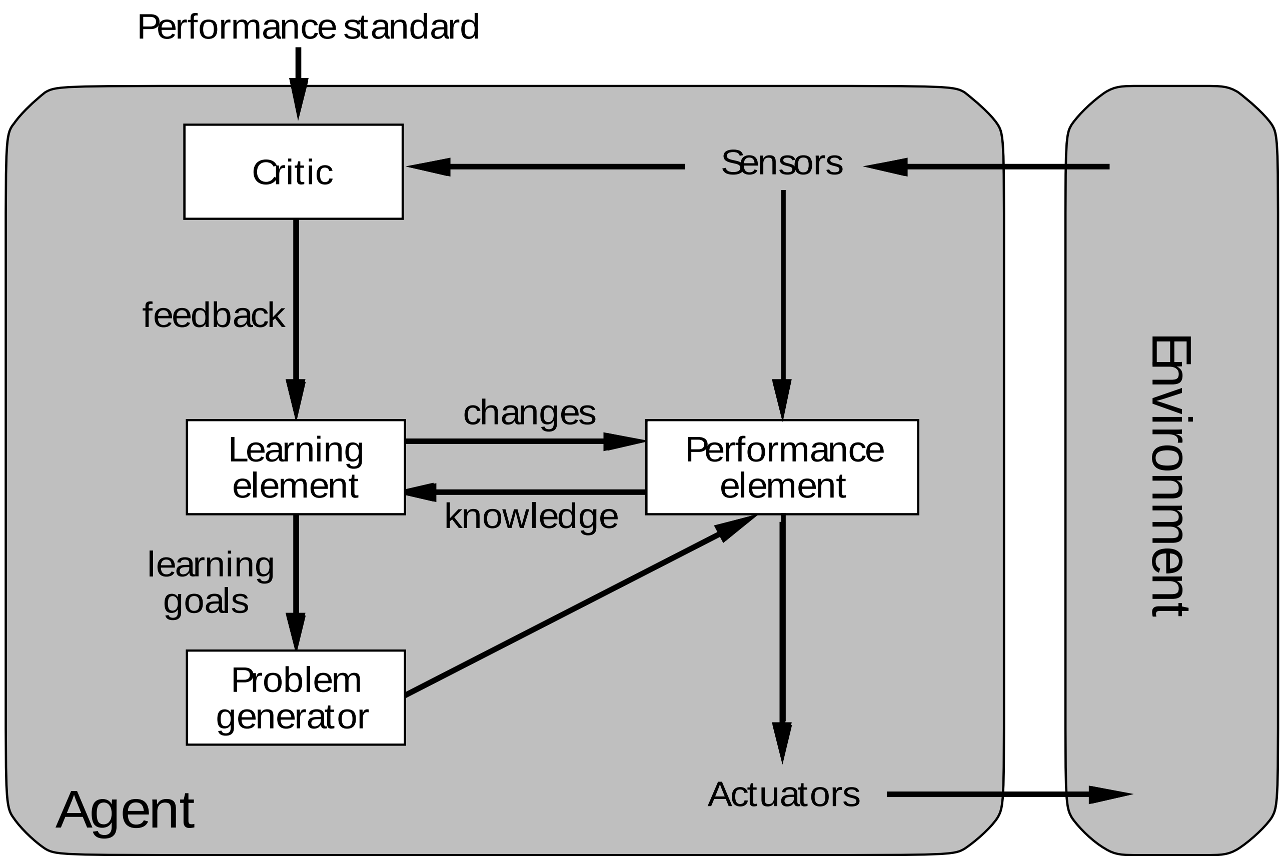
Utility-based agents

The idea is to provide a metric to assist the agent in moving toward the goal faster.
A utility function maps the states into a performance number.



An agent with explicit goals.

Learning Agents



A general model of learning agents.