1 Introduction

1.1 What Is Artificial Intelligence?

The term *artificial intelligence* stirs emotions. For one thing there is our fascination with *intelligence*, which seemingly imparts to us humans a special place among life forms. Questions arise such as "*What is intelligence*?", "*How can one measure intelligence*?" or "*How does the brain work*?". All these questions are meaningful when trying to understand artificial intelligence. However, the central question for the engineer, especially for the computer scientist, is the question of the intelligent machine that behaves like a person, showing intelligent behavior.

The attribute *artificial* might awaken much different associations. It brings up fears of intelligent cyborgs. It recalls images from science fiction novels. It raises the question of whether our highest good, the soul, is something we should try to understand, model, or even reconstruct.

With such different offhand interpretations, it becomes difficult to define the term *artificial intelligence* or *AI* simply and robustly. Nevertheless I would like to try, using examples and historical definitions, to characterize the field of AI. In 1955, John McCarthy, one of the pioneers of AI, was the first to define the term *artificial intelligence*, roughly as follows:

> The goal of AI is to develop machines that behave as though they were intelligent.

To test this definition, the reader might imagine the following scenario. Fifteen or so small robotic vehicles are moving on an enclosed four by four meter square surface. One can observe various behavior patterns. Some vehicles form small groups with relatively little movement. Others move peacefully through the space and gracefully avoid any collision. Still others appear to follow a leader. Aggressive behaviors are also observable. Is what we are seeing intelligent behavior?

According to McCarthy's definition the aforementioned robots can be described as intelligent. The psychologist Valentin Braitenberg has shown that this seemingly complex behavior can be produced by very simple electrical circuits [Bra84]. So-called Braitenberg vehicles have two wheels, each of which is driven by an independent electric motor. The speed of each motor is influenced by a light sensor on

W. Ertel, *Introduction to Artificial Intelligence*,
Undergraduate Topics in Computer Science,
DOI 10.1007/978-0-85729-299-5_1, © Springer-Verlag London Limited 2011

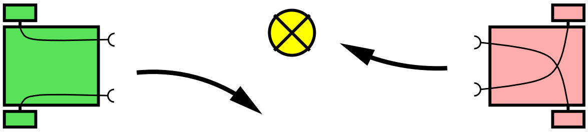


Fig. 1.1 Two very simple Braitenberg vehicles and their reactions to a light source

the front of the vehicle as shown in Fig. 1.1. The more light that hits the sensor, the faster the motor runs. Vehicle 1 in the left part of the figure, according to its configuration, moves away from a point light source. Vehicle 2 on the other hand moves toward the light source. Further small modifications can create other behavior patterns, such that with these very simple vehicles we can realize the impressive behavior described above.

Clearly the above definition is insufficient because AI has the goal of solving difficult practical problems which are surely too demanding for the Braitenberg vehicle. In the Encyclopedia Britannica [Bri91] one finds a Definition that goes like:

> AI is the ability of digital computers or computer controlled robots to solve problems that are normally associated with the higher intellectual processing capabilities of humans ...

But this definition also has weaknesses. It would admit for example that a computer with large memory that can save a long text and retrieve it on demand displays intelligent capabilities, for memorization of long texts can certainly be considered a *higher intellectual processing capability* of humans, as can for example the quick multiplication of two 20-digit numbers. According to this definition, then, every computer is an AI system. This dilemma is solved elegantly by the following definition by Elaine Rich [Ric83]:

> Artificial Intelligence is the study of how to make computers do things at which, at the moment, people are better.

Rich, tersely and concisely, characterizes what AI researchers have been doing for the last 50 years. Even in the year 2050, this definition will be up to date.

Tasks such as the execution of many computations in a short amount of time are the strong points of digital computers. In this regard they outperform humans by many multiples. In many other areas, however, humans are far superior to machines. For instance, a person entering an unfamiliar room will recognize the surroundings within fractions of a second and, if necessary, just as swiftly make decisions and plan actions. To date, this task is too demanding for autonomous[1] robots. According to Rich's definition, this is therefore a task for AI. In fact, research on autonomous robots is an important, current theme in AI. Construction of chess computers, on the other hand, has lost relevance because they already play at or above the level of grandmasters.

It would be dangerous, however, to conclude from Rich's definition that AI is only concerned with the pragmatic implementation of intelligent processes. Intelligent systems, in the sense of Rich's definition, cannot be built without a deep un-

[1] An autonomous robot works independently, without manual support, in particular without remote control.

derstanding of human reasoning and intelligent action in general, because of which neuroscience (see Sect. 1.1.1) is of great importance to AI. This also shows that the other cited definitions reflect important aspects of AI.

A particular strength of human intelligence is adaptivity. We are capable of adjusting to various environmental conditions and change our behavior accordingly through *learning*. Precisely because our learning ability is so vastly superior to that of computers, *machine learning* is, according to Rich's definition, a central subfield of AI.

1.1.1 Brain Science and Problem Solving

Through research of intelligent systems we can try to understand how the human brain works and then model or simulate it on the computer. Many ideas and principles in the field of neural networks (see Chap. 9) stem from brain science with the related field of neuroscience.

A very different approach results from taking a goal-oriented line of action, starting from a problem and trying to find the most optimal solution. How humans solve the problem is treated as unimportant here. The method, in this approach, is secondary. First and foremost is the optimal intelligent solution to the problem. Rather than employing a fixed method (such as, for example, predicate logic) AI has as its constant goal the creation of intelligent agents for as many different tasks as possible. Because the tasks may be very different, it is unsurprising that the methods currently employed in AI are often also quite different. Similar to medicine, which encompasses many different, often life-saving diagnostic and therapy procedures, AI also offers a broad palette of effective solutions for widely varying applications. For mental inspiration, consider Fig. 1.2 on page 4. Just as in medicine, there is no universal method for all application areas of AI, rather a great number of possible solutions for the great number of various everyday problems, big and small.

Cognitive science is devoted to research into human thinking at a somewhat higher level. Similarly to brain science, this field furnishes practical AI with many important ideas. On the other hand, algorithms and implementations lead to further important conclusions about how human reasoning functions. Thus these three fields benefit from a fruitful interdisciplinary exchange. The subject of this book, however, is primarily problem-oriented AI as a subdiscipline of computer science.

There are many interesting philosophical questions surrounding intelligence and artificial intelligence. We humans have consciousness; that is, we can think about ourselves and even ponder that we are able to think about ourselves. How does consciousness come to be? Many philosophers and neurologists now believe that the mind and consciousness are linked with matter, that is, with the brain. The question of whether machines could one day have a mind or consciousness could at some point in the future become relevant. The mind-body problem in particular concerns whether or not the mind is bound to the body. We will not discuss these questions here. The interested reader may consult [Spe98, Spe97] and is invited, in the course of AI technology studies, to form a personal opinion about these questions.

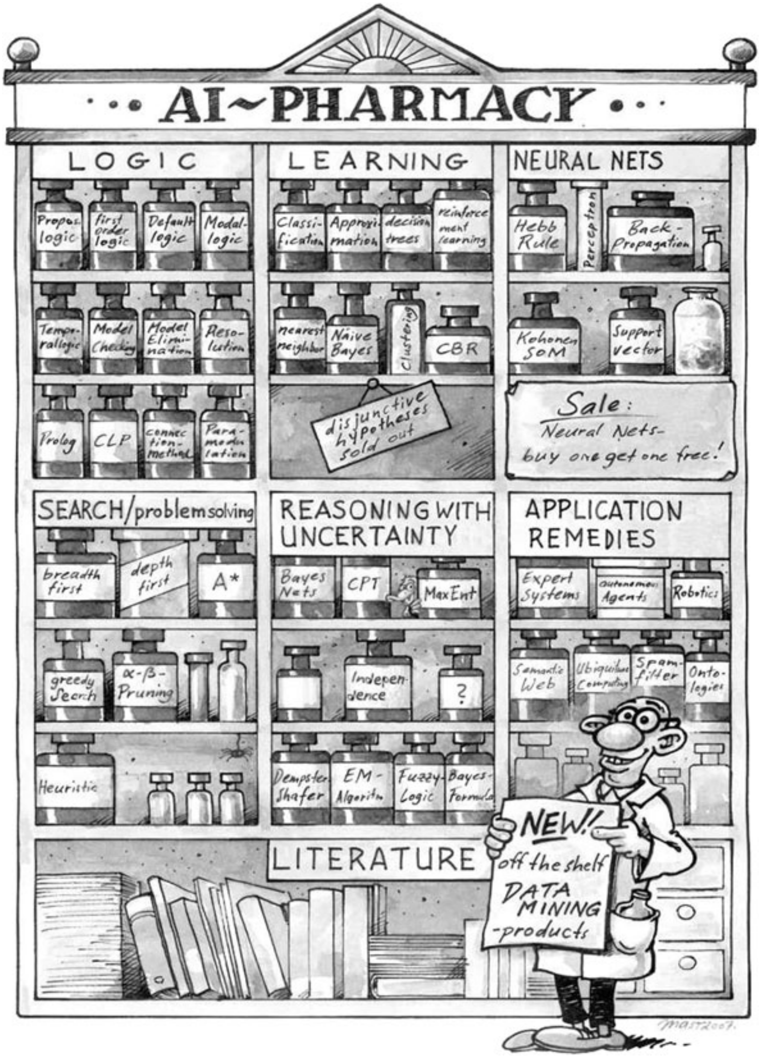


Fig. 1.2 A small sample of the solutions offered by AI

1.1.2 The Turing Test and Chatterbots

Alan Turing made a name for himself as an early pioneer of AI with his definition of an intelligent machine, in which the machine in question must pass the following test. The test person Alice sits in a locked room with two computer terminals. One terminal is connected to a machine, the other with a non-malicious person Bob. Alice can type questions into both terminals. She is given the task of deciding, after five minutes, which terminal belongs to the machine. The machine passes the test if it can trick Alice at least 30% of the time [Tur50].

While the test is very interesting philosophically, for practical AI, which deals with problem solving, it is not a very relevant test. The reasons for this are similar to those mentioned above related to Braitenberg vehicles (see Exercise 1.3 on page 14).

The AI pioneer and social critic Joseph Weizenbaum developed a program named *Eliza*, which is meant to answer a test subject's questions like a human psychologist [Wei66]. He was in fact able to demonstrate success in many cases. Supposedly his secretary often had long discussions with the program. Today in the internet there are many so-called *chatterbots*, some of whose initial responses are quite impressive. After a certain amount of time, however, their artificial nature becomes apparent. Some of these programs are actually capable of learning, while others possess extraordinary knowledge of various subjects, for example geography or software development. There are already commercial applications for chatterbots in online customer support and there may be others in the field of e-learning. It is conceivable that the learner and the e-learning system could communicate through a chatterbot. The reader may wish to compare several chatterbots and evaluate their intelligence in Exercise 1.1 on page 14.

1.2 The History of AI

AI draws upon many past scientific achievements which are not mentioned here, for AI as a science in its own right has only existed since the middle of the Twentieth Century. Table 1.1 on page 10, with the most important AI milestones, and a graphical representation of the main movements of AI in Fig. 1.3 on page 6 complement the following text.

1.2.1 The First Beginnings

In the 1930s Kurt Gödel, Alonso Church, and Alan Turing laid important foundations for logic and theoretical computer science. Of particular interest for AI are Gödel's theorems. The completeness theorem states that first-order predicate logic is complete. This means that every true statement that can be formulated in predicate logic is provable using the rules of a formal calculus. On this basis, automatic theorem provers could later be constructed as implementations of formal calculi. With the incompleteness theorem, Gödel showed that in higher-order logics there exist true statements that are unprovable.[2] With this he uncovered painful limits of formal systems.

Alan Turing's proof of the undecidability of the halting problem also falls into this time period. He showed that there is no program that can decide whether a given arbitrary program (and its respective input) will run in an infinite loop. With

[2] Higher-order logics are extensions of predicate logic, in which not only variables, but also function symbols or predicates can appear as terms in a quantification. Indeed, Gödel only showed that any system that is based on predicate logic and can formulate Peano arithmetic is incomplete.

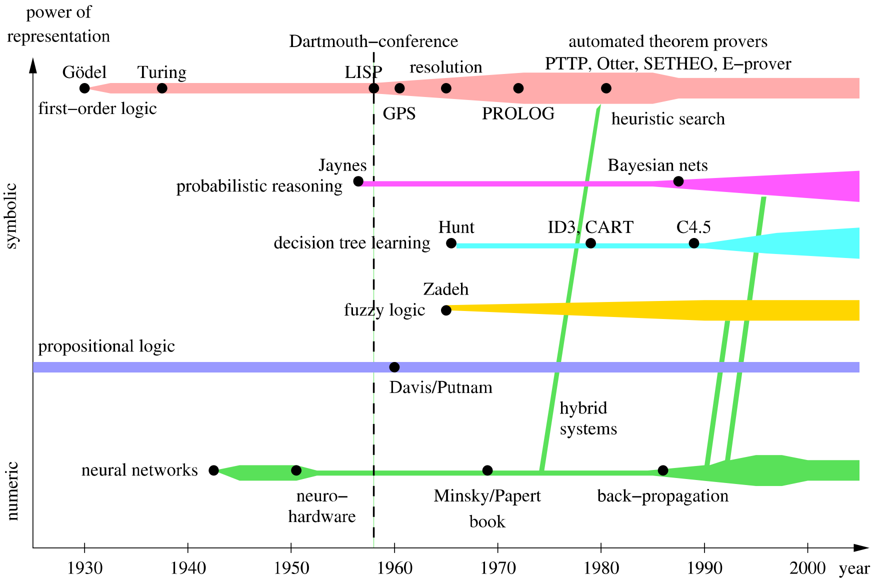


Fig. 1.3 History of the various AI areas. The width of the *bars* indicates prevalence of the method's use

this Turing also identified a limit for intelligent programs. It follows, for example, that there will never be a universal program verification system.[3]

In the 1940s, based on results from neuroscience, McCulloch, Pitts and Hebb designed the first mathematical models of neural networks. However, computers at that time lacked sufficient power to simulate simple brains.

1.2.2 Logic Solves (Almost) All Problems

AI as a practical science of thought mechanization could of course only begin once there were programmable computers. This was the case in the 1950s. Newell and Simon introduced Logic Theorist, the first automatic theorem prover, and thus also showed that with computers, which actually only work with numbers, one can also process symbols. At the same time McCarthy introduced, with the language LISP, a programming language specially created for the processing of symbolic structures. Both of these systems were introduced in 1956 at the historic Dartmouth Conference, which is considered the birthday of AI.

In the US, LISP developed into the most important tool for the implementation of symbol-processing AI systems. Thereafter the logical inference rule known as resolution developed into a complete calculus for predicate logic.

[3]This statement applies to "total correctness", which implies a proof of correct execution as well as a proof of termination for every valid input.

In the 1970s the logic programming language PROLOG was introduced as the European counterpart to LISP. PROLOG offers the advantage of allowing direct programming using Horn clauses, a subset of predicate logic. Like LISP, PROLOG has data types for convenient processing of lists.

Until well into the 1980s, a breakthrough spirit dominated AI, especially among many logicians. The reason for this was the string of impressive achievements in symbol processing. With the Fifth Generation Computer Systems project in Japan and the ESPRIT program in Europe, heavy investment went into the construction of intelligent computers.

For small problems, automatic provers and other symbol-processing systems sometimes worked very well. The combinatorial explosion of the search space, however, defined a very narrow window for these successes. This phase of AI was described in [RN10] as the "Look, Ma, no hands!" era.

Because the economic success of AI systems fell short of expectations, funding for logic-based AI research in the United States fell dramatically during the 1980s.

1.2.3 The New Connectionism

During this phase of disillusionment, computer scientists, physicists, and Cognitive scientists were able to show, using computers which were now sufficiently powerful, that mathematically modeled neural networks are capable of learning using training examples, to perform tasks which previously required costly programming. Because of the fault-tolerance of such systems and their ability to recognize patterns, considerable successes became possible, especially in pattern recognition. Facial recognition in photos and handwriting recognition are two example applications. The system Nettalk was able to learn speech from example texts [SR86]. Under the name *connectionism*, a new subdiscipline of AI was born.

Connectionism boomed and the subsidies flowed. But soon even here feasibility limits became obvious. The neural networks could acquire impressive capabilities, but it was usually not possible to capture the learned concept in simple formulas or logical rules. Attempts to combine neural nets with logical rules or the knowledge of human experts met with great difficulties. Additionally, no satisfactory solution to the structuring and modularization of the networks was found.

1.2.4 Reasoning Under Uncertainty

AI as a practical, goal-driven science searched for a way out of this crisis. One wished to unite logic's ability to explicitly represent knowledge with neural networks' strength in handling uncertainty. Several alternatives were suggested.

The most promising, *probabilistic reasoning*, works with conditional probabilities for propositional calculus formulas. Since then many diagnostic and expert systems have been built for problems of everyday reasoning using *Bayesian networks*. The success of Bayesian networks stems from their intuitive comprehensibility, the clean semantics of conditional probability, and from the centuries-old, mathematically grounded probability theory.

The weaknesses of logic, which can only work with two truth values, can be solved by *fuzzy logic*, which pragmatically introduces infinitely many values between zero and one. Though even today its theoretical foundation is not totally firm, it is being successfully utilized, especially in control engineering.

A much different path led to the successful synthesis of logic and neural networks under the name *hybrid systems*. For example, neural networks were employed to learn heuristics for reduction of the huge combinatorial search space in proof discovery [SE90].

Methods of decision tree learning from data also work with probabilities. Systems like CART, ID3 and C4.5 can quickly and automatically build very accurate decision trees which can represent propositional logic concepts and then be used as expert systems. Today they are a favorite among machine learning techniques (Sect. 8.4).

Since about 1990, *data mining* has developed as a subdiscipline of AI in the area of statistical data analysis for extraction of knowledge from large databases. Data mining brings no new techniques to AI, rather it introduces the requirement of using large databases to gain explicit knowledge. One application with great market potential is steering ad campaigns of big businesses based on analysis of many millions of purchases by their customers. Typically, machine learning techniques such as decision tree learning come into play here.

1.2.5 Distributed, Autonomous and Learning Agents

Distributed artificial intelligence, DAI, has been an active area research since about 1985. One of its goals is the use of parallel computers to increase the efficiency of problem solvers. It turned out, however, that because of the high computational complexity of most problems, the use of "intelligent" systems is more beneficial than parallelization itself.

A very different conceptual approach results from the development of autonomous software agents and robots that are meant to cooperate like human teams. As with the aforementioned Braitenberg vehicles, there are many cases in which an individual agent is not capable of solving a problem, even with unlimited resources. Only the cooperation of many agents leads to the intelligent behavior or to the solution of a problem. An ant colony or a termite colony is capable of erecting buildings of very high architectural complexity, despite the fact that no single ant comprehends how the whole thing fits together. This is similar to the situation of provisioning bread for a large city like New York [RN10]. There is no central planning agency for bread, rather there are hundreds of bakers that know their respective areas of the city and bake the appropriate amount of bread at those locations.

Active skill acquisition by robots is an exciting area of current research. There are robots today, for example, that independently learn to walk or to perform various motorskills related to soccer (Chap. 10). Cooperative learning of multiple robots to solve problems together is still in its infancy.

1.2.6 AI Grows up

The above systems offered by AI today are not a universal recipe, but a workshop with a manageable number of tools for very different tasks. Most of these tools are well-developed and are available as finished software libraries, often with convenient user interfaces. The selection of the right tool and its sensible use in each individual case is left to the AI developer or knowledge engineer. Like any other artisanship, this requires a solid education, which this book is meant to promote.

More than nearly any other science, AI is interdisciplinary, for it draws upon interesting discoveries from such diverse fields as logic, operations research, statistics, control engineering, image processing, linguistics, philosophy, psychology, and neurobiology. On top of that, there is the subject area of the particular application. To successfully develop an AI project is therefore not always so simple, but almost always extremely exciting.

1.3 Agents

Although the term *intelligent agents* is not new to AI, only in recent years has it gained prominence through [RN10], among others. *Agent* denotes rather generally a system that processes information and produces an output from an input. These agents may be classified in many different ways.

In classical computer science, *software agents* are primarily employed (Fig. 1.4 on page 11). In this case the agent consists of a program that calculates a result from user input.

In robotics, on the other hand, *hardware agents* (also called robots) are employed, which additionally have sensors and actuators at their disposal (Fig. 1.5 on page 11). The agent can perceive its environment with the sensors. With the actuators it carries out actions and changes its environment.

With respect to the intelligence of the agent, there is a distinction between *reflex agents*, which only react to input, and *agents with memory*, which can also include the past in their decisions. For example, a driving robot that through its sensors knows its exact position (and the time) has no way, as a reflex agent, of determining its velocity. If, however, it saves the position, at short, discrete time steps, it can thus easily calculate its average velocity in the previous time interval.

If a reflex agent is controlled by a deterministic program, it represents a function of the set of all inputs to the set of all outputs. An agent with memory, on the other hand, is in general not a function. Why? (See Exercise 1.5 on page 14.) Reflex agents are sufficient in cases where the problem to be solved involves a Markov decision process. This is a process in which only the current state is needed to determine the optimal next action (see Chap. 10).

A mobile robot which should move from room 112 to room 179 in a building takes actions different from those of a robot that should move to room 105. In other words, the actions depend on the goal. Such agents are called *goal-based*.

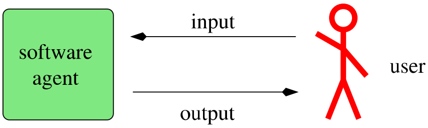
Table 1.1 Milestones in the development of AI from Gödel to today

1931	The Austrian Kurt Gödel shows that in first-order *predicate logic* all true statements are derivable [Göd31a]. In higher-order logics, on the other hand, there are true statements that are unprovable [Göd31b]. (In [Göd31b] Gödel showed that predicate logic extended with the axioms of arithmetic is incomplete.)
1937	Alan Turing points out the limits of intelligent machines with the halting problem [Tur37].
1943	McCulloch and Pitts model *neural networks* and make the connection to propositional logic.
1950	Alan Turing defines machine intelligence with the *Turing test* and writes about learning machines and genetic algorithms [Tur50].
1951	Marvin Minsky develops a neural network machine. With 3000 vacuum tubes he simulates 40 neurons.
1955	Arthur Samuel (IBM) builds a learning chess program that plays better than its developer [Sam59].
1956	McCarthy organizes a conference in Dartmouth College. Here the name *Artificial Intelligence* was first introduced.
	Newell and Simon of Carnegie Mellon University (CMU) present the *Logic Theorist*, the first symbol-processing computer program [NSS83].
1958	McCarthy invents at MIT (Massachusetts Institute of Technology) the high-level language *LISP*. He writes programs that are capable of modifying themselves.
1959	Gelernter (IBM) builds the Geometry Theorem Prover.
1961	The General Problem Solver (GPS) by Newell and Simon imitates human thought [NS61].
1963	McCarthy founds the AI Lab at Stanford University.
1965	Robinson invents the *resolution calculus* for predicate logic [Rob65] (Sect. 3.5).
1966	Weizenbaum's program Eliza carries out dialog with people in natural language [Wei66] (Sect. 1.1.2).
1969	Minsky and Papert show in their book *Perceptrons* that the perceptron, a very simple neural network, can only represent linear functions [MP69] (Sect. 1.1.2).
1972	French scientist Alain Colmerauer invents the logic programming language *PROLOG* (Chap. 5).
	British physician de Dombal develops an *expert system* for diagnosis of acute abdominal pain [dDLS$^+$72]. It goes unnoticed in the mainstream AI community of the time (Sect. 7.3).
1976	Shortliffe and Buchanan develop MYCIN, an expert system for diagnosis of infectious diseases, which is capable of dealing with uncertainty (Chap. 7).
1981	Japan begins, at great expense, the "Fifth Generation Project" with the goal of building a powerful PROLOG machine.
1982	R1, the expert system for configuring computers, saves Digital Equipment Corporation 40 million dollars per year [McD82].
1986	Renaissance of neural networks through, among others, Rumelhart, Hinton and Sejnowski [RM86]. The system Nettalk learns to read texts aloud [SR86] (Chap. 9).
1990	Pearl [Pea88], Cheeseman [Che85], Whittaker, Spiegelhalter bring probability theory into AI with *Bayesian networks* (Sect. 7.4). Multi-agent systems become popular.
1992	Tesauros TD-gammon program demonstrates the advantages of reinforcement learning.
1993	Worldwide *RoboCup* initiative to build soccer-playing autonomous robots [Roba].

Table 1.1 (continued)

1995	From statistical learning theory, Vapnik develops support vector machines, which are very important today.
1997	IBM's chess computer Deep Blue defeats the chess world champion Gary Kasparov.
	First international RoboCup competition in Japan.
2003	The robots in RoboCup demonstrate impressively what AI and robotics are capable of achieving.
2006	Service robotics becomes a major AI research area.
2010	Autonomous robots start learning their policies.
2011	IBM's natural language understanding and question answering program "Watson" defeats two human champions in the U.S. television quiz show "Jeopardy!" (Sect. 1.4).

Fig. 1.4 A software agent with user interaction



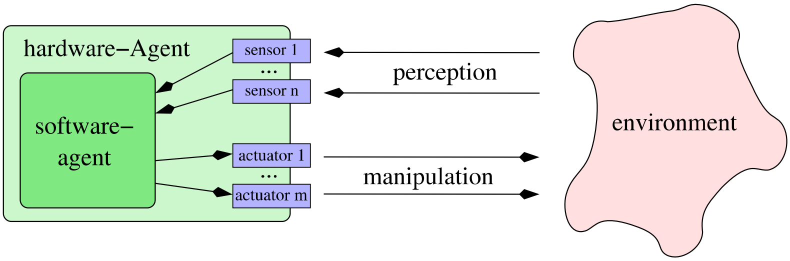


Fig. 1.5 A hardware agent

Example 1.1 A spam filter is an agent that puts incoming emails into wanted or unwanted (spam) categories, and deletes any unwanted emails. Its goal as a goal-based agent is to put all emails in the right category. In the course of this not-so-simple task, the agent can occasionally make mistakes. Because its goal is to classify all emails correctly, it will attempt to make as few errors as possible. However, that is not always what the user has in mind. Let us compare the following two agents. Out of 1,000 emails, Agent 1 makes only 12 errors. Agent 2 on the other hand makes 38 errors with the same 1,000 emails. Is it therefore worse than Agent 1? The errors of both agents are shown in more detail in the following table, the so-called "confusion matrix":

Agent 1:

		correct class	
		wanted	spam
spam filter decides	wanted	189	1
	spam	11	799

Agent 2:

		correct class	
		wanted	spam
spam filter decides	wanted	200	38
	spam	0	762

Agent 1 in fact makes fewer errors than Agent 2, but those few errors are severe because the user loses 11 potentially important emails. Because there are in this case two types of errors of differing severity, each error should be weighted with the appropriate cost factor (see Sect. 7.3.5 and Exercise 1.7 on page 14).

The sum of all weighted errors gives the total cost caused by erroneous decisions. The goal of a *cost-based agent* is to minimize the cost of erroneous decisions in the long term, that is, on average. In Sect. 7.3 we will become familiar with the diagnostic system LEXMED as an example of a cost-based agent.

Analogously, the goal of a utility-based agent is to maximize the utility derived from correct decisions in the long term, that is, on average. The sum of all decisions weighted by their respective utility factors gives the total utility.

Of particular interest in AI are *Learning agents*, which are capable of changing themselves given training examples or through positive or negative feedback, such that the average utility of their actions grows over time (see Chap. 8).

As mentioned in Sect. 1.2.5, *distributed agents* are increasingly coming into use, whose intelligence are not localized in one agent, but rather can only be seen through cooperation of many agents.

The design of an agent is oriented, along with its objective, strongly toward its *environment*, or alternately its picture of the environment, which strongly depends on it sensors. The environment is *observable* if the agent always knows the complete state of the world. Otherwise the environment is only *partially observable*. If an action always leads to the same result, then the environment is *deterministic*. Otherwise it is *nondeterministic*. In a *discrete environment* only finitely many states and actions occur, whereas a *continuous environment* boasts infinitely many states or actions.

1.4 Knowledge-Based Systems

An agent is a program that implements a mapping from perceptions to actions. For simple agents this way of looking at the problem is sufficient. For complex applications in which the agent must be able to rely on a large amount of information and is meant to do a difficult task, programming the agent can be very costly and unclear how to proceed. Here AI provides a clear path to follow that will greatly simplify the work.

First we separate *knowledge* from the system or program, which uses the knowledge to, for example, reach conclusions, answer queries, or come up with a plan. This system is called the *inference mechanism*. The knowledge is stored in a *knowledge base* (*KB*). Acquisition of knowledge in the knowledge base is denoted *Knowledge Engineering* and is based on various knowledge sources such as human experts, the knowledge engineer, and databases. Active learning systems can also acquire knowledge though active exploration of the world (see Chap. 10). In Fig. 1.6 on page 13 the general architecture of knowledge-based systems is presented.

Moving toward a separation of knowledge and inference has several crucial advantages. The separation of knowledge and inference can allow inference systems to be implemented in a largely application-independent way. For example, it is much

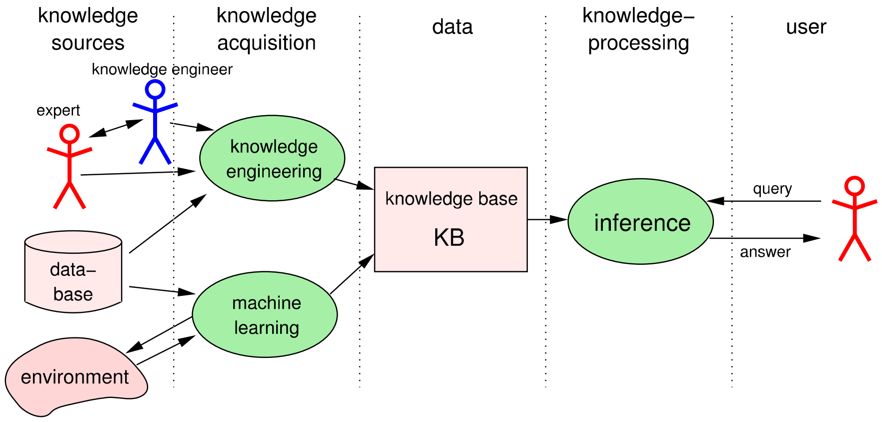


Fig. 1.6 Structure of a classic knowledge-processing system

easier to replace the knowledge base of a medical expert system than to program a whole new system.

Through the decoupling of the knowledge base from inference, knowledge can be stored declaratively. In the knowledge base there is only a description of the knowledge, which is independent from the inference system in use. Without this clear separation, knowledge and processing of inference steps would be interwoven, and any changes to the knowledge would be very costly.

Formal language as a convenient interface between man and machine lends itself to the representation of knowledge in the knowledge base. In the following chapters we will get to know a whole series of such languages. First, in Chaps. 2 and 3 there are *propositional calculus* and *first-order predicate logic* (PL1). But other formalisms such as probabilistic logic, fuzzy logic or decision trees are also presented. We start with propositional calculus and the related inference systems. Building on that, we will present predicate logic, a powerful language that is accessible by machines and very important in AI.

As an example for a large scale knowledge based system we want to refer to the software agent "Watson". Developed at IBM together with a number of universities, Watson is a question answering program, that can be fed with clues given in natural language. It works on a knowledge base comprising four terabytes of hard disk storage, including the full text of Wikipedia [FNA+09]. Watson was developed within IBM's DeepQA project which is characterized in [Dee11] as follows:

> The DeepQA project at IBM shapes a grand challenge in Computer Science that aims to illustrate how the wide and growing accessibility of natural language content and the integration and advancement of Natural Language Processing, Information Retrieval, Machine Learning, Knowledge Representation and Reasoning, and massively parallel computation can drive open-domain automatic Question Answering technology to a point where it clearly and consistently rivals the best human performance.

In the U.S. television quiz show "Jeopardy!", in February 2011, Watson defeated the two human champions Brad Rutter and Ken Jennings in a two-game, combined-point match and won the one million dollar price. One of Watson's particular strengths was its very fast reaction to the questions with the result that Watson often hit the buzzer (using a solenoid) faster than its human competitors and then was able to give the first answer to the question.

The high performance and short reaction times of Watson were due to an implementation on 90 IBM Power 750 servers, each of which contains 32 processors, resulting in 2880 parallel processors.

1.5 Exercises

Exercise 1.1 Test some of the chatterbots available on the internet. Start for example with www.hs-weingarten.de/~ertel/aibook in the collection of links under Turingtest/Chatterbots, or at www.simonlaven.com or www.alicebot.org. Write down a starting question and measure the time it takes, for each of the various programs, until you know for certain that it is not a human.

❄❄ **Exercise 1.2** At www.pandorabots.com you will find a server on which you can build a chatterbot with the markup language AIML quite easily. Depending on your interest level, develop a simple or complex chatterbot, or change an existing one.

Exercise 1.3 Give reasons for the unsuitability of the Turing test as a definition of "*artificial intelligence*" in practical AI.

➾ **Exercise 1.4** Many well-known inference processes, learning processes, etc. are NP-complete or even undecidable. What does this mean for AI?

Exercise 1.5

(a) Why is a deterministic agent with memory not a function from the set of all inputs to the set of all outputs, in the mathematical sense?

(b) How can one change the agent with memory, or model it, such that it becomes equivalent to a function but does not lose its memory?

Exercise 1.6 Let there be an agent with memory that can move within a plane. From its sensors, it receives at clock ticks of a regular interval Δt its exact position (x, y) in Cartesian coordinates.

(a) Give a formula with which the agent can calculate its velocity from the current time t and the previous measurement of $t - \Delta t$.

(b) How must the agent be changed so that it can also calculate its acceleration? Provide a formula here as well.

❄ **Exercise 1.7**

(a) Determine for both agents in Example 1.1 on page 11 the costs created by the errors and compare the results. Assume here that having to manually delete a spam email costs one cent and retrieving a deleted email, or the loss of an email, costs one dollar.

(b) Determine for both agents the profit created by correct classifications and compare the results. Assume that for every desired email recognized, a profit of one dollar accrues and for every correctly deleted spam email, a profit of one cent.