"Artificial Intelligence: A Modern Approach" - S. Russell and P. Norvig,

Chapter 1

Introduction – Foundation of Artificial Intelligence – Intelligent Agents – structure of agents – Definitions of a rational agent, reflex, model-based, goal-based, and utility-based agents, the environment in which a particular agent operates

UNIT 1

Big Questions

- Can machines think?
- And if so, how?
- And if not, why not?
- And what does this say about human beings?

What is artificial intelligence?

There are no clear consensus on the definition of AI

Q. What is artificial intelligence?

A. It is the science and engineering of making intelligent machines, especially intelligent computer programs. It is related to the similar task of using computers to understand human intelligence, but AI does not have to confine itself to methods that are biologically observable.

Q. Yes, but what is intelligence?

A. Intelligence is the computational part of the ability to achieve goals in the world. Varying kinds and degrees of intelligence occur in people, many animals and some machines.

Other possible AI definitions

- AI is a collection of hard problems which can be solved by humans and other living things, but for which we don't have good algorithms for solving.
- e. g., understanding spoken natural language, medical diagnosis, learning, self-adaptation, reasoning, chess playing, proving math theories, etc.
- System that Act like human (Turing test)
  - Natural Language processing for communication
  - Knowledge representation to store information
  - Automated Reasoning to use stored information
  - Machine Learning to adapt to new circumstances and to detect and extrapolate (to infer) patterns

Other possible AI definitions

- **System that Think like human** (human-like patterns of thinking steps)

  –Through Introspection trying to catch our own thoughts as they go by.

  –Through psychological experiments.

  Once we have a sufficiently precise theory of the mind it becomes possible to express the theory as a computer program.

- **System that think rationally** (logically, correctly)

Aristotle was one of the first to attempt to codify "right Thinking" which is called **Syllogisms**

<u>**Example**</u> Socrates is a man
all men are mortal
therefore Socrates is mortal

- **Acts Rationally** acting so as to achive one's goal given one's beliefs.

What’s easy and what’s hard?

- It’s been easier to mechanize many of the high level cognitive tasks we usually associate with “intelligence” in people
  - e. g., symbolic integration, proving theorems, playing chess, some aspect of medical diagnosis, etc.
- It’s been very hard to mechanize tasks that animals can do easily
  - walking around without running into things
  - catching prey and avoiding predators
  - interpreting complex sensory information (visual, aural, …)
  - modeling the internal states of other animals from their behavior
  - working as a team (ants, bees)

The foundations of Artificial Intelligence

1. Philosophy(428 B.C. – present)

Aristotle (384-322 B.C.) was the first to formulate a precise set of laws governing the rational part of the mind. He developed an informal system of syllogisms for proper reasoning, which allowed one to generate conclusions mechanically, given initial premises.

	Computer	Human Brain
Computational units	1 CPU,10^8 gates	10^{11} neurons
Storage units	10^{10} bits RAM 10^{11} bits disk	10^{11} neurons 10^{14} synapses
Cycle time	10^{-9} sec	10^{-3} sec
Bandwidth	10^{10} bits/sec	10^{14} bits/sec
Memory updates/sec	10^9	10_{14}

The foundations of Artificial Intelligence

2.Psycology (1879 – present)

The origin of scientific psychology are traced back to the wok if German physiologist Hermann von Helmholtz (1821-1894) and his student Wilhelm Wundt(1832 – 1920). In 1879,Wundt opened the first laboratory of experimental psychology at the university of Leipzig. In US, the development of computer modeling led to the creation of the field of **cognitive science**. The field can be said to have started at the workshop in September 1956 at MIT.

3.Computer engineering (1940-present)

For artificial intelligence to succeed, we need two things: intelligence and an artifact. The c computer has been the artifact of choice.

A1 also owes a debt to the software side of computer science, which has supplied the operating systems, programming languages, and tools needed to write modern programs

The foundations of Artificial Intelligence

4. Control theory and Cybernetics (1948-present)
Ktesibios of Alexandria (c. 250 B.C.) built the first self-controlling machine: a water clock with a regulator that kept the flow of water running through it at a constant, predictable pace.

Modern control theory, especially the branch known as stochastic optimal control, has as its goal the design of systems that maximize an objective function over time.

5. Linguistics (1957-present)
Modem linguistics and AI, then, were "born" at about the same time, and grew up together, intersecting in a hybrid field called computational linguistics or natural language processing.

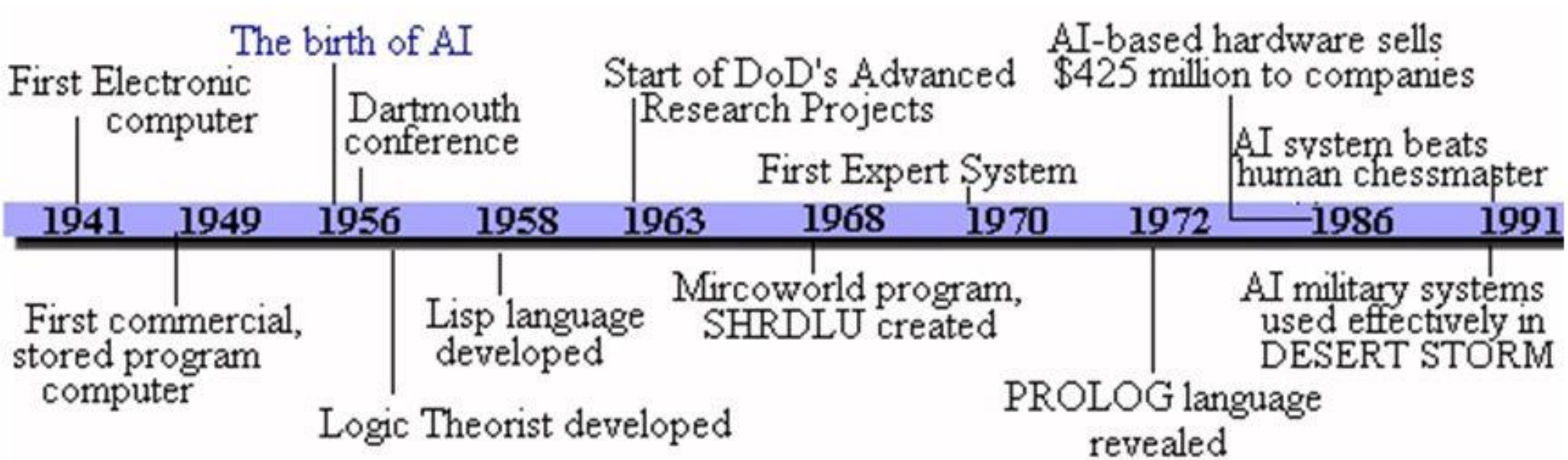
History of AI

- AI has roots in a number of scientific disciplines
  - computer science and engineering (hardware and software)
  - philosophy (rules of reasoning)
  - mathematics (logic, algorithms, optimization)
  - cognitive science and psychology (modeling high level human/animal thinking)
  - neural science (model low level human/animal brain activity)
  - Linguistics, economics, etc.
- The birth of AI (1943 – 1956)
  - Pitts and McCulloch (1943): simplified mathematical model of neurons (resting/firing states) can realize all propositional logic primitives (can compute all Turing computable functions)
  - Allen Turing: Turing machine and Turing test (1950)
  - Claude Shannon: information theory; possibility of chess playing computers

- Early enthusiasm (1952 – 1969)
  - 1956 Dartmouth conference
    John McCarthy (Lisp);
    Marvin Minsky (first neural network machine);
    Alan Newell and Herbert Simon (GPS) General Problem Solver;
  - Emphasize on intelligent general problem solving
    GSP (means-ends analysis);
    Lisp (AI programming language);
    Resolution by John Robinson (basis for automatic theorem proving);
    heuristic search (A*, AO*, game tree search)
- Emphasis on knowledge (1966 – 1974)
  - Domain specific knowledge is the key to overcome existing difficulties
  - Knowledge representation (KR) paradigms
  - Declarative vs. procedural representation

- Knowledge-based systems (1969 – 1979)
  - DENDRAL: the first knowledge intensive system (determining 3D structures of complex chemical compounds)
  - MYCIN: first rule-based expert system (containing 450 rules for diagnosing blood infectious diseases)
    EMYCIN: an ES shell
  - PROSPECTOR: first knowledge-based system that made significant profit (geological ES for mineral deposits)
- AI became an industry (1980 – 1989)
  - wide applications in various domains
  - commercially available tools
- Current trends (1990 – present)
  - more realistic goals
  - more practical (application oriented)
  - distributed AI and intelligent software agents
  - resurgence of neural networks and emergence of genetic algorithms

History


The birth of AI
First Electronic computer
Dartmouth conference
Start of DoD's Advanced Research Projects
First Expert System
AI-based hardware sells $425 million to companies
AI system beats human chessmaster
1941
1949
1956
1958
1963
1968
1970
1972
1986
1991
First commercial, stored program computer
Logic Theorist developed
Lisp language developed
Mircoworld program, SHRDLU created
PROLOG language revealed
AI military systems used effectively in DESERT STORM

The state of art: What can A1 do today?

Autonomous planning and Scheduling: NASA's Remote Agent program became the first on-board autonomous planning program to control the scheduling of operations for a spacecraft

Game playing: IBM's Deep Blue became the first computer program to defeat the world champion in a chess

Autonomous control: The ALVINN computer vision system was trained to steer a car to keep it following a lane

Diagnosis: Medical diagnosis programs based on probabilistic analysis have been able to perform at the level of an expert physician in several areas of medicine.

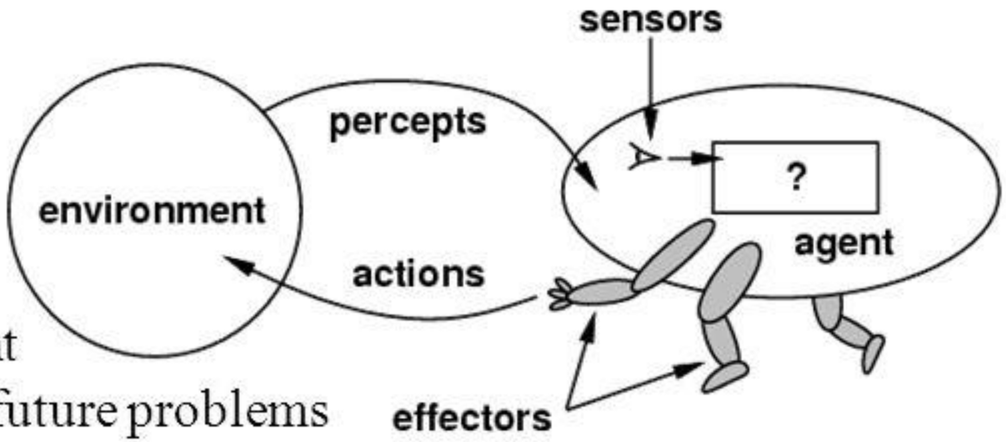
Logistics Planning: During the Persian Gulf crisis of 1991, U.S. forces deployed a Dynamic Analysis and Replanning Tool, DART (Cross and Walker, 1994), to do automated logistics planning and scheduling for transportation

Robotics: Many surgeons now use robot assistants in microsurgery.
HipNav (DiGioia et al., 1996) is a system that uses computer vision techniques

Language understanding and problem solving:PROVERB (Littman et al., 1999) is a computer program that solves crossword puzzles better than most humans, using constraints on possible word fillers, a large database of past puzzles, and a variety of information sources including dictionaries and online databases such as a list of and the actors that appear in them.

Chapter 2: Intelligent Agents

- Definition: An Intelligent Agent perceives it environment via sensors and acts rationally upon that environment with its effectors.
- Hence, an agent gets percepts one at a time, and maps this percept sequence to actions.
- Properties
  - Autonomous
  - Interacts with other agents plus the environment
  - Reactive to the environment
  - Pro-active (goal- directed) future problems



Agents and environments

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

A **human agent** has eyes, ears, and other organs for sensors and hands, legs, mouth, and other body parts for actuators.

A **robotic agent** might have cameras and infrared range finders for sensors and various motors for actuators.

A **software agent** receives keystrokes, file contents, and network packets as sensory inputs and acts on the environment by displaying on the screen, writing files, and sending network packets.

What do you mean, sensors/percepts and effectors/actions?

- Humans
  - Sensors: Eyes (vision), ears (hearing), skin (touch), tongue (gustation), nose (olfaction).
  - Percepts:
    - At the lowest level – electrical signals from these sensors
    - After preprocessing – objects in the visual field (location, textures, colors, …), auditory streams (pitch, loudness, direction), …
  - Effectors: limbs, digits, eyes, tongue, …
  - Actions: lift a finger, turn left, walk, run, carry an object, …
- The Point: percepts and actions need to be carefully defined, possibly at different levels of abstraction

A more specific example: Automated taxi driving system

- **Percepts**: Video, sonar, speedometer, odometer, engine sensors, keyboard input, microphone, GPS, …
- **Actions**: Steer, accelerate, brake, horn, speak/display, …
- **Goals**: Maintain safety, reach destination, maximize profits (fuel, tire wear), obey laws, provide passenger comfort, …
- **Environment**: U.S. urban streets, freeways, traffic, pedestrians, weather, customers, …
- **Different aspects of driving may require different types of agent programs!**

Rationality

- An ideal **rational agent** should, for each possible percept sequence, do whatever actions that will maximize its performance measure based on
  - (1) the percept sequence, and
  - (2) its built-in and acquired knowledge.
- Hence it includes information gathering, not "rational ignorance."
- Rationality => Need a performance measure to say how well a task has been achieved.

Autonomy

- A system is autonomous to the extent that its own behavior is determined by its own experience and knowledge.
- Therefore, a system is not autonomous if it is guided by its designer according to a priori decisions.
- To survive agents must have:
  - Enough built- in knowledge to survive.
  - Ability to learn.

Task environments

- We must think about **task environments,** which are essentially the "**problems**" to which rational agents are the "**solutions**."
- **Specifying the task environment**

  The rationality of the simple vacuum-cleaner agent, needs specification of

  the performance measure

  the environment

  the agent's actuators and sensors.

PEAS

All these are grouped together under the heading of the **task environment.**

We call this the **PEAS** (Performance, Environment, Actuators, Sensors) description.

In designing an agent, the first step must always be to specify the task environment as fully as possible.

Properties of task environments

- Fully observable vs. partially observable
- Deterministic vs. stochastic
- Episodic vs. sequential
- Static vs. dynamic
- Discrete vs. continuous
- Single agent vs. multiagent

Fully observable vs. **partially observable.**
If an agent's sensors give it access to the complete state of the environment at each point in time, then we say that the task environment is fully observable. A task envi- ronment is effectively fully observable if the sensors detect all aspects that are *relevant* to the choice of action;
An environment might be partially observable because of noisy and inaccurate sensors or because parts of the state are simplly missing from the sensor data.
Deterministic vs. **stochastic.**
If the next state of the environment is completely determined by the current state and the action executed by the agent, then we say the environment is deterministic; other- wise, it is stochastic.

- **Episodic** vs. **sequential**

In an **episodic task environment**, the agent's experience is divided into atomic episodes. Each episode consists of the agent perceiving and then performing a single action. Crucially, the next episode does not depend on the actions taken in previous episodes.
For example, an agent that has to spot defective parts on an assembly line bases each decision on the current part, regardless of previous decisions;
In **sequential environments**, on the other hand, the current decision could affect all future decisions. Chess and taxi driving are sequential:

- **Discrete** vs. **continuous.**

The discrete/continuous distinction can be applied to the *state* of the environment, to the way *time* is handled, and to the *percepts* and *actions* of the agent. For example, a discrete-state environment such as a chess game has a finite number of distinct states. Chess also has a discrete set of percepts and actions. Taxi driving is a continuous- state and continuous-time problem: the speed and location of the taxi and of the other vehicles sweep through a range of continuous values and do so smoothly over time. Taxi-driving actions are also continuous (steering angles, etc.).

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	Stochastic	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

Figure 1.7 Examples of task environments and their characteristics.

Examples of Agent Types and their Descriptions

Agent Type	Percepts	Actions	Goals	Environment
Medical diagnosis system	Symptoms, findings, patient's answers	Questions, tests, treatments	Healthy patient, minimize costs	Patient, hospital
Satellite image analysis system	Pixels of varying intensity, color	Print a categorization of scene	Correct categorization	Images from orbiting satellite
Part-picking robot	Pixels of varying intensity	Pick up parts and sort into bins	Place parts in correct bins	Conveyor belt with parts
Refinery controller	Temperature, pressure readings	Open, close valves; adjust temperature	Maximize purity, yield, safety	Refinery
Interactive English tutor	Typed words	Print exercises, suggestions, corrections	Maximize student's score on test	Set of students

Examples of agent types and their PEAS descriptions.

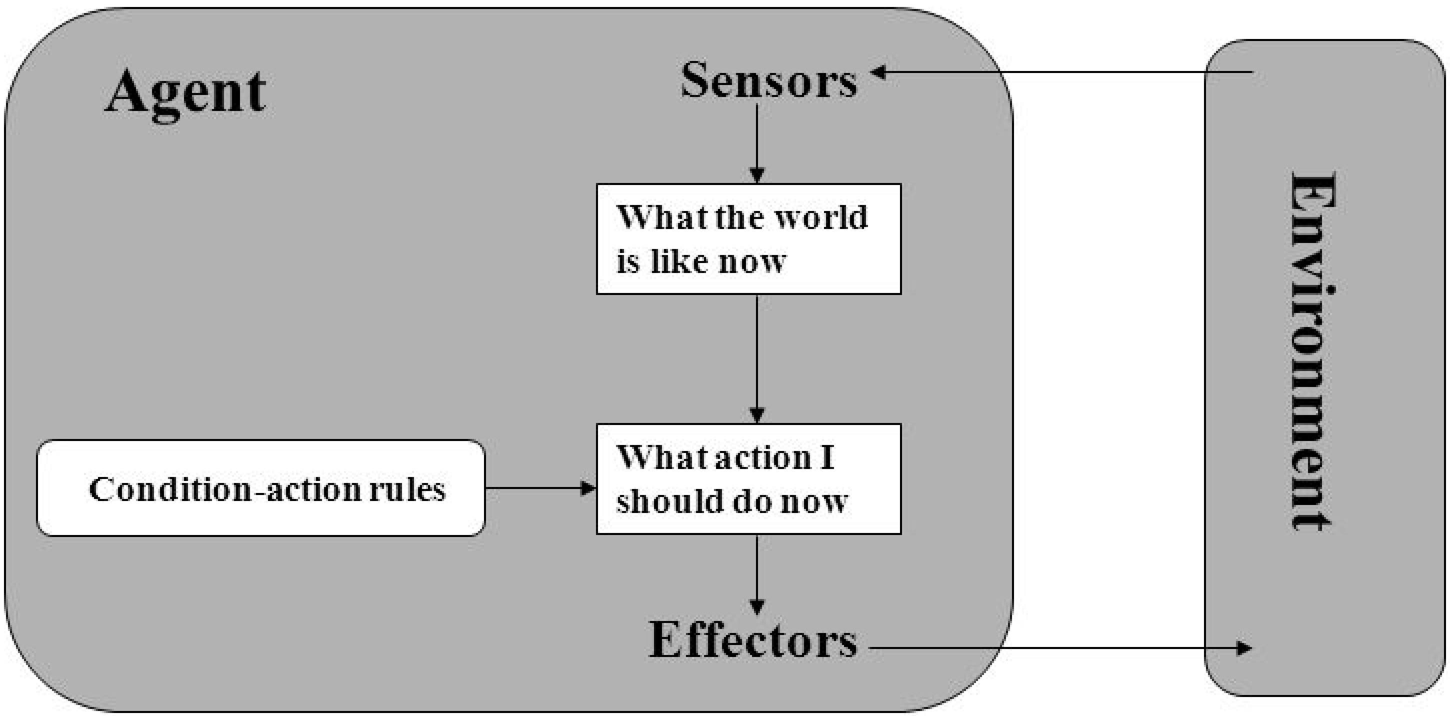
Some Agent Types

- **Table-driven agents**
  - use a percept sequence/ action table in memory to find the next action. They are implemented by a (large) lookup table.
- **Simple reflex agents**
  - are based on condition-action rules and implemented with an appropriate production (rule-based) system. They are stateless devices which do not have memory of past world states.
- **Agents with memory**
  - have internal state which is used to keep track of past states of the world.
- **Agents with goals**
  - are agents which in addition to state information have a kind of goal information which describes desirable situations. Agents of this kind take future events into consideration.
- **Utility-based agents**
  - base their decision on classic axiomatic utility-theory in order to act rationally.

Simple Reflex Agent

- **Table lookup** of percept- action pairs defining all possible condition- action rules necessary to interact in an environment
  - **Problems**
  - Too big to generate and to store (Chess has about 10^120 states, for example)
  - Not adaptive to changes in the environment; requires entire table to be updated if changes occur
- **Use *condition-action* rules to summarize portions of the table**
- *If Car-in-front –is –braking* ***then*** *initiate-braking*

A Simple Reflex Agent: Schema

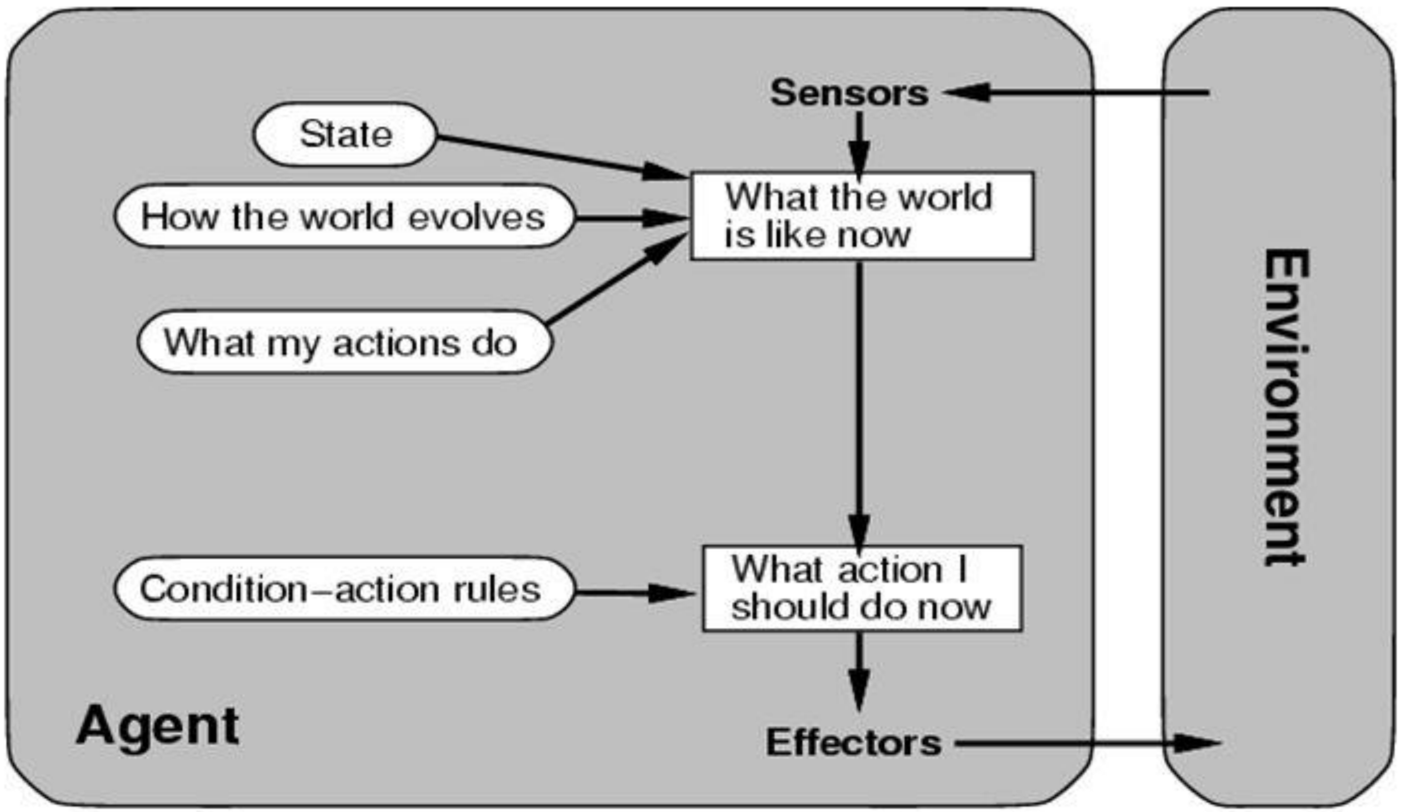


Reflex Agent with Internal State

- Encode "internal state" of the world to remember the past as contained in earlier percepts
- Needed because sensors do not usually give the entire state of the world at each input, so perception of the environment is captured over time. "State" used to encode different "world states" that generate the same immediate percept.
- Requires ability to represent change in the world; one possibility is to represent just the latest state, but then can't reason about hypothetical courses of action

A model based reflex agent

Agents that Keep Track of the World

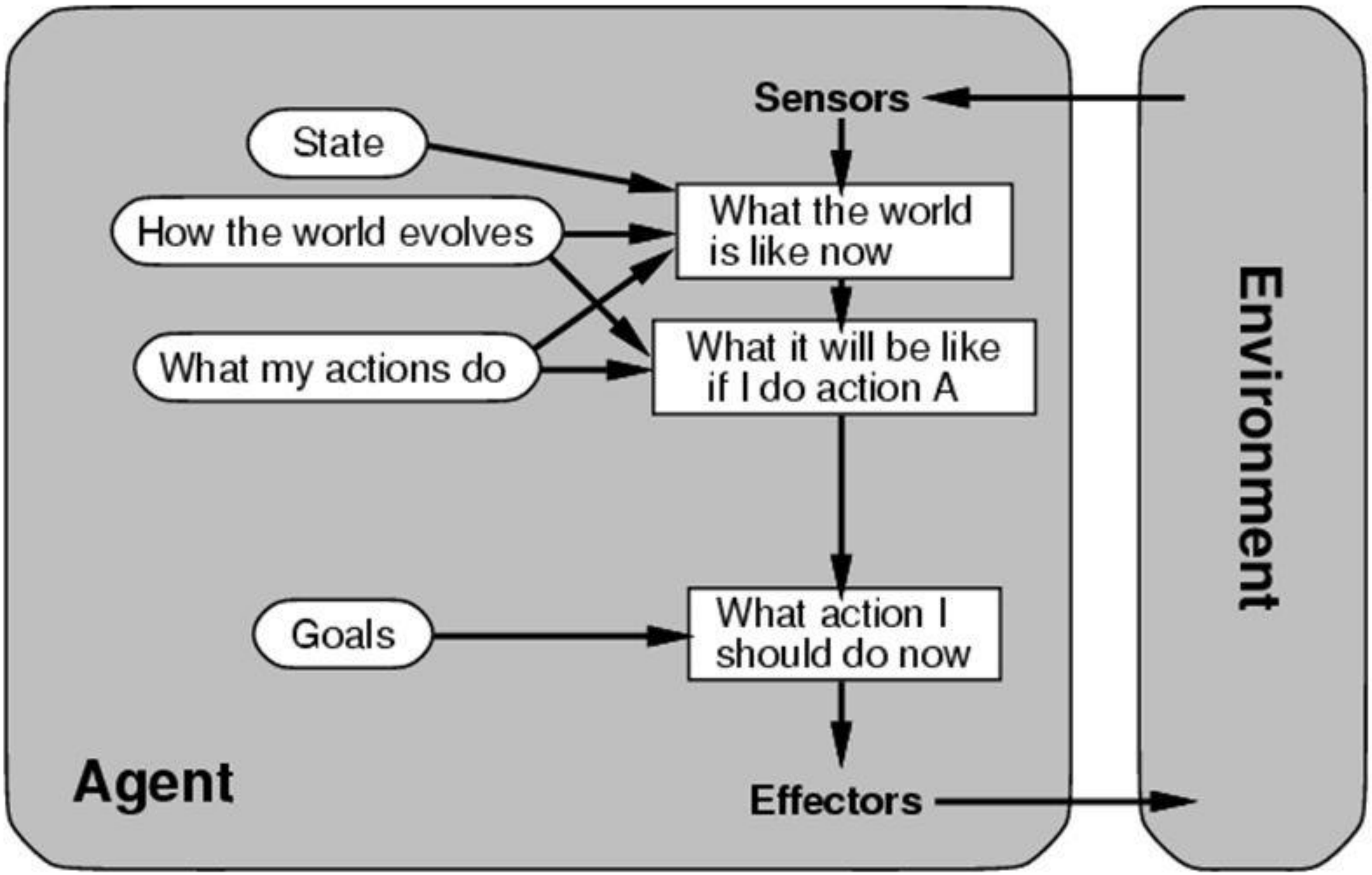


A model based reflex agent

Goal- Based Agent

- Choose actions so as to achieve a (given or computed) goal.
- A goal is a description of a desirable situation
- Keeping track of the current state is often not enough -- need to add goals to decide which situations are good
- May have to consider long sequences of possible actions before deciding if goal is achieved -- involves consideration of the future, *"what will happen if I do...?"*

Agents with Explicit Goals

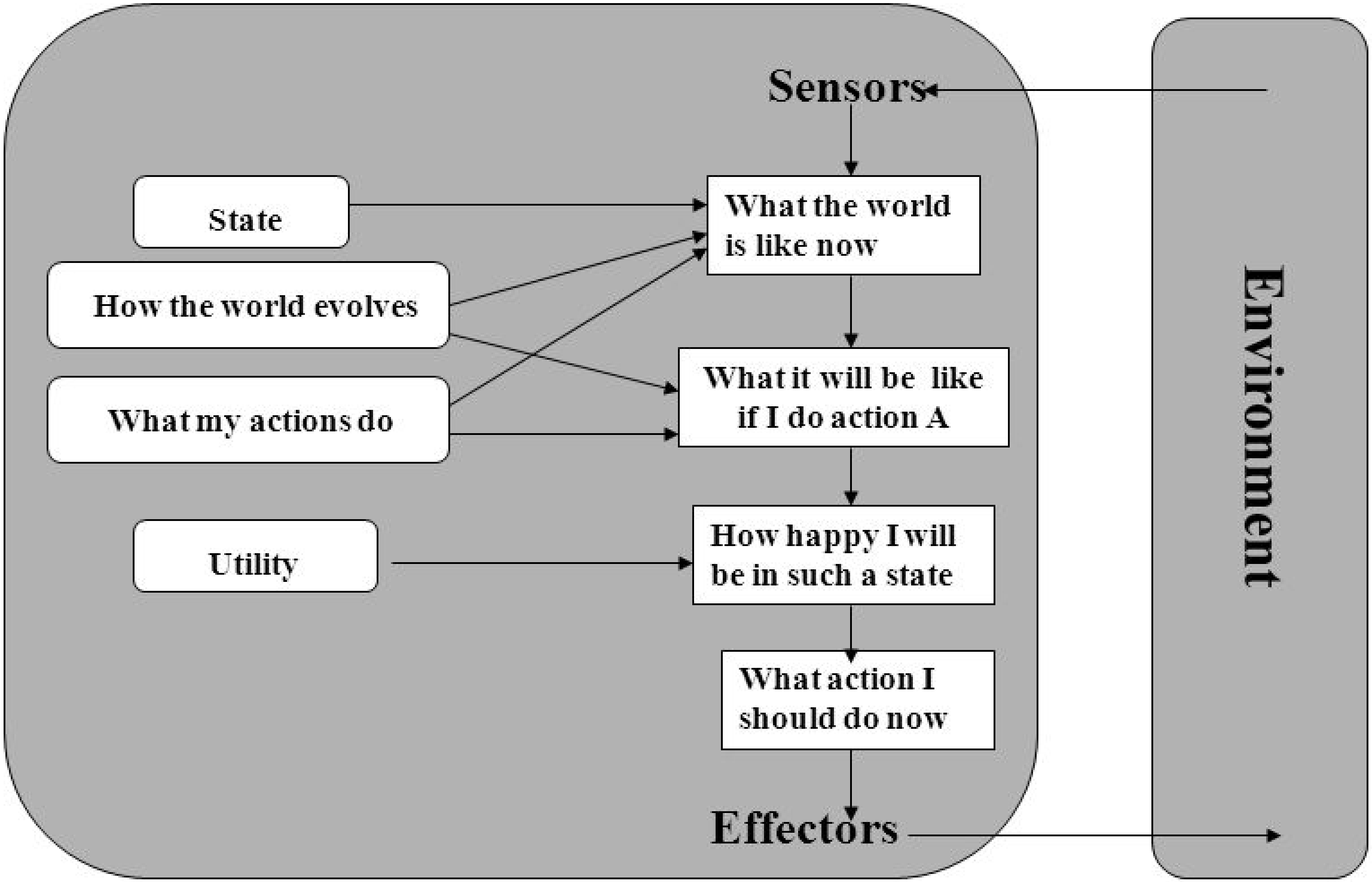


A goal based agent

Utility- Based Agent

- When there are multiple possible alternatives, how to decide which one is best?
- A goal specifies a crude distinction between a happy and unhappy state, but often need a more general performance measure that describes "degree of happiness"
- Utility function **U: States --> Reals** indicating a measure of success or happiness when at a given state
- Allows decisions comparing choice between conflicting goals, and choice between likelihood of success and importance of goal (if achievement is uncertain)

A Complete Utility- Based Agent



Utility based agent

Properties of Environments

- **Accessible/ Inaccessible.**
  - If an agent's sensors give it access to the complete state of the environment needed to choose an action, the environment is accessible.
  - Such environments are convenient, since the agent is freed from the task of keeping track of the changes in the environment.
- **Deterministic/ Non-deterministi**c.
  - An environment is deterministic if the next state of the environment is completely determined by the current state of the environment and the action of the agent.
  - In an accessible and deterministic environment the agent need not deal with uncertainty.
- **Episodic/ Nonepisodi**c.
  - An episodic environment means that subsequent episodes do not depend on what actions occurred in previous episodes.
  - Such environments do not require the agent to plan ahead.

Properties of Environments

- **Static/ Dynamic.**
  - A static environment does not change while the agent is thinking.
  - In a static environment the agent need not worry about the passage of time while he is thinking, nor does he have to observe the world while he is thinking.
  - In static environments the time it takes to compute a good strategy does not matter.
- **Discrete/ Continuous.**
  - If the number of distinct percepts and actions is limited the environment is discrete, otherwise it is continuous.
-

Characteristics of environments

	Accessible	Deterministic	Episodic	Static	Discrete
Solitaire					
Backgammon					
Taxi driving					
Internet shopping					
Medical diagnosis					

Characteristics of environments

	Accessible	Deterministic	Episodic	Static	Discrete
Solitaire	**No**	**Yes**	**Yes**	**Yes**	**Yes**
Backgammon					
Taxi driving					
Internet shopping					
Medical diagnosis					

Characteristics of environments

	Accessible	Deterministic	Episodic	Static	Discrete
Solitaire	No	Yes	Yes	Yes	Yes
Backgammon	**Yes**	**No**	**No**	**Yes**	**Yes**
Taxi driving					
Internet shopping					
Medical diagnosis					

Characteristics of environments

	Accessible	Deterministic	Episodic	Static	Discrete
Solitaire	No	Yes	Yes	Yes	Yes
Backgammon	Yes	No	No	Yes	Yes
Taxi driving	**No**	**No**	**No**	**No**	**No**
Internet shopping					
Medical diagnosis					

Characteristics of environments

	Accessible	Deterministic	Episodic	Static	Discrete
Solitaire	No	Yes	Yes	Yes	Yes
Backgammon	Yes	No	No	Yes	Yes
Taxi driving	No	No	No	No	No
Internet shopping	**No**	**No**	**No**	**No**	**No**
Medical diagnosis					

Characteristics of environments

	Accessible	Deterministic	Episodic	Static	Discrete
Solitaire	No	Yes	Yes	Yes	Yes
Backgammon	Yes	No	No	Yes	Yes
Taxi driving	No	No	No	No	No
Internet shopping	No	No	No	No	No
Medical diagnosis	**No**	**No**	**No**	**No**	**No**

→ Lots of real-world domains fall into the hardest case!

Summary

- An **agent** perceives and acts in an environment, has an architecture and is implemented by an agent program.
- An **ideal agent** always chooses the action which maximizes its expected performance, given percept sequence received so far.
- An **autonomous agent** uses its own experience rather than built- in knowledge of the environment by the designer.
- An **agent program** maps from percept to action & updates its internal state.
  - **Reflex agents** respond immediately to percpets.
  - **Goal-based agents** act in order to achieve their goal(s).
  - **Utility-based** agents maximize their own utility function.
- **Representing knowledge** is important for successful agent design.
- Some **environments** are more difficult for agents than others. The most challenging environments are inaccessible, non-deterministic, non-episodic, dynamic, and continuous.