3 The Fundamental Problems of Classical Artificial Intelligence and Cognitive Science

So far we have looked at the nature of intelligence and discussed the cognitivistic paradigm, which still by far dominates scientific and everyday thinking about intelligence. In a number of places, however, we have alluded to potential problems with this paradigm. In this chapter, we inspect more closely what these problems really are and why they have arisen in the first place. As we argue, the cognitivistic paradigm's neglect of the fact that intelligent agents, humans, animals, and robots are embodied agents that live in a real physical world leads to significant shortcomings in explaining intelligence.

Outlining the cognitivistic paradigm's problems and understanding their origins helps us, on the one hand, to avoid making the same mistakes again; on the other hand, it provides us with inspiration about what needs to be done differently. The chapter is relatively short. Most of the issues it raises have been discussed at length in the literature (e.g., Brooks 1991a,b; Clancey 1997; Franklin 1995; Hendriks-Jansen 1996; Winograd and Flores 1986), and an overview of those issues is sufficient here without repeating the details of the arguments. The goal is to outline the main problems that historically have led researchers to reconsider their approach to the study of intelligence.

We proceed as follows in the chapter: first, we work out the main distinctive characteristics of real and virtual worlds. We then present an overview of some of the well-known problems of traditional systems, followed by an inspection of some of the fundamental issues involved. We conclude with a number of suggestions as to what might be to be done in order to overcome these problems.

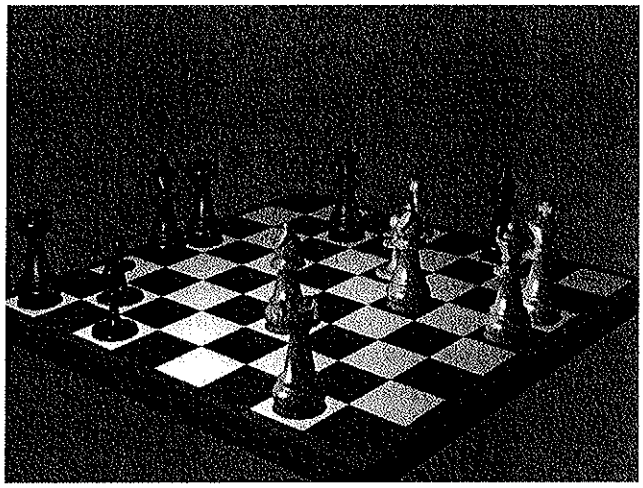
3.1 Real Worlds versus Virtual Worlds

Classical models, that is, models developed within the cognitivistic paradigm, focus on high-level processes like problem solving, reasoning, making inferences, and playing chess. Much progress has been made, as we have seen, for example, in the case of chess, with

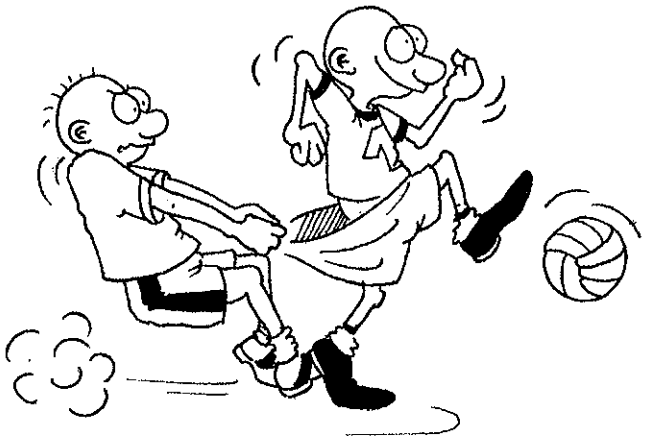
computers able to play well enough to defeat world champions. In other areas, progress has been less rapid; for example, in computer vision. It has turned out to be far more involved than expected to extract information from camera images, typically in the form of a pixel array, and map them onto internal representations of the world. The main reason for these difficulties—and the reason for the fundamental problems of AI in general—is that the models do not take the real world sufficiently into account. Much work in classical AI has been devoted to abstract, virtual worlds with precisely defined states and operations, quite unlike the real world.

To illustrate our argument, let us return to the game of chess (figure 3.1a). Chess is a formal game. It represents a virtual world with discrete, clearly defined states, board positions, operations, and legal moves. It is also a game involving complete information: If you know the board position, it is possible to know all you need to know to play the game, because given a certain board position, the possible moves are precisely defined and finite in number. Even though you may not know the particular move your opponent will make, you know that he will make a legal move; if he did not, he would cease to be playing chess any longer. (Breaking the chess board over the opponent's head is not part of the game itself.) Chess is also a static game, in the sense that if no one makes a move, nothing changes. Moreover, the types of possible moves do not change over time.

By contrast, consider soccer (figure 3.1b). Soccer is clearly a nonformal game. It takes place in the real world, where there are no uniquely defined states. The world of soccer—the real world—is continuous. As humans, we can make a model of a soccer game, and that model may have states, but not the soccer game as such. Having no uniquely defined states also implies that two situations in the real world are never identical. Moreover, in contrast to virtual worlds, the available information an agent can acquire about the real world is always incomplete. A soccer player cannot know about the activities of all other players at the same time, and those activities are drawn from a nearly infinite range of possibilities. In fact, it is not even defined what "complete" information means where a game like soccer is concerned. Completeness can be defined only within a closed, formal world. Since completeness is not defined, it is better to talk in terms of limited information. A soccer player has only limited information about the overall situation. In fact, information that can be acquired about the real world



a



b

Figure 3.1 Real worlds and virtual words. (a) Chess is a formal game. It represents a virtual world with precisely defined states, board positions, and operations, that is, the legal moves. (b) Soccer is an example of a nonformal game. There are no precisely defined states and operations. In contrast to chess, two situations in soccer are never exactly identical.

is always limited because of embodiment: the field of view is restricted, the range of the sensors is limited, and the sensory and motor systems take time to operate. Moreover, in the real world there is time pressure: things happen even if we do not do anything, and they happen in real time. If we want to avoid getting hit by cars, we may have to run away quickly. If we are jumping off a wall, the laws of physics act on the body (gravity), and we have to react quickly in order not to get hurt. Other laws of physics are also relevant: Friction is required for locomotion, motion requires

energy, and physical organisms all have a certain metabolism that also needs energy. These are physical phenomena. They do not have to be represented somehow in order to function. They are simply there.

In the real world, any physical device is subject to noise, disturbances, and malfunctions. This point holds in principle for any sensory or motor system. In other words, information gathered from the sensors is therefore always subject to errors. Finally, the real world is indefinitely rich: there is always more to be known about it. More precisely, since acquisition of information takes time, one has to restrict oneself to knowledge about a certain part of the real world. This point also holds in principle. It does not depend, say, on the sensory system's sophistication. Given these properties of the real world and the limitations of any kind of physical agent, it follows that the real world is only partially knowable, and this in turn implies that it is predictable only to a limited extent.

Let us conclude our comparison of real and virtual worlds with a note on terminology. We have used the term "virtual" to designate closed, formal worlds such as chess. The term "virtual world" or "simulated world" is often used in a different sense in the areas of artificial life (e.g., Langton 1995) and virtual reality (Kalawsky 1993). Video games are a case in point; another example are Karl Sims's simulated physical worlds, in which artificial creatures evolve under various conditions, for example, on land or in the water (Sims 1994a,b). In these worlds, one can define new physical laws, new laws of nature, which is one of the things that makes them so fascinating. For example, if gravity is simulated in a virtual world, one can adjust g, the constant of gravity, and one can observe the change in the behavior of the (simulated) organisms that inhabit this world. From the perspective of the agents that live in such a virtual world, this virtual world does have some of the characteristics that we pointed out for real worlds. For example, unexpected and novel things happen—from the point of view of the agent! Often, new kinds of enemies emerge who have unknown powers. However, from the point of view of the programmer who created the virtual world, the very same events are neither new nor unexpected: he designed them into the system.

In summary, real worlds differ significantly from virtual ones. The problems of classical AI and cognitive science have their origin largely in a neglect of these differences.

3.2 Some Well-Known Problems with Classical Systems

In what follows we summarize some of the better-known problems with classical AI systems. Throughout the discussion we use the term classical AI systems to denote pure symbolic systems such as expert systems or traditional planning systems like STRIPS. The goal in this section is to describe the issues and problems that historically have motivated researchers to look for alternatives. There seems to be consensus within a large part of the research community in AI that classical systems, lack robustness and generalization capabilities, and cannot perform in real time. This makes them poorly suited for behaving in the real world. Moreover, they are, in essence, sequential; that is, they perform one operation after another. They also run on sequential machines, whereas the human brain is massively parallel in its processing. Let us briefly examine each of these points.

Robustness and Generalization: Traditional AI systems often lack robustness, which means that they lack tolerance of noise and fault tolerance and cannot behave appropriately in new situations. A system has noise tolerance if it functions appropriately when the data contain noise i.e. there are random fluctuations in the data. Sensors are always noisy, because they are physical devices, and motor acts are always imprecise, because they arise from physical devices. A system has fault tolerance if it performs adequately when some of its components break down. Standard symbol processing models are neither noise nor fault tolerant unless their programming explicitly provides for noise and particular types of faults. The most important deficiency of traditional AI systems in terms of robustness, however, is their inability to perform appropriately in novel situations, that is, their lack of generalization capacity. If a situation arises that has not been predefined in its programming, a traditional system breaks down or stops operating. Generalization ability is especially important in the real world, where no two situations are ever exactly the same.

Real-Time Processing: Because the real world has its own dynamics, systems must be able to react quickly in order to survive and perform their tasks. Systems based on the classical paradigm embedded in real robots are typically slow, because they process information centrally. Recall our overview of JL in chapter 2, in which a central information processing module was postulated (see principle c6 in table 2.3), and the discussion in chapter 1 of the

view that the brain is the "seat of intelligence". If all sensor signals have to be transmitted to a central device for processing (integration with other sensory signals, mapping onto internal representations, planning of action sequences) and finally generation of motor signals, real-time response can hardly be achieved.

Sequential Nature of Programs: The architecture of today's AI programs is essentially sequential, and they work on a step-by-step basis. By contrast, the human brain's processing is massively parallel, with activity occurring in many parts of the brain at all times. This problem arises from the fact that current computer technology is largely based on architectures of the von Neumann type which are, at the information processing level, sequential machines. As an aside, note that at the physical level a von Neumann machine is also massively parallel, just like any other system in nature.

Other Problems: Additional criticisms have been that classical systems are goal-based, are hierarchically organized, and process information centrally. The problems with goal-based systems are discussed in Montefiore and Noble 1989; the latter two problems are considered in chapter 11.

The criticisms of AI models presented so far are well-known and long-standing. Since the mid-1980s a number of additional ones have been raised pertaining to fundamental issues. Specifically, it has been argued that traditional AI models suffer from the frame problem and the problem of symbol grounding, and that they lack the properties of embodiment and situatedness.

3.3 The Fundamental Problems

In section 3.1 we pointed out that one of the problems with classical AI is that it did not give the real world sufficient consideration. In fact, all the fundamental problems of classical AI concern the relation of an agent and the real world, in particular its interaction with it. Chapter 4 outlines a systematic way of dealing with these relations. In this section, we discuss some specific problems: the frame problem, the symbol-grounding problem, and lack of embodiment and situatedness are treated in detail, and we briefly discuss the homunculus problem and the problem of the substrate required for intelligence.

The Frame Problem

The frame problem was originally pointed out by McCarthy and Hayes (1969) and has more recently attracted a lot of interest (e.g., Pylyshyn 1987). It comes in several variations and lacks one single, overriding interpretation. The central point concerns how to model change (Janlert 1987): How can a model of a continuously changing environment be kept in tune with the real world? Assuming that the model consists of a set of logical propositions (which essentially holds for any representation), any proposition can change at any point in time. Let us explain the frame problem using an example given by Daniel Dennett (1987), who has been working in the field of philosophy of the mind for many years. The initial situation described in Dennett's example is illustrated in figure 3.2, depicting a robot employing a propositional representation. It consists of a set of propositions like INSIDE(R1,ROOM), ON(BATTERY, WAGON), and so forth.

Once upon a time there was a robot, named R1 by its creators. Its only task was to fend for itself. One day its designers arranged for it to learn that its spare battery, its precious energy supply, was locked in a room with a time bomb set to go off soon. R1 located the room, and the key to the door, and formulated a plan to rescue its battery. There was a wagon in the room, and the battery was on the wagon, and R1 hypothesized that a certain action which it called PULLOUT(WAGON, ROOM) would result in the battery removed from the room. Straightaway it acted, and did succeed in getting the battery out of the room before the bomb went off. Unfortunately, however, the bomb was also on the wagon. R1 knew that the bomb was on the wagon in the room, but didn't realize that pulling the wagon would bring the bomb out along with the battery. Poor R1 had missed that obvious implication of its planned act.

Back to the drawing board. "The solution is obvious," said the designers. "Our next robot must be made to recognize not just the intended implications of its acts, but also the implications about their side-effects, by deducing these implications from the descriptions it uses in formulating its plans." They called their next model, the robot-deducer, R1D1. They placed R1D1, in much the same predicament that R1 had succumbed to, and as it too hit upon the idea of PULLOUT(WAGON, ROOM) it began, as designed, to consider the implications of such a course of action. It had just finished deducing that pulling the wagon out of the room would not change the colour of the room's walls, and was embarking on a

proof of the further implication that pulling the wagon out would cause its wheels to turn more revolutions than there were wheels on the wagon—when the bomb exploded.

Back to the drawing board. "We must teach it the difference between relevant implications and irrelevant implications," said the designers, "and teach it to ignore the irrelevant ones." So they developed a method of tagging implications as either relevant or irrelevant to the project at hand, and installed the method in their next model, the robot-relevant-deducer, R2D1 for short. When they subjected R2D1 to the test that had so unequivocally selected its ancestors for extinction, they were surprised to see it sitting, Hamlet-like, outside the room containing the ticking bomb, the native hue of its resolution sicklied o'er with the pale case of thought, as Shakespeare (and more recently Fodor) has aptly put it. "Do something!" they yelled at it. "I am," it retorted. "I'm busily ignoring some thousands of implications I have determined to be irrelevant. Just as soon as I find an irrelevant implication, I put it on the list of those I must ignore, and ..." the bomb went off. (pp. 41–42)

Let us briefly summarize the essential points of Dennett's example.

1. Assume that the symbolic description of the situation given in figure 3.2 is stored in R1's memory. It then has the problem of

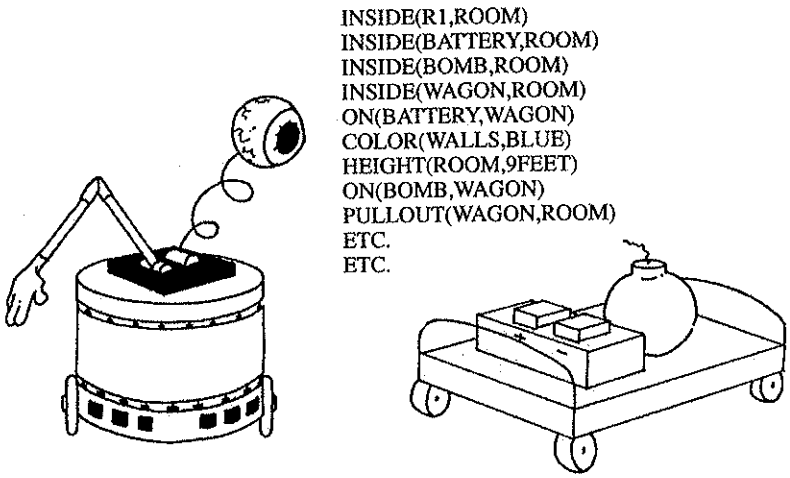


Figure 3.2 The frame problem. The robot R1/R1D1/R2D1 (R1 stands for robot, R1D1 for robot-deducer, and R2D1 robot-relevant-deducer) is standing near the wagon with a battery and a bomb. R1/R1D1/R2D1 uses a symbolic representation of the situation to draw inferences and guide its behavior.

determining the implications of an action. In this particular situation, the action of moving the wagon has the side effect that the bomb is also moving, since it is sitting on the wagon. Unfortunately, the robot does not know that this is relevant. What is obvious to a human observer has to be made explicit for R1.

2. R1D1 tries to take a vast number of potential side effects into account. Assessing all of these potential side effects takes a lot of time, and most are entirely irrelevant. For example, the fact that moving the cart does not change the color of the room is totally irrelevant in the current situation.
3. R2D1 tries to distinguish between relevant and irrelevant inferences. But in order to do this it has to consider all of them anyhow, which implies that R2D1 has no significant advantage over R1D1.

There have been a number of proposals for resolving the frame problem. One is the "sleeping dog strategy," in which the robot is programmed to assume that if something is not explicitly changed, it has not changed at all. Physical objects normally do not cease to exist if nothing happens to them, or they do not start to fly without reason, or the color of the room does not change significantly in a short period of time unless it is painted, and so forth. The robot then relies on this assumption in planning its course of action. However, ice cubes can melt, that is, they can cease to exist without an explicit manipulation of them. The bomb on the wagon changes its position if the wagon is moved. Either this fact must be represented explicitly, which would imply that there are very many relations of this kind, requiring significant memory space, collectively, for their representation, or the robot has to infer that the bomb will also move. As we have seen, however, there are typically a very large number of possible inferences that can be drawn and determining the relevance alone does not help (as poor R2D1 found). While the sleeping dog strategy is often useful, it does not completely resolve the frame problem. For example, it does not solve the problem of finding a way for the robot to determine the relevance of relations without having to check all the inferences.

Minsky (1975) and Schank and Abelson (1977) suggested that the robot's attention be focused on the relevant inferences by employing frames (or scripts). (Figure 2.7 offered an example of a script that focuses the attention on things happening in restaurants.) McCarthy (1980) suggested circumscription, which is also a way to restrict the number of inferences. All of these suggested

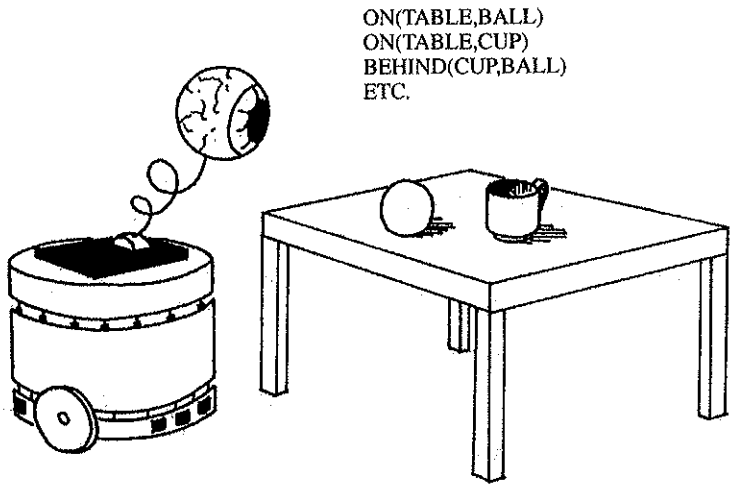


Figure 3.3 The frame problem and situatedness. R1/R1D1/R2D1 is standing in front of a table. From its current perspective, the cup is behind the ball, and this relationship is reflected in the symbolic description it uses to represent its environment. If R1/R1D1/R2D1 moves to the other side of the table, the symbolic description has to be updated, from the robot's perspective, the ball is now behind the cup. If a robot has a large set of such descriptions, many of them, but not all, may have to be updated as it moves around. Finding the right ones is a fundamental problem. For example, if R1/R1D1/R2D1 moves to the other side of the table, the relative position of the ball and the cup change, but the ball and the cup are still in exactly the same place. In the symbolic approach a way must therefore be found to reflect the change in the position of the objects relative to the robot without altering the robot's representation of their absolute positions. A situated agent can merely "look at" the situation.

solutions try to tackle the problem at the logical level, in a sense, on the inside. The problem, however, is really about the *system-environment* interaction: how models of a changing environment can be kept in tune with the environment. This is not a problem of logic, but rather one of modeling the world.

Another problem arises when modeling the real world that is related to the frame problem. R1D1 represents the situation shown on the right in figure 3.3 by means of a number of propositions. If R1D1 moves around the table, many of the propositions in the model R1D1 uses have to be updated, even though only the position of R1D1 is changing. In the real world it is not necessary for us to build a representation of the situation in the first place: We can simply look at it, which relieves us of the need for cumbersome updating processes. Moreover we can point to things when talking about them. As a robot, R1D1 could also take advantage of these possibilities—if designed properly.

According to Janlert (1987) the frame problem has two aspects. Our robots R1, R1D1, and R2D1 were suffering from one, the *prediction problem*, which has to do with determining what is relevant. The other, called the *qualification problem*, is equally nasty: It involves the preconditions under which an action can be applied. For example, if you are getting into a car, you have to assume that there is no bomb in the car, that nobody put sugar into the gas tank, that nobody has taken out the engine, that no skunk is in the car, that no lion is in the car, that the clutch is still in the same place, and so forth almost infinitely. Another example is that when sitting down on a chair, you do not explicitly assume that it will not break. You do not have to do that because you can be confident that if there were a problem you would recognize it. (But note that this strategy may occasionally fail, and you might indeed land on your behind on the floor.) Humans certainly do not explicitly assume that these preconditions are given. Because we are "grounded" in our environment, we know the things we have to check. To function properly in a changing environment, a robot must somehow be provided with the same capacity.

The frame problem is a fundamental one, and it is intrinsic to any world modeling approach whatsoever. Any model of a changing environment presents a frame problem; the more sophisticated and elaborate the model, the more the frame problem shows up. Thus, we see that the frame problem exists not only for traditional AI models but for models in general. An important goal of intelligent systems design is to minimize the implications of the frame problem. Embodied cognitive science's approach is to minimize the amount of world modeling in the first place.

The Symbol-Grounding Problem

The symbol-grounding problem, which refers to how symbols relate to the real world, was first discussed by Steven Harnad (1990). In traditional AI, symbols are typically defined in a purely syntactic way by how they relate to other symbols and how they are processed by some interpreter (Newell and Simon 1976; Quillian 1968); the relation of the symbols to the outside world is rarely discussed explicitly. In other words, we are dealing with closed systems, not only in AI but in computer science in general. Except in real-time applications, the relation of symbols (e.g., in database applications) to the outside world is never discussed; it is assumed

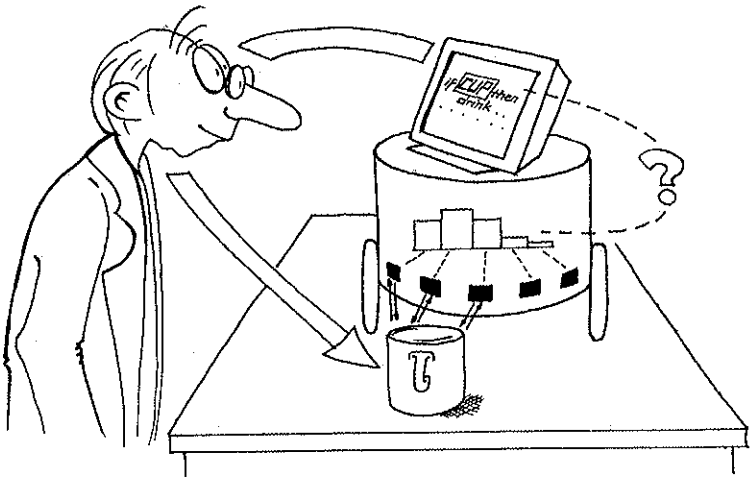


Figure 3.4 The symbol-grounding problem. The scientist has no difficulty associating the cup in the real world with the symbol "cup" on the screen standing on top of the robot. But if the robot is programmed with symbols representing objects and has to interact with its environment on its own, it has to be able to map the sensory stimulation (from the cup itself) onto its internal symbolic representation (the word "cup")—a very hard problem.

as somehow given, with the (typically implicit) assumption that designers and potential users know what the symbols mean (e.g., the price of a product). This idea is also predominant in linguistics: it is taken for granted that the symbols or sentences correspond in some way with the outside world. The study of meaning then relates to the translation of sentences into some kind of logic-based representation whose semantics are clearly defined (Winograd and Flores 1986, p. 18).

Using symbols in a computer system is no problem as long as there is a human interpreter who can be safely expected to be capable of establishing the appropriate relations to some outside world: the mapping is "grounded" in the human's experience of his or her interaction with the real world. However, once we remove the human interpreter from the loop, as in the case of autonomous agents, we have to take into account that the system needs to interact with the environment on its own. Thus, the meaning of the symbols must be *grounded* in the system's own interaction with the real world, as figure 3.4 illustrates. Symbol systems, such as computer programs, in which symbols refer only to other symbols are not grounded because they do not connect the symbols they employ to the outside world. The symbols have meaning only to a designer or a user, not to the system itself. The robot in figure 3.4 is

in trouble because it is trying to map a sensory stimulation, a cup, onto an internal symbol, the word "cup." Providing the robot with this capacity is very hard to do, even in simple cases, let alone for more complex ones. But this mapping will always have to be present if there are symbols in the system. (As we argue later, the symbol grounding problem is really an artifact of symbolic systems and "disappears" if a different approach is used. Specifically, in chapter 12 we show how "concepts" can evolve in the interaction of an autonomous agent with its environment, without the need for introducing symbols of any sort within the agent. We put "concepts" in quotes to indicate that we do not mean symbolic concepts.) For a long time, the symbol-grounding problem attracted little attention in AI or cognitive science, and it has never been an issue in computer science in general. Only with the renewed interest in autonomous robots has it reemerged.

The Problems of Embodiment and Situatedness

The problem of embodiment refers to the fact that abstract algorithms do not interact with the real world. Rodney Brooks forcefully argued that intelligence requires a body (Brooks 1991a,b). Only if a system is embodied do we know for sure that it is able to deal with the real world. Moreover, systems that are not embodied all suffer from the symbol-grounding problem. Their connection to the outside world requires a human interpreter in the loop.

Many researchers in AI have recognized this problem. For example, Margaret Boden noted, in *Artificial Intelligence and Natural Man* (1977):

In everyday life you usually remember your "place" largely because the external world is there to remind you what you have or haven't done. For instance, you can check up on whether you have already added the vanilla essence by sniffing or tasting the mixture, or perhaps by referring to the pencil and paper representation of the culinary task that you have drawn up for this mnemonic purpose. A computational system that solves its problems "in its head" rather than by perceiving and acting in the real world, or pencil and paper models of it, has to have all its memory aids in the form of internal representations. (p. 373)

At the time the importance of real-world interaction in controlling behavior was fully recognized, however, the implications—

embodiment—had not been further elaborated; they were fully understood only when people started to use robots for the study of intelligence. As embodied systems, robots have the potential to "solve" the symbol-grounding problem, but this requires them to have "situatedness."

An agent is "situated" if it can acquire information about the current situation through its sensors in interaction with the environment. A situated agent interacts with the world on its own, without an intervening human. To illustrate this point, let us look at an example of a system entirely lacking situatedness. Imagine a remote-controlled device without sensors, such as a remote-controlled toy car. The toy car is controlled only by information from the operator; it has no information about the current situation from its own perspective. A situated agent has the potential to acquire its own history, if equipped with appropriate mechanisms. To understand situatedness and to design situated agents, we have to adopt the agent's perspective, rather than the observer's. For understanding situated agents (e.g., animals), it is important to realize that the world may look very different from the perspective of the animal than from our own. Ants, for example, have completely different eyes so what they see is not what we see. In designing situated agents, adopting the agent's perspective is important because the programs that control the agent's actions are based on the sensor data the robot gets. Since the relation between observer and agent is of fundamental importance, we discuss it in more detail in chapter 4. It turns out that situated agents, that is, agents having the property of situatedness, are much better at performing in real time because they exploit the system-environment interaction and therefore minimize the amount of world modeling required.

Note that embodiment does not automatically imply situatedness. Agents can be equipped with detailed models of their environment to be used in the planning processes. If these plans are employed significantly in controlling the agent's behavior, it will not be situated. Moreover, as we saw in the last chapter when discussing, plan-based systems quickly run into combinatorial problems (cf. also Chapman 1987). If the real world changes, one of the main problems is keeping the models in tune with the environment. Inspection of the problem of behaving in the real world shows that it is neither necessary nor desirable to develop very comprehensive and detailed models (e.g., Brooks 1991a; Suchman

1987; Winograd and Flores 1986): the more comprehensive and the more detailed the models, the more strongly the agent is going to be affected by the frame problem. Typically only a small part of an agent's environment is relevant for its behavior. In addition, instead of performing extensive inference operations on internal models or representations, the situated agent can interact with the current situation: The real world is, in a sense, part of the "knowledge"[1] the agent needs to behave appropriately. It can merely "look at it" through the sensors. In a sense, the world is its own best model. Figure 3.3 illustrates this point.

The concept of situatedness has recently attracted a lot of interest and led to heated debates about the nature of intelligence and the place of symbol-processing systems in studying intelligence. For example, a complete issue of the journal *Cognitive Science* in 1993 was dedicated to the role of situatedness in cognitive science (see also Clancey 1997).

Other Fundamental Problems

A number of other problems with classical systems can be found in the literature, for example, the homunculus problem and the problem of the underlying substrate. "Homunculus" literally means "little man"; as used here, it designates a "little man in the head." The *homunculus problem*, or the *homunculus fallacy*, as it is also called, refers to circular accounts of psychological processes. These processes are circular because they ascribe to some internal mechanism (the homunculus) the very psychological properties being investigated in the first place. For example, a theory of vision might postulate that there is within the brain a mechanism that scans, views, or inspects images on the retina. Such a theory would be vacuous, however, since scanning, viewing, and inspecting are all instances of the very visual processes the theory was supposed to illuminate in the first place (Gregory 1987, p. 313). In other words, the theory has assumed the very things it set out to explain. When used to criticize AI systems, the term "homunculus" designates a subsystem that executes a function specified in purely formal terms (as in the cognitivistic paradigm). In a sense, a homunculus is required to perform the function that the formal system is intended

[1] We put "knowledge" within quotation marks to indicate that this is not the standard way of using knowledge in AI. The standard way refers to knowledge structures that are represented internally.

to explain. For example, we saw that the robot R2D1 was lacking a means to determine the relevant inferences. With respect to the homunculus problem, the real problem is that it is not possible to determine the relevance of an inference on a purely formal basis (i.e., by inspecting only its database of symbolic representations and drawing inferences): a link to the environment and thus to the meaning of the representation is required. In other words, the homunculus problem and the symbol grounding problem are closely related: a system containing ungrounded symbols will always require a homunculus giving meaning to them. We do not explore the subtleties of this argument, any further (for a more comprehensive discussion, see, e.g., Edelman 1992 or Bursen 1980).

To bring our review of some fundamental problems to an end let us mention one which is still fairly prominent, the *problem of the underlying substrate*. There is a folklore that true intelligence requires a biological substrate as a basis. Only natural brains can, in this folklore, exhibit "true intelligence." Note that this issue does not only apply to classical AI, but rather concerns any endeavor to build intelligent systems. As far as we can tell there is to date no evidence demonstrating the in-principle impossibility of having intelligence based on substrates other than natural brains. But even if it turned out to be true that a biological substrate were required, we could still use computers and robots to build models.

3.4 Remedies and Alternatives

In this final section, we briefly examine a number of possible ways to deal with the problems we have raised. Again, the overview is very short and the field is very large. Because we want to leave room to present embodied cognitive science, we cannot possibly do justice to all the research that has been done. We have labeled the various positions we present "pessimist," "traditionalist," "pragmatist," and "optimist." These labels are not to be taken too seriously.

The Pessimist: Giving Up. The pessimist knows the fundamental problems of traditional approaches to AI, believes these criticisms to be universally valid, and strongly doubts that there are viable alternatives. For him, the only solution is to give up on the endeavor to build intelligent systems. An example of this position can be seen in the implications of Winograd and Flores' *Understanding Computers and Cognition* (1986), which represents a fundamental

criticism of traditional AI and the traditional understanding of intelligence, in particular, natural language. Winograd and Flores' suggestion is to build computer systems that support human activity, in order to support and enhance human intelligence, rather than trying to build computer systems that are themselves intelligent, which is, in their opinion, a futile effort. This view is maintained by a relatively strong group in the area of software engineering that capitalizes on "designing for humans." Greenbaum and Kyng (1991) offer an interesting overview of this field.

The Traditionalist: Improving Classical Methods. Many researchers in traditional AI and psychology have realized the problems with classical approaches. Clearly, there is a lot of room for improvement. Such researchers have pursued solutions intended to overcome the problems classical approaches present. Problems with generalization and robustness, for example, can largely be overcome with neural networks. Neural networks are also massively parallel and thus less subject to the criticism of being sequential. Then, there is a large field dealing with situated planning where high-level plans are used but they are no longer employed to tightly control behavior, but as resources that can be accessed whenever required (For reviews of this approach, see, e.g., Hasemann 1995; Wolfe and Chun 1992). Methods in computational vision have also been improved significantly. The processors have become so fast that real-time issues become less and less of a problem. This list could still be extended considerably.

The Pragmatist: Working Toward Practical Applications. The pragmatist is not worried about the foundations: His goal is to get things to work. For him, the ultimate test of whether a solution works is if it can be deployed and routinely used in everyday working environments. Whether a program is labeled "expert system," "decision support system," or "intelligent agent" is entirely irrelevant to the pragmatist—except insofar as it might help sales. The pragmatist is also free to combine various techniques and approaches. For example, neural networks have wonderful properties: They can learn and are adaptive. They are ideal for taking care of low-level sensory-motor control. Rule-based systems have the advantage that they can be quickly built and are easy to understand. Moreover, they can be connected to symbolic planning systems, with the idea that neural networks connect the low-level sensory-motor systems to the high-level symbolic layers. The presence of a symbolic layer has the advantage of facilitating

communication between the human and the robot. The pragmatist's point is, Does it work? Do people think they are getting their money's worth? This is a perfectly acceptable position, but not the one adopted in this book. It is our conviction that ultimately, the pragmatist will benefit from the research described here.

The Optimist: Embodied Cognitive Science. In spite of the improvements achieved by the traditionalist we feel that a radically different approach is required. We now embark on this endeavor.

Issues to Think About

Issue 3.1: Prerequisites for Intelligence

In our discussion of the fundamental problems of classical AI, we briefly mentioned the problem of the underlying substrate, the view that a biological substrate is a prerequisite for intelligence. The implication is that there can in principle be no artificial systems that exhibit intelligent behavior. The remainder of this book, however, is for the better part concerned with such synthetic agents. Before reading on, we would like you to reflect for a moment on your own view on this topic. Do you think that, indeed, a biological brain and body is needed for intelligent behavior to emerge, or are you willing to ascribe intelligence to artificial agents? In the latter case, what would agents have to do in order for you to describe their behavior as intelligent?

Issue 3.2: The Symbol-Grounding Problem

Take a concept from your everyday life, for example, "drinking." Now try to make explicit what "drinking" means to you. You may be surprised how tightly concepts are tied to the body, are grounded in sensory-motor experiences. Just to get you started, here are a few points. Drinking relates to liquids; liquids are kept in particular containers like cups or glasses. They can be hot or cold; if they are hot you can get burned. If you grasp the coffee cup, you move it to your mouth slowly. Why? Because you know that liquids spill when you move the cup fast. You then move it close to your lips until it touches them, which you can feel both on your lips and from the feedback from your arm muscles. You then tilt the cup and move your lower lip forward so the liquid can drop into your

mouth. You are applying the physical law that the surface of the liquid stays horizontal as the container moves. Then you sense the liquid and its temperature in your mouth, on your lips, and perhaps in your throat and stomach. You also recognize various liquids by their specific reflective properties, viscosity, and so forth. This is what sensory-motor grounding is all about. Now try to do the same thing with an object like a newspaper. How about with more abstract concepts, like "responsibility"?

Points to Remember

- Classical AI systems have been criticized on various grounds: that they lack robustness and generalization capabilities, and cannot perform in real time. Moreover, they are sequential and run on sequential machines. Additional points of criticism have been that they are goal based and organized hierarchically, and that their processing is done centrally.
- Real worlds differ significantly from virtual ones. Virtual worlds have states, there is complete information about them, the possible operators within them are given, and they are static. The real world is quite different. In particular, the real world has its own dynamics, which force the agents to act in real time.
- The frame problem concerns how models of parts of the real world can be kept in tune with the real world as it is changing. It is especially hard to determine which changes in the world are relevant to a given situation without having to test all possible changes. The frame problem has two aspects, a prediction problem and a qualification problem.
- The symbol-grounding problem concerns how symbols relate to the real world. The symbol-grounding problem becomes obvious if the human observer is taken out of the loop and the system must interact on its own with the environment. It is a characteristic of symbolic approaches; nonsymbolic approaches do not have a symbol-grounding problem.
- An agent is situated if it acquires information about its environment only through its sensors in interaction with the environment. A situated agent interacts with the world on its own, without an intervening human. It has the potential to acquire its own history if equipped with appropriate mechanisms.
- Although there have been many suggestions for resolving the fundamental problems with classical systems, we think that the solu-

tion can be achieved only through a new approach that capitalizes on an agent's interaction with the world. This is the major concern of embodied cognitive science.

Further Reading

Brooks, R. A. (1991). Intelligence without reason. *Proceedings of the International Joint Conference on Artificial Intelligence*, 569–595. (An important paper by the founder of the field of behavior-based intelligence. It summarizes the major criticisms of the classical approach by drawing on the vast literature in various fields like robotics, psychology, neurobiology, and computer science.)

Clancey, W. J. (1997). *Situated cognition: On human knowledge and computer representations.* New York: Cambridge University Press. (An elaborate text summarizing 10 years of research trying to come to grips with the large literature on knowledge engineering, knowledge representation, symbolic AI, and subsymbolic approaches, as well as neurobiological theories. Proposes the metaphor of "situated cognition" as an alternative to the classical symbol processing view.)

Harnad, S. (1990). The symbol grounding problem. *Physica D 42*, 335–346. (The standard reference to the symbol-grounding problem. Whereas Harnad is trying to solve the symbol-grounding problem, we think that a different approach—as described in this book—should be used, so that this problem does not occur in the first place.)

Pylyshyn, Z. W. (Ed.). (1987). *The robot's dilemma: The frame problem in artificial intelligence.* Norwood. NJ: Ablex. (An interesting collection of papers about the frame problem.)

Winograd, T., and Flores, F. (1986). *Understanding computers and cognition.* Reading, MA.: Addison-Wesley. (*The* criticism of AI, and more generally the rationalistic view in computer science. A must for any computer scientist, AI researcher, or cognitive scientist.)