4 Intelligent Systems: Properties and Principles

In the 1960s, the Japanese psychologist Masanao Toda proposed to study hypothetical creatures he called “Fungus Eaters” as a fun way to think about intelligence, an alternative to the traditional methods of academic psychology. Fungus Eaters are artificial creatures that are sent to a distant planet to collect uranium ore. Because they have to collect ore, they must be physical systems, i.e., they must be embodied—a computer simulation simply wouldn’t do. Also, since there are no people on this planet, the Fungus Eaters have to be autonomous—i.e., independent of human control; they should be self-sufficient, which means that they should be able to take care of themselves over extended periods of time, and they must to be situated, i.e., they have to be able to learn about the environment through their own sensory systems. These hypothetical creatures are called Fungus Eaters because they feed on a particular type of fungus that grows on the planet. The planet is so far away that they cannot be remote controlled because the signals take too long to travel between Earth and the planet. By comparison, NASA’s engineers wanted to maintain as much control as possible over the Mars Sojourner, because apparently they did not fully trust its autonomous operating abilities. As a compromise, the robot was extremely slow; it traveled only a few meters per day, adding up to a little over 100 meters in three months. Sojourner’s replacements, the twin Mars exploration rovers Spirit and Opportunity, can travel more than 100 meters per day (very speedy compared to Sojourner), but the target locations to which they have to move are still commanded from Earth. Toda’s Fungus Eaters illustrate the many challenges facing a complete agent: it must fend for itself, deal with unforeseen situations, create its own objectives, and forage for energy, among other things. In traditional artificial intelligence, on the other hand, agents were much more limited and did not have to deal with all of the difficulties of the real world.

Toda further argued—and many psychologists would probably agree with him—that in laboratory studies people are often tested on tasks that are not only somewhat artificial but also unusually difficult for humans: subjects are asked, for example, to remember long lists of numbers or to read text upside down. Toda stressed that if we are to learn something relevant about intelligence—something that holds true in real-world behavior—we need to study *complete* systems, i.e., systems that have to act and perform tasks autonomously in the real world (Toda, 1982). While Toda's Fungus Eaters provide a rough intuition about the sorts of systems we are interested in, we will make the notion of complete agents more precise in this chapter.

In the previous chapter we outlined what a theory of intelligence should look like, and we discussed some of the general theoretical considerations in the study of intelligent systems: diversity-compliance, frame of reference, the synthetic methodology, time scales, and emergence. But we have not yet said much about how to actually design real agents when applying the synthetic methodology; we will do so in this chapter. The agents we are interested in designing are complete creatures—Fungus Eaters, so to speak—endowed with everything needed to behave in the real world, which obviously implies that they have to be embodied and situated, autonomous, and self-sufficient. All of the robots that we discuss in this book are autonomous in the simple sense that they are not directly controlled by a human. Of course, their level of autonomy is still very limited because they depend on humans for their energy supply, for maintenance, and to be placed in their proper task environment. Clearly, like intelligence, autonomy is not an all-or-none property; an agent may be controllable to a greater or lesser extent by another agent. There is a long-running philosophical debate about the concept of autonomy and how it relates to intelligence, but we will not go into that debate here; the interested reader is referred to Pfeifer and Scheier, 1999.

In this chapter we will briefly describe what we mean by the "real world," and contrast it to virtual ones. Then we will discuss the properties of embodied agents and describe what happens when they interact with the real world. Finally, we will introduce the basic set of design principles.

4.1 Real Worlds and Virtual Worlds

This book is about embodied agents that have to function in the real world. The real world has properties very different from those that characterize virtual or formal worlds, and intelligent agents have to be able to deal with the physical world if they are to survive or function in it for an extended period of time. Moreover, unlike virtual worlds, the real world challenges an agent in various ways. First, because real-world agents are embodied, acquisition of information always takes time: if I want to know who is in the room next door, I have to go there and look, call them, or ask someone.

Second, the information that an agent can acquire about the real world is always very limited: we can only see what is in the range of our visual field or hear the sounds that reach our ears. Thus we can never have complete information. This situation is different from a formal game like chess, where knowledge of the board position constitutes all the information about the state of the game, assuming that the strategies of the players are not part of the game proper. Moreover, it is not clear what "complete information" in the real world would mean in the first place: would it imply that an agent must have knowledge about the state of all the atoms in the universe? This is clearly an absurd idea. One way of summarizing information about a part of the real world is to make an abstract model of it. For example, we can characterize a lecture hall by specifying the number of students in it, the temperature, the light settings, and whether the projector is on or off, which for many purposes will be entirely sufficient. But such a model abstracts away most of the potential information available: it does not contain anything about the students' blood flow or their thoughts about the quality of the lecture.

Third, physical devices are always subject to disturbances and malfunctions, and since sensors are physical devices, the information acquired through them will always contain errors. From these considerations it follows that since knowledge about the real world is always very limited, it is therefore intrinsically uncertain and only predictable to a limited extent: For example, if it's noisy you may not hear the car that is approaching you from behind because of the physical limitations of your ears: they only deliver the summed noise, so that you may not pick out the sound of the car. Note that this point holds irrespective of the speed and accuracy of the agent's sensors: even if we have an ultra-high-resolution camera, if it suddenly gets dark, the images it delivers will be blurred and noisy. The uncertainty and limited predictably of information collected from the real world is a principle that holds for any agent.

Fourth, the real world is not characterized by clearly defined, discrete states: the weather is never simply good or bad, but rather sunny, cloudy, misty, rainy, windy, or dull, all to greater or lesser extents. Because there are no discrete states, there are therefore no clearly defined actions that can be executed when the world is in a particular state: it is a good idea to take your umbrella with you when it is raining outside, but what if it is only cloudy, or raining a little bit, or perhaps likely to rain later? This lack of definable states is different from formal worlds like chess, where there are uniquely prescribed board positions—a piece either is or is not occupying a square—and for every board position there is a finite set of possible moves from which a player has to choose.

Fifth, agents in the real world always have several things to do simultaneously: animals have to eat and drink, but they also have to take care that they are not eaten by predators, they have to build nests, clean themselves, breathe, fight off infection, reproduce, and care for their offspring. Similarly, robots which have to function in the real world always have many tasks to perform in parallel. For example, a robot designed to serve coffee to employees in an office has to keep itself functioning, recharge its batteries, avoid breaking or bumping into things, and not harm humans, all while it is serving coffee. In contrast, in the formal world of chess there is only one thing to do: make one move at a time in order to win the game.

Sixth, because the real world has its own dynamics—things out in the world happen even if we do not do anything—there is always time pressure due to ongoing change. Thus agents are always forced to act, whether they want to or not. In many formal settings an agent can take as long as needed to decide which action to take. And finally, related to this point, the real world is a highly complex dynamical system, making it intrinsically unpredictable because of its nonlinear nature and its sensitivity to initial conditions (see focus box 4.1). (Herbert Simon has coined the term *bounded rationality* to designate, in essence, decisions that have to be taken under such circumstances [Simon, 1976, 1969]).

To summarize before continuing, the real world requires time to extract information from it, and extraction is always partial and error-prone; it is not neatly divisible into discrete states; it requires agents operating in it to do several things at once; and finally the real world changes of its own accord, not only in response to agent action. So, the real world is challenging and "messy." Clearly, there are several constraints that a physical agent faces as a result of being in the real world: there are certain things it simply cannot do, such as extract noise-free information instantaneously from the environment. In the next section we will describe how these constraints place certain hard limitations on real-world agents, but also provide them with opportunities. These

Focus Box 4.1
Dynamical Systems

There is a vast literature on dynamical systems, and although at a high level there is general agreement on the basic concepts, a closer look reveals that there is still a considerable diversity of ideas. We will use the terms *dynamical systems*, *chaos*, *nonlinear dynamics*, and *complex systems* synonymously to designate this broad research field, although there are appreciable differences implied by each of these terms. Our purpose here is to provide a very short, informal overview of the basic notions that we need for the book. Although we do not employ the actual mathematical theory, we will make use of the concepts from dynamical systems theory because they provide a highly intuitive set of metaphors for thinking about physically embodied agents and groups of agents.

A dynamical system in the real world is one that changes according to certain laws: examples include the quadruped robot Puppy, human beings, economical systems, the weather, a swinging pendulum, or a society of monkeys. Dynamical systems can be modeled using differential equations (or their discrete analogs, difference equations). The mathematical theory of dynamical systems investigates how the variables in these equations change over time: for example the angles of Puppy's joints can be used as variables in a set of differential equations that describe, mathematically, how the robot moves. However, to keep matters simple, we will not use differential equations in this book.

The dynamical systems we look at here are nonlinear because interesting systems in the real world are typically nonlinear. One of the implications of nonlinearity is that we can no longer, as we can with linear systems, decompose the systems into subsystems, solve each subsystem individually, and then reassemble them to give the complete solution. In real life, this principle fails miserably: if you listen to two of your favorite songs at the same time, you don't double your pleasure! (We owe this example to Strogatz, 1994.) Similarly, we cannot understand the motion of one of Puppy's legs without considering how it is affected by the other three. In other words, the system must always be treated as a whole (see the complete-agent principle). Another important property of nonlinear systems is their *sensitivity to initial conditions*: if the same system is run twice using very similar initial states, after a short period of time, they may be in completely different states. This is also in contrast to linear systems, in which two systems started similarly will behave similarly. The weather is a famous example of a nonlinear system—small changes may have enormous effects—which is what makes weather forecasting so hard.

The *phase space* of a system is the space of all possible values of its important variables. For Puppy we could, for example, choose the joint angles as important variables and characterize its movement by the way the angles change over time. If there are two joints per leg, this yields an eight-dimensional phase space: each point in phase space represents a set of values for all eight joints. (Alternatively, we could use the contact sensors on the feet only, a different and simpler way of defining the phase space, which would then be only four-dimensional). Neighboring points in phase space represent similar values of the joint angles. As Puppy runs, the joint angles change continuously. Thus we can say that these changes are analogous to the way the point in phase space (the values of all joint angles at a particular moment) moves over time. The path of this point in phase space, i.e., the values of all these joint angles over time, is called the *trajectory* of the system.

An *attractor state* is a preferred state in phase space toward which the system will spontaneously move if it is within its *basin of attraction*. There are four types of attractors: point, periodic, quasi-periodic, and chaotic. Physical systems, such as Puppy, by their very nature as physical systems, have attractor states. It is important to realize that the attractors will always depend on the way the actuators are driven and on the environmental conditions.

Focus Box 4.1
(continued)

If Puppy runs and settles into a particular gait, the joint angles, after a short period of time (less than 1 sec), will more or less repeat, which means that the trajectory will return to roughly the same location as before: the values of the joint angles will be very similar to what they were in the previous cycle. This cyclic behavior is known as a *periodic attracto*r, or, because the angles in the real world never exactly repeat, a *quasi-periodic attractor*. Puppy's different gaits correspond to different (quasi-) periodic attractors: this is illustrated by figure 4.2. If Puppy falls over and stops moving, then its joint angles no longer change over time, and the trajectory in the phase space remains at a single point: such points are called—not surprisingly—*point attractors*. Finally, if the trajectory moves within a bounded region in the phase space but is unpredictable, this region is called a *chaotic attractor*. Systems tend to fall into one of their attractors over time: the sum of all of the trajectories that lead into an attractor is known as the *basin of attraction*. Attractors—and this is relevant for our ideas on emergence of cognition (see chapter 5)—are discretely identifiable entities within a continuous system: Puppy's joint angles change smoothly over time, but we can reliably tell whether Puppy is walking, running, or standing still.

Again, there is a frame-of-reference problem here. How do you know the system is in an attractor state? And how does the agent itself know it? So, you need to provide some way of measuring the system's change over time: for example, if you are interested in locomotion, you can measure joint angles using sensors (as in the example given), or you can put pressure sensors on the feet. On the basis of these measurements, the robot (or the researcher) can then detect its attractor states and may change its actuation pattern: changing the frequency of actuation and the phase difference between front and hind legs (e.g., when the front legs start stretching, the hind legs may start bending), alters the dynamics and thus the system might transition into another attractor state, such as from walking to running. While the notion of an attractor is powerful and has intuitive appeal, it is clear that transitions between attractor states are equally important, e.g., for generating sequences of behavior.

Attractors, together with the transitions between them, reflect in some sense the natural dynamics of the system, in our case the agent. If the agent is driven by an oscillator (to generate periodic motion), the complete system will, depending on the frequency, settle into a (quasi-periodic) attractor state whose period is emergent from the coupling of the neural and the physical system yet different from the period dictated by the oscillator. This phenomenon is known as *mutual entrainment*: the resulting frequency will represent a "compromise" between the systems involved (see also our discussion of Sten Grillner's experiments on the Lamprey in chapter 5).

For those who would like to know more about the mathematical foundations of dynamical systems we recommend Strogatz (1994), and for those interested in its application to cognition, Port and van Gelder (1995) and Beer (2003).

limitations and opportunities can be described as a set of properties that all complete agents share.

4.2 Properties of Complete Agents

Here are the most important properties of complete agents that follow from their embodied nature:

1. *They are subject to the laws of physics* (energy dissipation, friction, gravity).
2. *They generate sensory stimulation* through motion and generally through interaction with the real world.
3. *They affect the environment* through behavior.
4. *They are complex dynamical systems* which, when they interact with the environment, have *attractor states*.
5. *They perform morphological computation.*

The interesting point here is that these properties are simply unavoidable consequences of embodiment. These are also the properties that can be exploited for generating behavior, and how this can be done is specified in the design principles. Before we go on to the design principles, let us briefly clarify each of these properties.

1. *A complete agent is subject to the laws of physics.* Walking requires energy, friction, and gravity in order to work. Because the agent is embodied, it is a physical system (biological or not) and thus subject to the laws of physics from which it cannot possibly escape; it must comply with them (see also our discussion of compliance in chapter 3). If an agent jumps up in the air, gravity will inevitably pull it back to the ground.

2. *A complete agent generates sensory stimulation.* When we walk, we generate sensory stimulation, whether we like it or not: when we move, objects seem to flow past us (this is known as optic flow); by moving we induce wind that we then sense with our skin and our hair; walking also produces pressure patterns on our feet; and we can feel the regular flexing and relaxing of our muscles as our legs move.

3. *A complete agent affects its environment.* When we walk across a lawn, the grass is crushed underfoot; when we breathe, we blow air into the environment; when we walk and burn energy, we heat the environment; when we drink from a cup, we reduce the amount of liquid in the glass; when we drop a cup it breaks; when we talk we put pressure waves

out into the air; when we sit down in a chair it squeaks and the cushion is squashed.

4. *Agents tend to settle into attractor states.* Agents are dynamical systems, and as such they have a tendency to settle into so-called attractor states. Horses, for example, can walk, trot, canter, and gallop, and we—or at least experts—can clearly identify when the horse is in one of these walking modes, or gaits, the more technical word for these behaviors. These gaits can be viewed as attractor states. The horse is always in one of these states, except for short periods of time when it transitions between two of them, for example from canter to gallop. We should point out here that the attractor states into which an agent settles are always the result of the interaction of three systems: the agent's body, its brain (or control system), and its environment. Because the concept of dynamical systems and attractor states is important for our arguments, we will elaborate it a bit more by returning to the case study of Puppy, the four-legged running robot that we introduced in chapter 3 (see also focus box 4.1 and chapter 5).

5. *Complete agents perform morphological computation.* By "morphological computation" we mean[1] that certain processes are performed by the body that otherwise would have to be performed by the brain (see figure 4.1). An example is the fact that the human leg's muscles and tendons are elastic so that the knee, when the leg impacts the ground while running, performs small adaptive movements without neural control. The control is supplied by the muscle-tendon system itself, which is part of the morphology of the agent.

It is interesting to note that systems that are not complete, in the sense of the word used here, hardly ever possess all of these properties. For example, a vision system consisting of a fixed camera and a desktop computer does not generate sensory stimulation because it cannot produce behavior, and it influences the environment only by emitting heat and light from the computer screen. Moreover, it does not perform morphological computation and does not have physical attractor states that could be useful to the system.

The Quadruped Robot Puppy as a Dynamical System

In what follows we will use the robot Puppy to illustrate how cognition might emerge from the simple, basic actions of walking or running. We have tried to capture this idea of going from locomotion to cognition with the phrase "bootstrapping cognition from the bottom up," in order

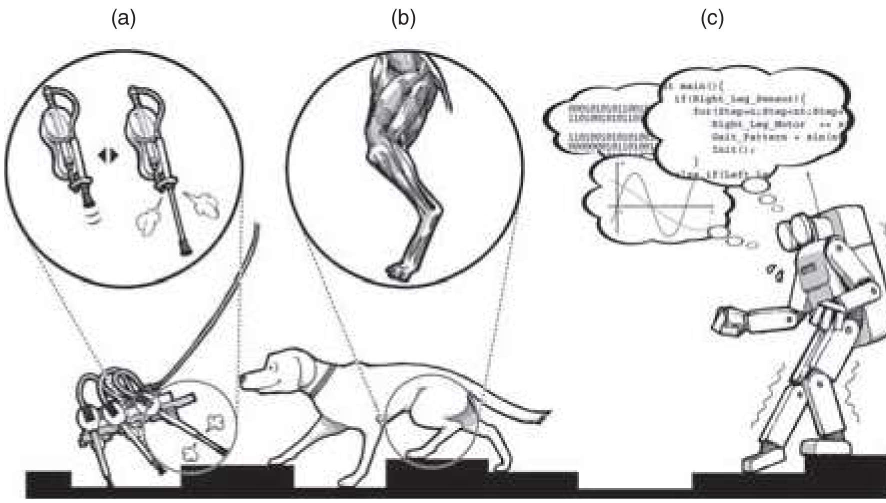

Figure 4.1
Morphological computation. (*a*) Sprawl robot exploiting the material properties of its legs for rapid locomotion. The elasticity in the linear joint provided by the air pressure system allows for automatic adaptivity of locomotion over uneven ground, thus reducing the need for computation. (*b*) An animal exploiting the material properties of its legs (the elasticity of its muscle-tendon system) thus also reducing computation. (*c*) A robot built from stiff materials must apply complex control to adjust to uneven ground and will therefore be very slow.

to distinguish it from the goal of traditional AI, which was to somehow program "thinking" directly into a computer.

We mentioned that running is considered a hard problem in robotics. Running by definition includes a certain time when all legs are off the ground, which is known as the flight phase; the stance phase refers to the rest of the time, when one or more feet are on the ground. Figure 3.2 shows some of the details of Puppy's morphology. Continuing our description from chapter 3, there are two springs attached to each leg, inspired by the muscle-tendon systems in four-legged animals. Also, there is a strong elastic metal blade that can bend up and down, providing the robot with a spine that is flexible, although somewhat different in design from the segmental spines of animals or the humanoid robot Kenta (Japanese for "tendon boy"; we will come back to Kenta in chapter 5). The springs and blade give Puppy a more dynamic and organic feel compared to most other robots, which are tightly engineered and move rigidly using complex control programs and strong motors: this aspect of traditional robotics is parodied by a dance called the Robot that was popular in the 1970s and the 1980s, which required the dancer to hold

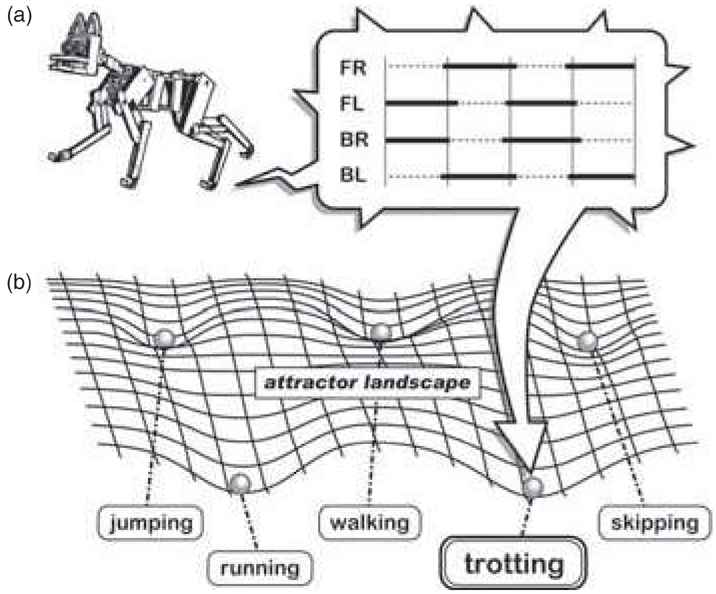

Figure 4.2
Attractor states. (*a*) Different gait patterns for Puppy as recorded from pressure sensors on the feet: the dark lines in the graph indicate when a foot is touching the ground; the dotted lines indicate when it is not. These gait patterns correspond to attractor states of the joint physical/neural system. (*b*) The same gait patterns shown in the "attractor landscape." The gait patterns correspond to minimum energy basins in the attractor landscape.

his body rigid and produce a disconnected series of localized, discrete movements.

The body, the legs and the feet are built from aluminum, which implies that on most surfaces the feet will slip a little. This slippage turns out to be an important factor in stabilizing the robot when it is running: if we increase the friction by putting rubber pads on the feet, the robot has a strong tendency to fall over. All Puppy's controller does is move the legs back and forth in a periodic manner. When the robot is put on the ground it will, after a few steps, settle into a natural running rhythm: the robot's interaction with the environment causes a particular gait pattern to emerge (see figure 4.2). For example, all four feet occasionally leave the ground together for a short period of time, causing the robot to exhibit alternating flight phases and stance phases.

In the Puppy experiment, the speed at which the robot runs cannot be varied arbitrarily, even though the speed of the motors can: within certain ranges, the robot moves erratically or even falls over, but within others, stable gaits emerge. A few observations about Puppy's behavior are in order here. First, the number of stable gaits for any given system is limited: a legged robot (or animal, for that matter) has certain preferred speeds corresponding to those gaits. Second, because the gaits are attractor states that the robot "falls into" based on its motor speeds, morphology, and environment, the robot will resettle into an attractor after it has been perturbed slightly. For example when the robot moves from smooth to rough terrain it may struggle a bit, but when it re-enters an environment with smooth terrain it will settle back into its original gait. However if the perturbation is too big, the robot will change behavior and settle into a new attractor: it may fall over and come to rest, or fall on its side and kick itself around in a circle (Mimicking the infamous stage antics of Angus Young, lead guitarist for the rock band AC/DC), or switch from running to walking. If the perturbation is not too large, the system will move back into the original attractor state, as we mentioned before. This region of states is called a *basin of attraction*. The important point here is that this falling back into a natural gait—or falling into a new one, for that matter—does not have to be controlled by a program running on the robot's microprocessor but arises naturally as a result of the usual suspects: the robot's morphology and environment. And third, related to this point, some gaits are more stable than others, i.e., they have a larger basin of attraction.

One of the big differences between a legged and a wheeled robot is that wheeled robots can typically move at any speed, and they can speed up and slow down continuously. In other words, there are no preferred patterns of motion or speeds that are clearly distinct from others, except perhaps for stopping. Legged robots and animals, by contrast, do have preferred speeds, corresponding to the different types of gaits: walking very quickly or jogging very slowly often feels uncomfortable for us, and we tend to want to slow down or speed up. Wheeled robots, like legged robots, can also have attractor states, but because of their simpler dynamics, the attractor states are less interesting and their number is much smaller. For example, a light-seeking Braitenberg-style vehicle moves toward a light source by performing a kind of "wiggling" behavior: the robot always turns in the direction of its most stimulated light sensor, which then causes the opposing sensor to face the light and become more stimulated, causing the robot to turn back, and so forth. This behavior might be called an attractor state of the robot. In any case, it will not have many of them. The point here is that all physical systems, because they are physical, will have attractor states, but those with complex morphologies have more (Kauffman, 1993). Therefore, although so far we have restricted ourselves to simple robots, in the future we want to work with more complex ones that have a large number of attractor states. It is important to have many, because attractor states may ultimately become the building blocks for cognition, as we will see in detail later on. For now, it is sufficient to think of the connection between attractor states and cognition by adapting an ancient metaphor: the wider you build the base (the more attractor states there are), the higher you can build your tower (the richer are the possibilities for combining attractor states). In the next chapter we will explore how attractor states can be used to form the basis of a kind of symbol-processing system.

To summarize the discussion so far, complete agents must comply with the laws of physics; they generate sensory stimulation when they act; they perform morphological computation—bodies can perform functions that would otherwise have to be performed by brains—and finally, complete agents are dynamical systems and their behaviors can be viewed as attractors. Also, because unlike formal systems, the real world is messy, so to speak, we cannot expect a clean, axiomatic theory or a set of principles that logically follow from one another. So the set of design principles that we will present is not a formal system, but a tightly interdependent set of design heuristics that on the one hand provide guidance on how to go about building agents, and on the other characterize the nature of intelligent systems. There is partial overlap and a certain level of redundancy among the principles, but this is not undesirable: they support one another because of this overlap. Moreover, all the design principles apply to all agents, to a greater or lesser degree. Finally, the individual principles should always be viewed in the context of the other principles: they form an interdependent set and should not be considered in isolation.

Let us now go through the agent design principles one by one.

4.3 Agent Design Principle 1: The Three-Constituents Principle

Designing an intelligent agent involves the following constituents: (1) definition of the ecological niche, (2) definition of the desired behaviors and tasks, and (3) design of the agent.

Intelligence, as we have said, is not a property of an agent, nor is it a "thing" that resides in a box inside an agent's brain, but rather it arises from the interactions of an agent with its physical and social environment. Thus, when designing an agent it is not sufficient to focus on the agent itself, but we also have to think about the ecological niche in which it is to function, as well as what the agent is supposed to do.

The three-constituents principle can be summarized as follows. Designing an intelligent agent involves the following constituents: (1) definition of the ecological niche, (2) definition of the desired behaviors and tasks, and (3) design of the agent itself. The first two constituents are often collectively referred to as the task environment. The ecological niche, in the case of robots, is always a physical and social environment: for entertainment robots the niche encompasses children's homes, including other people, the siblings, the parents, friends, and pets. In this chapter we focus on the physical aspects of the task environment, and in the next we consider the social aspect.

Design Stances

If we design a robot to entertain children, it will have to function in people's homes and should behave so that it achieves the desired goal: keeping kids amused over extended periods of time. Finding the kinds of properties and behaviors that the robot should have in order to achieve this goal has turned out to be a formidable challenge. Cute robots like Sony's AIBO (the Artificial Intelligence roBOt, which in Japanese also means something like "buddy"), Omron's NeCoRo (a cat robot covered with fur), or NEC's PaPeRo (Partner-type Personal Robot) that, to some extent, can respond to sentences uttered by a human partner are popular examples of this particular species of robot. More straightforward examples are robots for mowing lawns or assembling motorbikes on an assembly line in a factory: in these cases the ecological niche and the desired behaviors can be more clearly defined.

In the design of such robots, the ecological niche—people's homes, backyards, factory environments—and the desired behaviors and tasks are taken as given, and the agent is designed such that in its interaction with the environment, the desired behaviors emerge and the robot achieves its tasks. But there are two additional versions of the design task. The second alternative is to take a given robot, put it into an ecological niche, and observe what sorts of behaviors appear. And the third, given the robot and the desired behaviors, is to look for the niches in which it will in fact function properly. We will give examples of some "design stances" in this chapter and in chapter 9 when discussing business applications of the design principles.

Recall from our discussion about frame of reference and about Puppy's gaits that behavior always emerges from the agent-environment interaction and cannot be directly programmed into the robot. Therefore robot behaviors can only be indirectly designed: to use the term introduced in the last chapter, we have to design for emergence. If we want to make a robot walk, we have to account for adaptivity: it has to be able to deal with uneven ground, slopes, walking over loose material, walking while carrying something, and so on. It becomes impossible to preprogram all the different varieties of walking needed for the near-infinite variety of agent-environment interactions that the robot will encounter in the real world. More simply, if the walking movements are entirely preprogrammed, the robot will fall over whenever something unanticipated—something not programmed into the robot—arises. Indeed, many walking robots do fall over when they encounter uneven ground.

The relationship between an agent and its ecological niche is complex; so, let us briefly discuss some of the implications. First, the ecological niche of a robot is not simply the environments in which it can operate successfully: as in biology, there is always competition for resources. Entertainment robots have to compete not only with other entertainment robots, but also with toys, pets, and humans. Ultimately, the market will decide which (if any) entertainment robots get to share this niche with the occupants (toys, pets, and humans). If on the other hand we are interested in explaining the behavior of natural systems we can start from a particular set of behaviors that we observe, try to identify the ecological niche, and then ask how the behaviors come about. The orientation behavior of desert ants that we already discussed is a case in point. Their highly specialized sensors enable them to navigate over large areas in relatively featureless terrain. Recognizing the characteristics of their unique ecological niche—the desert—has helped biologists to better investigate and understand their behavior.

We can also turn the design problem around. If we already have an agent designed for a particular ecological niche, such as the AIBO robot designed for entertainment, we can drop it into a different ecological niche and ask what kinds of behavior will emerge. A company with a robot already on the market might look for additional ecological niches in which the robot will display its desirable behaviors and achieve its tasks, and thus widen its consumer base. For example, in addition to homes, AIBO might in fact also be useful in schools, thereby serving as an educational tool.

There is yet another way in which we can look at the design problem. Often engineers—the clever ones—design the agent and its ecological niche at the same time because in this way much better solutions can be achieved. The global positioning system or GPS is a great example of this idea. Putting satellites into the sky largely solves the navigation problem on Earth once and for all, at least outdoors; robots that need to orient can be made much simpler because they don't require sophisticated navigation strategies, but only a sensory system for tuning into the GPS signals!

Scaffolding

Scaffolding describes the way in which we, and other agents, structure our environments to simplify our tasks. In the GPS example, having many satellites in orbit makes the lives of robots—and of many car drivers—much easier. Another example is the use of road signs: if signage is done properly, the driver needs absolutely no geographical knowledge and can easily arrive at the target location by simply following the signs. Thus with adequate scaffolding, the mechanisms required for successful navigation will be very cheap, so to speak: there is no need to plan the route or consult a map. This exemplifies the principle of cheap design, which we will shortly discuss. Information and communication technology provides powerful scaffolding, leveraging our intellectual abilities far beyond those of our ancestors two thousand years ago, even though our brains have not grown in the meantime. Bioinformatics, which is the combination of new scientific instruments, database and networking technology, and pattern detection and modeling algorithms, has provided the "scaffold" which enabled the research community to sequence the human genome.

Aside from technology, language is another extremely potent means of scaffolding: because our knowledge can be written up in books, and thus communicated, we are now able to perform tasks that before the existence of written language would simply not have been possible. Now we can build on top of what has already been established and written down: the ideas in one text rely (directly or indirectly) on those in other texts, and so on. The World Wide Web, stuffed as it is with text, images, sound and video, has simply made this web of ideas more explicit and much more easily accessible. Natural language and information technology are among the most powerful scaffolding tools around, a point that is elaborated in the engaging book by the British philosopher Andy Clark, *Natural-Born Cyborgs*.

Recall how embodied agents always affect their environment when they act: as the "Swiss robots" make their clusters, they also make free space to move around in. But manipulating the environment to serve one's purposes can be found everywhere: we take notes, we write documents and books, we type things into computers, we use sticky-note pads, we store phone numbers in our mobile phones, we put information on bulletin boards, we take pictures and videos, and we put up Web pages. Given the obvious usefulness of changing the environment to simplify our lives—that is, of scaffolding our environment—it is truly surprising that most robots do not significantly change their environments to make their tasks easier! Thus, scaffolding is an important part of the three-constituents principle, because it requires consideration of the agent's niche, what tasks it is to perform, and how it should be designed.

4.4 Agent Design Principle 2: The Complete-Agent Principle

The complete agent principle states that when designing agents we must think about the complete agent behaving in the real world.

This principle contrasts with the paradigm of "divide and conquer" that pervades virtually all scientific disciplines: decompose a problem or system into simple subsystems which can then be developed separately. Once the subsystems have been designed, they can then be put together again. But it often turns out that in practice, subsystems create unnecessary problems, known as artifacts, which would not exist if the system were taken into account in its entirety. A good example of this comes from the field of computer vision, where it seemed obvious at the outset that vision could be understood as a separate process from the rest of the agent's behaviors. Computer vision thus focused almost exclusively on the analysis of static photographic pictures, such as desks cluttered with objects. Highly sophisticated and computationally intensive algorithms were developed to "understand" the images by identifying and categorizing the object in the image. However, vision turns out to be much easier when the agent interacts with the environment. In other words, we should treat vision as an interactive process, not just a set of operations performed on a set of static images. If you move your head back and forth, objects that move more quickly over your visual field are closer to you than objects that move less; if one object blocks your view of another object, you can simply walk to another location and look again. Simple. Having a body and being able to act in the world simplifies vision—and many other things as well, as we will see. This insight helps us when building agents, but it is also useful in trying to understand existing agents, like ourselves.

Here is another example drawn from the related research area of perception, in which researchers in the cognitive sciences, psychology, and neuroscience try to figure out how individuals can interpret sensory stimulation in the real world. It has been demonstrated in many experiments that the function of a particular part of the brain can be very different depending on whether the agent—typically an animal—is studied as it behaves in the real world, or the particular subsystem, in this case the vision system, is studied in isolation. In what would eventually lead to a Nobel Prize in 1981, the neuroscientists David Hubel and Torsten Wiesel conducted a famous experiment in the late 1950s in which they inserted a microelectrode into individual cells in the visual cortex of an anesthetized cat. They then presented the immobilized animal with different kinds of visual stimuli while recording the signals from these cells. One of their fascinating results was that some cells did not respond to light intensity but rather to orientation of edges. In other words, some of these cells would respond only if the left of the visual scene was light and the right was dark (or vice versa), while others would respond only if the top was light and the bottom dark, and so on. It seemed natural to conclude from this that some neurons in the cat's visual cortex act as edge detectors. Later, when experiments with moving cats became technologically possible, it was found that these cells were in fact involved in many other activities as well (Haenny et al., 1988). Although it is correct to say that there is a correlation of the activity of these so-called edge-detection cells and the presentation of the visual stimuli containing the edges, it cannot be said that they are edge-detection modules, because they are involved in other behaviors as well. We are not criticizing Hubel and Wiesel's groundbreaking experiments but merely pointing out that these neurons cannot be considered basic modules from which the complete system could be assembled. The results still hold: they only need to be reinterpreted.

Often, it turns out that viewing only part of an agent when explaining its behavior causes us to attribute more "brainpower" to it than may actually be there. In other words, by considering the entire agent we can often find other, simpler mechanisms for achieving the behavior. So the complete agent principle is related to the principle of cheap design that we will discuss next: given the right body for the job, and keeping the agent's behavior and environment in mind, agents can get away with less computational hardware. Remember the navigation behavior of the desert ant *Cataglyphis*? It has been shown in many experiments that the ant can use landmarks to find the precise location of the nest when it returns from a foraging trip. In these experiments, the landmarks are typically large black cylinders that are placed around the nest. In order for the landmarks to be useful, the ant has to recognize them first, then make a decision in which direction to move; at least that is what we would think should happen. Recognizing landmarks is a difficult task that would require a perceptual system potentially entailing a lot of computation, as we explained in the computer vision example. However, as described in chapter 2, the ants take a kind of "snapshot" of the surroundings as they leave the nest. When they come back near the nest, they simply compare the stored snapshot with what they currently see—the current sensory stimulation—and they move in the direction that will further reduce the difference between the two. When the two fully match, the ant is precisely at the location of the nest. At this point, we can say that the ant has recognized the landmarks, but the "recognition" is fully integrated into the behavior of the ant, and we cannot separate "finding the nest" from "recognizing the landmarks." This implies on the one hand that there are not two separate modules for these tasks, and on the other that by looking at the behavior of the complete agent, rather than at the perceptual subtask only, we can see that the solution is much "cheaper," from the perspective of the agent's design (for more details see Lambrinos et al., 2000).

Furthermore, when observing complete agents behaving in the real world, we are less prone to modularize our systems inappropriately: in the previous example, two incorrect modules that we could have proposed to explain the ants' behavior are "find the nest" and "recognize landmarks." Psychology has as its research topic the most complex known system in the universe, the human. In order to come to grips with the awesome complexities involved, researchers in this field carve up the human psyche in particular ways for the purpose of investigation, for example into cognition, perception, categorization, memory, attention, social interaction, learning, development, motivation and emotion, motor action, problem solving and reasoning, planning, creativity, communication, language, awareness, and consciousness, to mention but a few. Separate fields within psychology are devoted to the study of many of these areas. If we look at the complete agent and ask what processes underlie behaviors such as walking, talking, or recognizing a face in a crowd, we see immediately that these subdisciplines do not so much correspond to actual "modules" but are in fact different perspectives on the same (or at least largely overlapping) set of processes. For example, learning makes no sense without perception, and memory makes no sense without learning. Planning can only be performed on the basis of perception and memory, and so on.

Moreover, when studying complete agents, we always have to deal with complete sensory-motor loops. If we follow this principle we will never be in danger of decoupling certain aspects—such as the planning of movements—from the sensory system, as is usually done in classical robotics. In 1999, I (Rolf) was a guest in a research laboratory of a large car manufacturer in Germany, where, for the first time in my life, I was served coffee by a robot. It was a great experience: the robot went over to a table, grabbed a cup, moved over to the coffee machine, deposited the cup, pushed the button, waited for the cup to be filled, moved over to my chair, and handed me the cup. I was impressed. But in fact it was not actually as smooth as all that: motion planning had been developed separately from the rest of the agent, which led to a few problems. For example, while performing the planned movement, at the time (I am sure this has changed meanwhile) the robot received no sensory feedback from the environment. As a result, the robot grasped the cup in a slightly different way from how it was supposed to, causing the cup to bend and almost break the tube where the coffee came out of the machine. In a complete-agent approach, one is forced to always take the complete sensory-motor loops into account: if the robot had been able to sense the way it grasped the cup or the strain the cup was placing on the coffee dispenser, this particular problem would have been avoided. This also illustrates the principle of sensory-motor coordination, described below.

In summary, the complete agent principle has important implications both for how we study agents, as in psychology and neuroscience, and for how we design and build them, as in robotics. This principle also emphasizes that in a complete agent, everything is tightly interconnected.

4.5 Agent Design Principle 3: Cheap Design

The principle of cheap design states that if agents are built to exploit the properties of the ecological niche and the characteristics of the interaction with the environment, their design and construction will be much easier, or "cheaper."

Recall for a moment our discussion of what we intuitively mean by intelligence. We suggested that the concept is related to compliance and diversity. Agents that comply with and exploit their ecological niche in order to generate diverse behavior are intuitively considered more intelligent. We have to comply with the givens: there is no way in which we can ignore the fact that there is gravity on Earth; or rather, to ignore it will generally not be very beneficial to the organism. If I step off a roof-terrace, I will fall, whether I like it or not. But this is not always a negative thing: the laws of physics can also be exploited in smart ways. It is worth distinguishing here between two closely related aspects of exploiting the ecological niche: the properties of the niche, which includes the laws of physics, gravity, friction, electromagnetic forces; and the properties of the interaction with the environment, such as the sensory stimulation generated as an agent moves. The principle of cheap design simply states that if agents are built to exploit these kinds of properties, their design and construction will be much easier. So cheap design is about exploitation of the properties of a niche, and the term *cheap* should not be taken too literally. However, it is indeed often the case that if the principle is applied properly, the resulting agents will be cheap in the literal sense of the word: if they are simple, they will be inexpensive to design, manufacture, and maintain. The related design principle of ecological balance (described below) helps us to figure out *how* this exploitation should be done; cheap design simply illustrates *that* the more and better the exploitation, the simpler agent it will be.

We are now going to illustrate these points with a few examples: the Swiss robots that we have already introduced; the "passive dynamic walker," a "brainless" and nonmotorized robot capable of walking down a slope without control, and its successor, "Denise," which has a little bit of brain mass and some actuation; and finally we will look at how insects can coordinate their legs when they are walking even though there is no center in their brain that actually manages the synchronization of the movements. We introduce these examples in the context of the cheap design principle, but it should be kept in mind that all of the design principles apply to all agents to a greater or lesser extent.

The Swiss Robots

Recall the case study of the Swiss robots that we introduced in the previous chapter where the task was to design robots that together tidy up an arena cluttered with Styrofoam cubes. (This is admittedly not the most glamorous of tasks, but it is definitely related to one of the reasons we want robots in the first place!) Intuitively we would think that the following steps have to be taken. First, the robot has to find a cube. Once it has found one it has to look for the nearest heap or cluster. Then it has to move and deposit the cube there, and the procedure is repeated until all the cubes are clustered. These steps all require sophisticated visual processing and planning, and would thus be computationally expensive, so to speak. Remember that just because visual perception is natural and effortless for us humans by no means implies that it is a simple process.

The Swiss robots take an alternative approach: they master the job by exploiting their own morphology and the properties of the ecological niche. Remember that in fact they achieve the task by being programmed only with simple reflexes for obstacle avoidance. In order for the clustering to come about, the following aspects of the ecological niche had to be exploited: the size of the cubes (if they are too big or too small it does not work), their weight (if cubes were too heavy for pushing, it would no longer work), the fact that the environment is enclosed by surrounding walls (otherwise, the robots would drive away, rather than cleaning up), the fact that the ground is flat and provides, together with the tires of the wheels, the right kind of friction (if you put soap on the ground, it will no longer work). If any of these constraints do not hold, the Swiss robots will miserably fail to achieve their task. But if they are fulfilled, this solution works very well, and it is cheap: the Swiss robots exploit the properties of their niche, the laws of physics, and their own morphology in clever ways, so that computationally expensive vision is not required. The Swiss robots do not need to know what they are doing; they merely react to sensory stimulation.

The Passive Dynamic Walker and "Denise"

The passive dynamic walker, illustrated in figure 4.3b, is a type of robot (or, more accurately, a mechanical device, since it has no sensors or motors and no control program) that was first proposed by McGeer (1990). It is capable of walking down a ramp without any sensing, actuation, or control: in other words, it is literally brainless, if you like. In this sense, it is not really an agent. Nonetheless, in order for this task to be achieved, the dynamics of the robot—how gravity, friction, and the forces generated by the swinging of the legs and arms act on it—must be exploited. The result of this exploitation is that the walking behavior is very energy efficient and looks surprisingly natural.

However, its ecological niche, i.e., the environment in which the robot is capable of operating, is extremely narrow: it consists only of slopes of certain angles. Just as in the case of the Swiss robots, if you change anything whatsoever in the ecological niche, such as the angle of inclination or the surface properties (e.g., by putting a soft rug on it), the device will no longer work. The fact that an agent will cease to function if some aspect of its niche (specifically some aspect that the agent exploits) is

(a)

(b)

Figure 4.3
Passive dynamic walkers. (*a*) The patron saints of the city of Zurich in Switzerland: Felix, Regula, and Exuperantius. They were beheaded in the third century because of their religious beliefs. The legend says that they carried their heads under their arms to a spot where later the Grossmünster church, the symbol of Zurich, was built. Legend? No, passive dynamic walkers! (*b*) The "classical" passive dynamic walker by Steve Collins that walks down a declined ramp with no actuation (left), together with the 3D biped robot "Denise" by Martijn Wisse (right). Denise is a hybrid passive dynamic walker: its ankle and knee joints swing passively, while a motor drives the hips to induce walking over flat ground.

changed is an unavoidable trade-off of the principle of cheap design. Energy efficiency is achieved because the leg movements are entirely passive, driven only by gravity in a pendulum-like manner. To make this work, a lot of attention was devoted to morphology and materials. For example, the robot is equipped with wide feet of a particular shape, elastic heels, and counterswinging arms that all help it to walk in this way (Collins et al., 2001).

Loosely speaking, we can also say that the neural processing normally required for controlling walking is taken over by the proper morphology and the right materials, and thus is another instance of morphological computation. In fact, the neural control for this robot reduces to zero. But, if anything is changed, e.g., the angle of the incline, the agent ceases to function—the price of cheap design.

Because the fully passive dynamic walker exploits many properties of its ecological niche, it is entirely dependent on that niche. But the ecological niche can be widened if we augment the agent's capabilities: by adding motors, adding some control, and modifying the morphology of a passive dynamic walker we enable the robot to walk over flat terrain. This has been achieved by the team led by Martijn Wisse, a highly creative young engineer at the Technical University of Delft, in Holland, who was also involved in the development of the passive dynamic walker at Cornell University. He recently created "Denise" (figure 4.3b), an almost completely passive dynamic walker, by augmenting the earlier model with some actuation, adding two electrical motors to move the legs. Its walking behavior (or should we say "her" walking behavior?) is actually quite natural, presumably because it exploits the passive forward swing of the leg.

One might be inclined to say that cheap design only works for very simple systems, and admittedly the examples we have given so far are all indeed very simple. But look at humans for a moment. When we walk, the forward swing of our legs is—like "Denise's"—mostly passive, i.e., the muscles are passive and the leg swings forward like a pendulum, thereby exploiting gravity. Our legs are complex indeed, with their bones, joints, tendons, ligaments, muscles, nerve cells, and skin, but complexity does not preclude exploitation. In this sense we can say that we ourselves as humans, even though we are incredibly complex, are "cheaply" designed. It will certainly be interesting to see whether Wisse's approach to robot walking will scale up to even more complex systems, in particular complex humanoid robots, or whether alternative approaches will have to be employed.

Even though the passive dynamic walker is an artificial system (and a very simple one at that), it has a very natural feel. The term "natural" applies not only to biological systems but to artificial ones as well: perhaps the natural feel comes from the exploitation of the dynamics, e.g., the passive swing of the leg (for an elaboration on this point see Pfeifer and Glatzeder, 2004).

Leg Coordination in Insect Walking

The first two examples were drawn from robotics, so let us now look at one from biology: leg coordination in insect walking. It has been known for a long time that leg movements in insects are controlled by largely independent controllers (von Holst, 1943), in other words, there seems to be no center in the brain that coordinates the legs in walking. But if there is no such coordination center, how then can insects walk in the first place, and how does leg coordination come about? And the legs do need to be coordinated, otherwise walking is not possible. A couple of years ago the radical thinker and trendsetting German biologist Holk Cruse, who has been studying insect walking for many years, cracked the conundrum. It turns out that the trick these insects use is to exploit their interaction with the environment (Cruse, 1990). Assume that the insect stands on the ground and then, in order to move forward, pushes backward with one of its legs. As a result, the joint angles of all the legs standing on the ground will instantaneously be changed. The body is pushed forward, and consequently the other legs are also pulled forward and the joints will be bent or stretched. This is one of those unavoidable repercussions of being an embodied agent, and the insect can do nothing about it. However, and this is Cruse's fascinating finding, this fact can be exploited to the animal's advantage. All that is needed is angle sensors in the joints—and they do exist—for measuring the change, and there is global communication between the legs! But the communication is through the interaction with the environment, not through neural connections.

So, the local neural leg controllers need only exploit this global communication. There is an additional benefit of all this. Because the insect is moving forward, the angles of the other legs are all moving in the right direction—information that, in addition to being free, i.e., available without computation—is extremely useful and can be directly exploited for controlling the individual legs. This is not trivial, but Cruse and his colleagues have worked out a neural network architecture that does the job (Dürr et al., 2003). And this architecture, the WalkNet, can also be used to control six-legged robots.

This is another beautiful instance of cheap design: if the insect had to do everything through computation, it would be more costly and much slower. This is also an instance of morphological computation: part of the task that would have to be done by the brain—the communication between the legs and the calculation of the angles on all the joints—is performed by the interaction with the real world.

The principle of cheap design is very general because it only states that the ecological niche can be exploited to simplify the agent, but does not tell us *how* the exploitation should be accomplished or what dynamics should be exploited. Other design principles such as ecological balance, redundancy, and sensory-motor coordination are more specific and more about the *how*. But cheap design can be applied to more specific issues, such as the design of the visual system—a field that is becoming known as "cheap vision." The literature on vision is full of examples of how an ecological niche and specific interactions with the environment can be exploited. An instructive and entertaining example, the "Eyebot" robot, is discussed later in this chapter.

To conclude the discussion of cheap design let us briefly mention some examples that do not conform to this principle, in order to clarify it a bit. A laptop computer, as explained before, does not exploit the environment in interesting ways, and neither does a humanoid robot in which the movements required for walking are largely "programmed into" the robot. Famous humanoids like Asimo, Qrio, or HRP (from the Japanese Humanoid Robotics Program) are largely preprogrammed, and there is no substantial exploitation of their system-environment interaction (yet).

4.6 Agent Design Principle 4: Redundancy

The redundancy principle states that intelligent agents must be designed in such a way that (a) their different subsystems function on the basis of different physical processes, and (b) there is partial overlap of functionality between the different subsystems.

The redundancy principle is geared toward designing robust systems, i.e., systems that continue to function even if there are significant changes in the environment. The term *redundancy* has a long history and is used in many different ways, and so, once again, rather than trying to come up with a definition we introduce the term intuitively using a number of examples.

Visual and Haptic Systems in Humans

The term *modality* is often used in the literature to designate different sensory channels: we talk about the visual, the haptic, or the auditory modality. The visual system, or visual modality, provides us with precise spatial information that enables us to move around very quickly because we can see where to go and where obstacles and desired objects are. Because this visual information is so extremely valuable, many species have evolved, one way or another, visual systems. But what if it suddenly gets dark? Vision, alas, only works in the presence of light. But all is not lost: we can resort to other sensory modalities; we can still hear and feel. Although we can extract some spatial information from our auditory system—we can roughly hear where a sound is coming from—it is much less precise than what we get from the visual system. But from our sense of touch—also called the haptic system—we can get very precise spatial information: we can feel with our hands and fingers, and it is relatively easy to identify an object. Moreover, we often consider touch to be more reliable than what we observe with our eyes: sometimes we have to touch things because we do not fully trust what we see. One of the authors of this book (Josh) learned this lesson the hard way: when a guest at a party, it is best to reach out with your hand when crossing from the house into the backyard so as not to blunder headfirst through a hard-to-see screen door and thereby turn yourself into the focus of the party.

While touch is good at short distances, it is not very efficient in the long range. For walking around, it can be used as long as we go slowly, because unlike the visual system it requires physical contact. All this is common sense, of course, but the essential point is that even if we have to slow down, we can still function because we can rely on a different set of sensors appropriate to the new situation. The reason this works is that the two systems are based on different physical processes: the visual one on stimulation by electromagnetic waves, and the haptic one on mechanical pressure. Nevertheless, the two modalities yield partially overlapping information: the information extracted from one can be used to—at least partially—predict the information that can be extracted from the other. If you see a glass of beer on the table with condensation on the outside you already know more or less what it will feel like when you touch it. The information contained in both sensory channels is technically referred to as mutual information and plays an important role in building concepts: the concept of a glass of beer e.g., includes not only information extracted from the visual system but information from the haptic and the taste systems as well.

The Meaning of Redundancy

Because the information extracted from one sensory system includes some information that can also be extracted from another, this phenomenon is called redundancy. The term *redundancy* is actually a tricky one because it depends very much on the point of view we are adopting and it has many different interpretations. Sometimes redundancy is taken to mean duplication of components, or the part of a message that can be deleted without essential loss of information.

Natural language is a good example of this latter interpretation of the word. We can understand others even if there is a lot of noise in the environment and we physically hear only part of the message, or when the sentences are grammatically wrong and some words are missing. If we ask someone how they are and they reply with either "I'm feeling better, thanks," or simply a mumbled "better," the same message is conveyed. Granted, we often repeat what we say if we think the listener did not understand us, but more often than not we tend to say the same thing in different ways—or support our ideas with a bunch of examples—in order to make sure we get our message across. In fact this entire book is filled with different versions of the same message: intelligence requires a body.

In general, biological systems are extremely redundant because redundancy makes them more adaptive: if one part or process fails, another, similar part or process can take over. Brains also contain a lot of redundancy; they continue to function even if parts are destroyed—which should come as comfort to many of us since we know that alcohol has a tendency to destroy brain tissue. So, it sounds like redundancy is a good thing. Note, however, that redundancy also has its price. Additional parts have to be genetically represented (one way or other), they consume energy, they have weight, they take up space on the organism, etc. In short, adaptivity has to be paid for: there is no free lunch.

In engineering, redundancy often means duplication of components. In an airplane, instead of having one navigation system, there are two. But duplication on its own is not very interesting. If you have, say, two eyes instead of one, or even if you have a thousand, this is not very helpful if it gets dark. However, if you then have a touch system and an acoustic system, which are independent of whether there is light or not, you can still function. Interesting redundancy is also found in aircraft engineering. The braking system consists of two or three parts: the wheels, the jets, and sometimes, in high-speed aircraft, the parachutes. If there is ice on the runway, wheels are not very efficient, but then the jets can be used because their functioning does not depend on the condition of the runway. If the electrical system of the airplane ceases to function, the parachutes, which are purely mechanical, will still work. Wheels are used not only for braking but for maneuvering on the ground in general, jets are also—in fact, mostly—used for propulsion, whereas the parachutes are used only in emergency situations for braking; they usually do not cost much in terms of weight and manufacturing expense, but might come in handy.

What we can learn from these examples is that we should design agents that must function reliably under many different conditions with redundancy in such a way that there is partial overlap of functionality. If the overlap were complete, the two systems would be doing the same thing, which is not very economical and—normally—not terribly interesting in terms of adaptivity.[2] Another way of viewing partial overlap of functionality is that the same task can be achieved in different ways: braking can be done by using the wheels or the jets; recognizing an object can be achieved by looking at it or by touching it.

Robot Whisker Systems: The Artificial Mouse

Let us now look at an example from robotics: the "Artificial Mouse" developed at the Artificial Intelligence Laboratory at the University of Zurich by the engineer Hiroshi Yokoi, the neuroscientist Miriam Fend, and the theoretical physicist Simon Bovet (Fend et al., 2002). Rats and mice have sophisticated whisker systems that they can employ to acquire all kinds of information about the world. They can be used to detect the distance to an object (if the object is within reach of the whiskers), surface texture, and vibrations. Often water in the jungle is too muddy to see through: cats can solve the problem of hunting fish by dipping their whiskers into the water so that, through the vibrations produced by the movement of the fish, they can with uncanny precision locate and catch them. Rats and mice perform active whisking, i.e., they not only passively sense the environment with their whiskers as they move past objects, they also have muscles that enable them to move their whiskers back and forth. This ability has been built into the Artificial Mouse as well.

If a whisker from a real rat is attached to a microphone which in turn is connected to an amplifier, and the whisker is moved over different surfaces such as plastic, glass, wood, fabric, or sandpaper, one can, by merely listening to the sound produced, easily discriminate the different textures. The goal of the Artificial Mouse project is to study the use of the whisker system, in particular how the information from two morphologically very different sensor modalities, such as the visual and the whisker system, can be exploited by an animal or a robot to solve a problem, such as finding its way through a maze in which the walls have different textures. If there is a partial overlap in the kind of information that can be extracted by two sensor systems, it may be possible that over time, information that at first has to be extracted from the whisker system can—at least partially—be acquired through vision, which would presumably enable the rat or mouse to move around faster than if it had to test everything with its whiskers first. At least in the Artificial Mouse, this is definitely the case. This idea, known as cross-modal learning, will be further explored in the context of the principle of sensory-motor coordination and development in the next chapter.

At first sight, the redundancy principle might seem to contradict the principle of cheap design because the former calls for additional subsystems, whereas the latter calls for more simplicity. However, the two principles are complementary: even a highly redundant system like a human being can exploit, for example, passive dynamics. The systems can also work together. For example, discriminating textures by vision alone might require a lot of computation, whereas combining vision with a touch sensor—whisker or skin—might make the task much simpler. This is closely related to the fact that through a particular type of interaction—sensory-motor coordination—one sensory modality can help structure the stimulation in others, an idea which is covered by the principle of sensory-motor coordination.

4.7 Agent Design Principle 5: Sensory-Motor Coordination

The principle of sensory-motor coordination states that through sensory-motor coordination structured sensory stimulation is induced.

As we explained at the start of this chapter, one of the important properties of embodied agents is that as they move through their environment, they automatically generate sensory stimulation: they cannot help it. When discussing the principle of cheap design we explained how this sensory stimulation can be exploited for a particular purpose, as in animals that exploit the signals from their angle sensors to coordinate leg movements for locomotion. Another way of saying this would be that the animal lifts its leg not only for walking, but in order to generate sensory stimulation. And this is precisely the idea of sensory-motor coordination: embodied agents can generate useful sensory stimulation by interacting with the environment in particular ways.

The fact of the matter is that perception is really hard. Remember that the real world is no clean eight-by-eight chessboard: it is a hectic, noisy place. Imagine an agent such as yourself walking through Bahnhofstrasse, the posh shopping street in the center of the Swiss city of Zurich. The sensory stimulation happening at the retinas of your eyes is continuously and rapidly changing, on the one hand because other people, the trams, and the cars move, but on the other because you yourself move. What also happens when you move is that the distance between you and the other objects in the environment changes, but also your relative orientation to them varies: sometimes we see people from the front, then from the side, then partly from behind; or they are partly hidden by other people or objects. Moreover, the lighting conditions change, we walk into a department store, we put on or take off sunglasses, it begins to rain or it gets dark in the evening. Surprisingly, in spite of all this variation, we have no problem recognizing—in no time flat—a friend, a shop, or a bar of Toblerone lying on a pile behind a bunch of people.

So, the variation in the sensory stimulation is, in a way, the bad news: how can we ever build robots that can handle all this change? The good news is that through the interaction with the real world, this sensory stimulation can be simplified so that it is easier to make sense of it. This holds in particular for sensory-motor-coordinated interactions: shaking your head around randomly—not sensory-motor coordinated—generates a lot of stimulation on your retinas and in your inner ear (which senses your body's orientation relative to gravity), but that stimulation is probably not very useful. We will soon say more about what we mean by sensory-motor-coordinated interactions, but for now it is enough to think of them as interactions where the sensory stimulation influences the action and the action in turn influences the sensory stimulation. A very simple example of sensory-motor coordination is looking at an object. Foveation is the technical term for this, i.e., moving the head and the eyes in such a way that the object appears in the fovea, the high-resolution center of the retina. This is a process of sensory-motor coordination because the movement induces sensory stimulation and this sensory stimulation in turn influences the movement—compensating head and eye motions—so that the object remains at the center of the visual field.

It is important to point out here that sensory-motor coordination is always performed with respect to a particular goal or intention. If I walk past the table on which a coffee cup is standing, without specifically looking at the cup, then my behavior is not sensory-motor coordinated with respect to the cup. But it is sensory-motor coordinated with respect to walking, because in order to walk properly I have to react to the sensory stimulation that I receive from the touch sensors of my feet, from the force sensors in the muscles and tendons, and from the inner ear, which helps me keep my balance.

Inducing Correlations

Sensory-motor coordination turns out to be especially useful because it induces correlations within a sensory channel and between sensory channels. When I look at the coffee cup on my desk by foveating on it—when I center it in my visual field—the image on my retina is, at least for a short period of time, stabilized and the resulting sensory signals can thus be more easily processed by the visual system. When I then grasp the cup, I also induce sensory stimulation in other sensory channels, such as the touch sensors on my fingertips and the proprioceptive sensors in my arm (the sensors that measure internal stimulation such as force on muscles or tendons). Through sensory-motor coordination, signals from the different sensory modalities become correlated: when I grasp and lift the cup, there is simultaneous stimulation of the touch and proprioceptive sensors in my hand and arm, and of the visual system. And because these signals are correlated they can be more easily processed: instead of a mass of complex, independent signals, there is a synchronized set of signals from which useful information can more easily be extracted. But, most important, these correlations allow learning to take place: associations between the different modalities can be formed. As simple as they may sound, we believe that these ideas will in fact help us make inroads toward clarifying the mystery of perception. The real beauty of sensory-motor coordination is that it shows not only *that*, but *how* embodiment affects the incoming sensory signals, and thus suggests what processing needs to be done by the brain: when my touch sensors tell me I have grasped a cup and I see that it's full, I must prepare to support its weight when I lift it, because I "know" that my proprioceptive sensors will soon fire, indicating that the cup is heavy. Put differently, sensory-motor coordination shows how body and information are connected. And this is one of the deep implications of embodiment. For a more in-depth exploration of these ideas see Pfeifer and Scheier (1999); for a psychological perspective see O'Regan and Noë (2001).

So far the idea that the agent's body induces correlations seems very plausible, but admittedly it is also qualitative and intuitive. For scientific purposes we have to "prove" this idea to be the case, i.e., we have to support our intuitions with scientific evidence. In other words, we have to be able to demonstrate quantitatively, using statistical or information-theoretic methods, whether that is true. The young and innovative Italian computer scientist and engineer Max Lungarella from the University of Tokyo, the Dutch ethologist Rene te Boekhorst of the University of Hertfordshire, the American neuroscientist and "neuroboticist" Olaf Sporns of Indiana University, and I (Rolf) have shown that through sensory-motor coordination, correlations are induced in the sensory stimulation and these correlations provide the basis for perception and learning.

The idea of sensory-motor coordination is in fact very old. We borrowed the term from the American philosopher and psychologist John Dewey, who introduced it in his famous and provocative article "The Reflex Arc in Psychology," published in 1896. (Note that the concept of sensory-motor coordination also plays an important role in Jean Piaget's theory of intelligence development, where it is used to characterize a particular stage; Piaget, 1952). Dewey argued that perception should not be seen as a process that starts from sensory stimulation, passes through internal processing, and finally produces an action: this is the classical behavioral view of input-processing-output. Rather, he suggested that "we begin not with a sensory stimulus, but with a sensorimotor co-ordination. . . . In a certain sense it is the movement which is primary, and the sensation which is secondary, the movement of the body, head, and eye muscles determining the quality of what is experienced" (Dewey, 1896; reprinted in McDermott, 1981, pp. 127–128). In fact, we would not argue that the movement is primary but that, to use once again the dynamical systems metaphor, both sensory and motor processes are coupled—they depend on each other. Trying to identify which is primary and which is secondary would be like attempting to solve the chicken-and-egg problem. We speculate that what Dewey did not know at the time was *why* sensory-motor coordination is so fundamental: we suggest that in addition to mastering the manipulation of objects, there are significant information-theoretic implications, as we just discussed.

We already pointed out that categorization is one of the most fundamental of cognitive abilities. Perceptual categorization, as well as perception in general, in animals and humans has all the characteristics of sensory-motor coordination, and once we consider different activities—looking, grasping, drinking, walking, writing, and listening—more carefully, we realize that we are always engaged in sensory-motor coordination. It is one of the most important processes in the development from infant to adult, and it constitutes the basis of many forms of learning. We will come back to both of these points in the next chapter, where we sketch out how cognition might emerge from a developmental process in a bottom-up fashion.

Recognizing Objects by Manipulating the Environment

To conclude our discussion of sensory-motor coordination for the time being, let us look at an example from robotics. The Italian engineer and computer scientist Giorgio Metta, while working at MIT's Artificial Intelligence Laboratory on the humanoid robot Cog, was interested in getting the robot to recognize objects in its environment. Cog, developed in Rodney Brooks's laboratory during the 1990s, consists of a legless torso, a head with vision and auditory systems, and two arms with hands. Often, as we know from our discussion about computer vision, recognizing objects is a hard task, especially if there are many objects heaped together in a cluttered environment. The objects may be colored similarly to the background, the lighting conditions may not be very good, or contrast may be poor. One strategy, often used by humans, is to move the head while looking at an object. Through these sensory-motor-coordinated movements, sensory stimulation is induced that can be exploited to extract information from the environment. This strategy of moving your head and eyes—or the robot's "head" and cameras—to support perceptual processes is also applied in so-called active vision systems.

To take things one step further, Metta programmed Cog with a particular sensory-motor strategy that goes beyond mere head or eye movement: Cog was programmed to actually manipulate the environment by poking objects in front of it in order to see how they move. This is done by tracking only the movement of the robot's arm (which is easy to do and requires relatively little computation), i.e., all the robot can "see" through its own vision system is the motion of its arm and hand. By having a motion-detection algorithm, the robot continuously sees its hand and arm moving through space. If as the robot moves there is a sudden spread of motion activity in its field of vision, this is an indication that the robot is pushing an object, because the stationary object has begun to move as well. *Spread of motion activity* means that the area in which the robot detects motion suddenly becomes much larger than that described by the arm alone. This can be very easily detected, and the robot "knows" immediately what portion of the visual field constitutes the object (see figure 4.4, which shows a similar robot, the Babybot, that Metta had developed for his Ph.D. thesis). Through the interaction with the environment—by poking an object—the robot has induced sensory stimulation that distinguishes the moving arm and the moving object

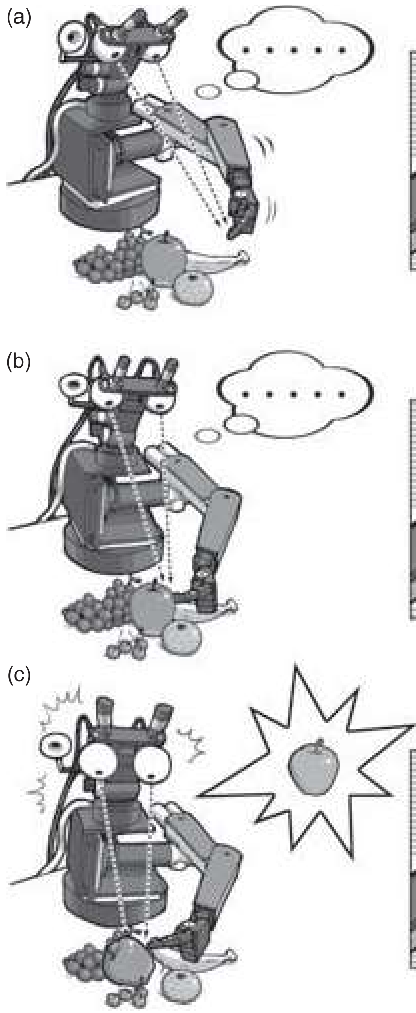

Figure 4.4
Generation of sensory stimulation through the interaction with the real world: sensory-motor coordination. Lira Lab's "Babybot" (University of Genova, Italy), exploring the cluttered area in front of it. The panels show the output from the motion-detection system. (*a*) The arm is clearly visible because it is moving, whereas the apple and the other objects on the table are at this point in time not visible by motion detection. (*b*) Babybot touches the apple, but the apple is still not moving and thus invisible to the motion-detection system. (*c*) Babybot pushes the apple, thus inducing motion so that the apple becomes visible and can easily be identified as an object. Reaching toward the apple, touching it, and pushing it are processes of sensory-motor coordination. (Experiments by Giorgio Metta and Paul Fitzpatrick.)

from the rest of the environment. This is a beautiful illustration of the principle of sensory-motor coordination.

When discussing design for emergence we said that if we could demonstrate that an agent designed by artificial evolution conformed to one of the design principles, this would provide additional support for the validity of that principle. As we will see in chapter 6, the "Block Pushers"—agents produced by artificial evolution—also exploit sensory-motor coordination in order to move, even though this coordination was not programmed into the system.

4.8 Agent Design Principle 6: Ecological Balance

The principle of ecological balance has two parts. The first states that given a certain task environment, there has to be a match between the complexities of the agent's sensory, motor, and neural systems. The second aspect is closely related to the first; it states there is a certain balance or task distribution between morphology, materials, control, and environment.

Let us briefly inspect the first aspect of ecological balance, the idea that, given a certain task environment, there has to be a match between the complexity of the agent's sensory systems, motor systems, and neural substrate. A nice illustration of this principle is given by Richard Dawkins in his book *Climbing Mount Improbable*, where he describes a hypothetical snail with human-like, and human-sized, eyes. This snail would have a hard time carrying along these giant eyes, but more importantly, they would be only moderately useful, if at all: human eyes, and the eyes of mammals in general, are adapted to our particular mode of life, which requires the detection of fast-moving objects, high-resolution images, and so on. A snail has little use for such abilities: why bother detecting fast-moving predators if you cannot run away from them, or detecting running prey if you are vegetarian? The complexity, weight, and size of the human eyes would only constitute unnecessary baggage, an example of an entirely unbalanced system.

Let us look at another, very different example. Recall the Braitenberg vehicles we introduced in chapter 2, in which the light sensors are directly wired to the motors in such a way that they would follow a light. The "brains" of these robots are extremely simple, consisting of only two wires, or "synapses" if you like, but they are sufficient for the purpose of light-following or light-avoiding. If you now replace the two-synapse brain by a brain of human complexity with 10^{14} synapses, how does the agent benefit from such a brain? The answer is simply that there is no benefit because the system is not ecologically balanced. Depending on the tasks required of the robot, the complexity of the sensory and motor systems would have to be augmented as well if the brain were to be useful. A Khepera robot with only two motors that can turn either forward or backward is, like Dawkins's snail, rather limited in what it can accomplish: equipping it with a high-resolution camera, for instance, is ecologically unbalanced, because it does not expand the behavior capabilities of the robot; it only weighs it down.

Because biological agents—animals and humans—have evolved, they are all ecologically balanced vis-à-vis their ecological niche. Humans, for example, have enormous brains, but they also possess, taken as a whole, the most sophisticated sensory and motor systems of any species on Earth. Some animals admittedly possess amazingly dexterous appendages, such as an elephant's trunk or the tentacles of an octopus, or impressive sensory organs—for example, the bat's echolocation scheme for catching flying insects—but consider the flexibility of the human hand or the astounding intricacy of our vocal tract. A complex hand allows good tool use; a complex vocal tract allows for language. A heavy-duty brain indeed is required for coordinating our complex sensory and motor systems in order to carry out a wide range of tasks. Again, it is worth pointing out that one system did not evolve ahead of the others; rather, they increased in complexity roughly together.

The second aspect of the principle of ecological balance is basically a generalization of the first aspect: given a particular task environment, there must be a certain balance or task distribution between morphology, materials, control, and environment. This second part has been elaborated in great detail in many papers (e.g., Bongard and Pfeifer, 2001; Hara and Pfeifer, 2000; Ishiguro and Kawakatsu, 2003; Pfeifer, 2000; Pfeifer et al., 2004), so here we only provide a few example systems to build an intuition of what this balance is all about.

The Human Hand-Arm-Shoulder System

Although the situation is slowly changing, most robots are still built from hard materials like aluminum and plastics, and for actuation they rely on electrical motors. It turns out that the control programs for such systems tend to be very complicated because every little movement of every joint, down to the fingertips (if the robot has any) has to be explicitly controlled (as illustrated in figure 4.1c). By contrast, in the human hand-arm-shoulder system the muscles and tendons have a certain degree of elasticity. One of the important points of ecological balance is that these material properties will dramatically reduce the amount of control required to achieve the same kinds of movement as compared to a completely stiff system. Imagine that you are sitting at your desk and you intend to grab the coffee cup sitting on the desk in front of you. There is a natural position for your arms which is determined by the anatomy of your torso, shoulders, and arms, and by the elastic, material properties of the muscle-tendon system. Grasping the cup with your right hand would normally be done with the palm facing left, but you could also—with considerable additional effort—grab the cup with your right hand twisted such that the palm is facing to the right. If you now relax the muscles from this awkward position, your arm will automatically turn back to its natural position. This is achieved not by neural control but by the material properties of the muscle-tendon system. Normally in robotics, returning to a default position is a function of electronic control, whereas for agents with biological muscles, it is achieved (mostly) through the material properties of the muscle-tendon system. In other words, the materials of the muscle-tendon system take over some of the control tasks that the brain, if the system had been designed without muscles, would have to deal with explicitly. So, to simplify the problem, when building our robots we might consider using artificial muscles that have properties similar to natural ones. Thus it could be said that neural control or program control is traded against materials.

In our discussion we have focused on the material properties of the muscle-tendon system. But it is clear that the morphology itself—or, as we say when talking about humans, the anatomy—also provides important constraints which make control much easier. For example, the skeletal arrangement of the human hand, together with the tissue holding the hand together, guarantees that when it closes, the fingers naturally come together.

Puppy as an Ecologically Balanced Robot

We can use our case study of the robot Puppy to illustrate some of the important points about ecological balance. Robotics researchers often come from a background of control theory, and some control theorists argue that the bottleneck in achieving rapid locomotion in robots is the electronics for controlling the sensors and motors. In other words, the circuits are too slow to process the sensor signals and calculate the motor commands fast enough. This is a puzzling idea, because today's electronics have cycle times on the order of microseconds (10^{-6} sec) to nanoseconds (10^{-9} sec), whereas the neural substrate of biological systems is much slower with "cycle times," so to speak, somewhere between 10 (10^{-2} sec) and 100 milliseconds (10^{-1} sec). Of course, we cannot really speak of cycle times in biological neural networks because they are continuous and have no clock like digital electronic circuits, but clearly the operating time scale of biological neurons is much slower. Nevertheless, there are biological organisms like dogs, horses, cheetahs, and humans that move much faster than today's legged robots. There are probably two factors involved in this surprising fact. First, biological organisms benefit from the massive parallelism of their neural systems, as well as the existence of local reflexes. In other words the signals do not have to travel all the way from the muscles up to the brain and back but they can be processed directly by the spinal cord, thus shortening the response times significantly.

But this alone would not suffice to make biological systems so fast. What is required in addition is the exploitation of the morphological and material properties of their bodies as the agent interacts with the real world. Let us briefly explain what this phrase means. Recent thinking in biomechanics (e.g., Blickhan et al., 2003), the field in which locomotion behavior of animals and humans is studied, draws our attention to the importance of the springlike properties of the muscle-tendon system. For example, the way that the knee joint moves when your foot hits the ground is not controlled by the brain or the spinal cord: rather, it is a result of the elastic properties of the leg's muscle-tendon system. What the neural system does control is the specific elasticity of the system: neural signals create a particular elastic stiffness of the muscle in the leg according to the phase of the gait that the animal is currently in. Therefore the trajectories of the individual joints are not controlled completely by the brain or spinal cord; some of the control is taken over by the material properties of the system itself.

Another instance of exploitation of morphological properties is the passive swing of the leg during human walking, a phenomenon that is mirrored in the design of the passive dynamic walker and its offshoot "Denise," discussed earlier in this chapter. During the swing phase there is little control of the leg's motion: the desired movement is achieved by passive exploitation of gravity and momentum. Robot designers have traditionally ignored this fact, and instead have tried to reproduce the walking movements in humanoid robots using complex control algorithms. As a consequence, the robots, even though some have achieved considerable speed, do not move naturally and only in certain environments, for example only on flat surfaces with particular frictional properties.

As we explained earlier, Puppy's legs are moved back and forth by servo motors at the "shoulder" and "hip" joints only; all the other joints are passive: they are not driven by any motors. The two springs that are attached to each of the legs (see figure 3.2b) can be seen as very simple artificial muscles or muscle-tendon combinations, and because of their intrinsic material properties less electronics are required: the springs take over the task that would otherwise have to be explicitly controlled. Springs are, of course, extremely simple, but they do capture some of the key properties of natural muscle-tendon systems, such as the elastic movement of the knee joint when the foot hits the ground. One of the problems with springs, though, is that their spring constant (that is, how stiff they are) does not change, whereas an important property of natural muscles is that their spring constants, so to speak, can be changed on the fly to meet the demands of the current situation. For example, on impact it is important that the muscles controlling the knee joint have the right stiffness. The higher an animal jumps, the more stiffness is required to support the body on landing, but there still must be some elasticity to soften the impact. But what exactly is the right stiffness for running or for jumping? Note how our thinking has moved from controlling trajectories of joints to controlling morphological properties; now we are asking what the right material properties of Puppy's springs should be. It is just this focus on morphology that we want to stress in artificial intelligence, because such considerations will benefit our design of robots and, ultimately, our understanding of intelligence. This also relates back to the idea of designing for emergence: if we get the material properties right, the desired trajectories will emerge from the interaction with the environment. Finding the proper stiffness for each situation, however, is a hard problem and will require a lot of research.

Artificial muscles are an emerging robot technology that now exist in many variations, but the most popular kind so far has been the pneumatic actuator—a kind of rubber tube surrounded by a braided fabric—that contracts when air pressure is applied. Because of the rubbery material, there is intrinsic elasticity and passive compliance, meaning that the muscle will yield elastically if the agent in which it is embedded encounters an object. And if we have robots interacting with humans, we want them to yield elastically so they will not hurt anyone: this general idea of yielding to external objects is known in robotics as *compliance*. A number of other technologies for artificial muscles are beginning to be used by roboticists: polymers that work on the basis of charge displacement; gels that contract depending on the chemical properties of the solution they are immersed in; metals whose lengths vary depending on the current that flows through them; and several others that are still just being developed in research laboratories. Like any kind of technology, each variety of artificial muscle has its pros and cons. Some cannot be bought off the shelf, some can extend quickly but only retract slowly; another type may wear out quickly or be too slow, etc. Pneumatic actuators are fast and robust and can be bought off the shelf in many variations. Their main disadvantage is that pressurized air is required for their operation and that they have to be controlled by valves.

One desirable property that we get free from artificial muscles—in contrast to servomotors—is that because of their springlike properties, they act as energy stores: on impact, part of the kinetic energy from the flight phase is transformed into potential energy in the muscle (or rather the muscle-tendon system), and some of it can be reused for the next step. A hopping kangaroo, for instance, regains about 40% of the energy absorbed in landing when it bounces up again (Vogel, 1998).

But back to Puppy. The right combination of material and morphological properties, i.e., the particular shape of the body and the limbs, is what allows Puppy to run. The servo motors that move the legs back and forth provide the energy supply and the basic rhythmic activation. The springs, the elastic spine, and the specific morphology take care of the harmonious distribution of the forces throughout Puppy's body when it interacts with the environment and make it adaptive to variations in its environment. The slightly slippery materials of the feet provide the additional degrees of freedom required for self-stabilization, the robot's ability to stabilize its gait without explicit control. Note that because Puppy is only a very simplified version of a dog, its dynamics is very different from that of an actual dog, but its movement is natural with respect to its own construction. As a consequence, there is a definite sense of aesthetics in Puppy's movements. You can verify the naturalness of Puppy's movements by watching the video clip at the book's Web page.

The Brain Controlling the Body, or Vice Versa?

While for a robot there is a clear distinction between the controller—which resides in the microprocessor—and the controlled—the actual physical robot, this distinction is far less clear in natural systems. Ultimately, the neural system of an animal or human is just as physically embodied as the rest of the body: it is not hidden away in a microprocessor that operates more or less independently from the body. One criterion that distinguishes the controller from the controlled in robots is energy consumption: typically the energy consumption of the controller is much less than that of the motors that are controlled by it. However, as is well known, the energy consumption of the human brain is very high, making up about 20% of the body's total energy usage. But the distinction gets even more fuzzy if we take into account that the body itself—the morphology and the materials—and the system-environment interaction also take over control tasks, i.e., perform morphological computation.

To illustrate: Imagine that you are running along a level jogging path and then the path goes downhill a bit. You will start running a bit faster, not because the brain "tells" the body to run faster, but because gravity accelerates the body, which in turn makes the limbs move faster, which in turn speeds up the brain's oscillatory circuits! So, the body "controls" the brain just as much as the brain controls the body. In other words, no one system is dominant over the other; the body and brain mutually determine each other's behavior. We will see more examples of this mutual coupling throughout the book.

"Computation" by Sensor Morphology: The "Eyebot"

In about 1995 the theoretical physicist and AI researcher Lukas Lichtensteiger, together with his colleague Peter Eggenberger, came up with a brilliant idea inspired by insects. In insects, at least in some species, the specific arrangement of the facets in their compound eye can be seen to perform an important function, i.e., to compensate for motion parallax. Facets are the small units that together make up an insect's compound eye. Motion parallax is just a fancy name for a phenomenon that is very familiar to all of us. Assume that you are sitting on a train looking out the window in the direction in which the train is moving and, still far away, you see a tree. As long as you are far away, this tree will move slowly across your visual field. When you pass close by the tree it will move much more quickly across your visual field, even though the train is moving at constant speed. This is purely a geometric phenomenon and holds for the human eye just as for the insect eye: objects nearby move faster across the visual field than objects farther away. Even though the insect eye is much more primitive than the human eye, it is nevertheless extremely effective and suited for its task, i.e., for guiding the insect during rapid flight.

The prominent neuroscientist and robot enthusiast Nicolas Franceschini, working at the Centre National de Recherche Scientifique (CNRS) in Marseille, France, found that in the housefly, the spacing of the facets is not homogeneous: the density toward the front is higher than on the side. What could be the advantage of this arrangement? First of all, it makes sense to have high resolution in the direction where you are going, which is usually forward. But second, with this arrangement of facets, a slow-moving point of light—from a distant object—will pass from one facet to the next at the front of the eye roughly at the same rate as a fast-moving point of light—from a close object—at the side of the eye. So the eye, because of its morphology, effectively compensates for motion parallax (see figure 4.5).

Let us assume that an insect "wants" to fly past an obstacle at a certain safe distance. One way of doing this is to maintain a fixed lateral distance from the object during flight, as do the railway lines going past the tree. Because of the facet distribution, all the insect needs to do is maintain a constant optic flow; that is, it has to move such that the time interval needed for a point of light to travel from one facet to the next remains constant: this is cheap design indeed! If there were a homogeneous arrangement of facets, because of motion parallax, computation would be more complicated (differently tuned neural circuits would have to be used for different pairs of facets). This is another illustration of morphological computation, or trading morphology for computation: the computation is, so to speak, performed by the morphology of the insect eye.

Inspired by these discoveries about the morphology of insect eyes, Lichtensteiger and Eggenberger developed the "Eyebot," a robot with a linear array of "facets," which are simply plastic tubes with a light sensor inside each one (see figure 4.5). These "facets" can be moved individually by electrical motors, and the motors in turn can be controlled by a program. Now, the ability to adapt one's behavior is normally attributed to plasticity of the brain. Lichtensteiger and Eggenberger were interested in the adaptive potential of morphology and asked the following question. Assume that an agent has the task of moving in such a way that its lateral distance to an obstacle remains constant: if we keep the brain fixed for the duration of the experiment, but we allow the agent to change its own morphology, will it be able to solve the task by adjusting its morphology (in this case the arrangement of the facets)? They ran an evolutionary algorithm (see chapter 6) on the "Eyebot" to optimize the angular position of the facets so that light would move past each facet

(a)

(b)

(c)

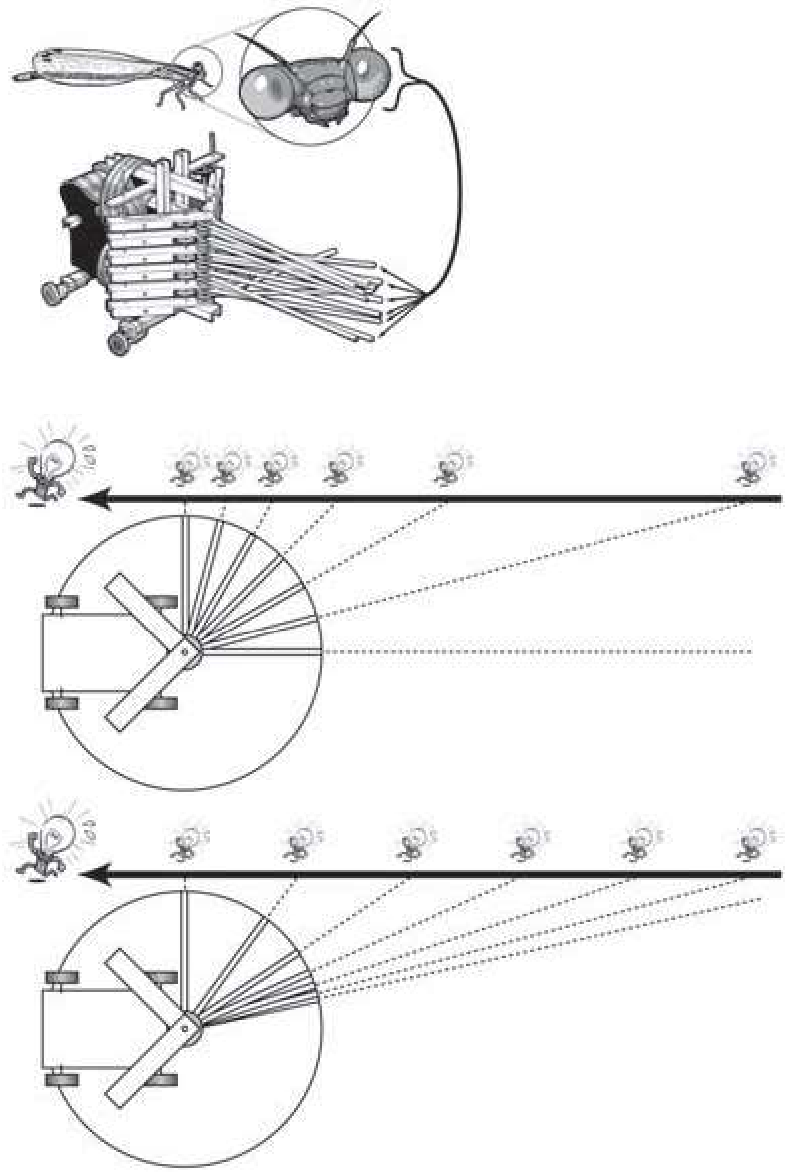

Figure 4.5
Ecological balance: morphological computation through sensor morphology. (*a*) The "Eyebot" has adjustable hollow tubes with light-sensitive cells at the base, thereby mimicking the facets of an insect eye. (*b*) If the facets are evenly spaced, a point of light, depicted by the running lightbulb, moves slowly across the visual field if the lightbulb is in front and far away, but moves fast as it passes by the side of the robot. This is the phenomenon of motion parallax. (*c*) If, however, the facets are more dense toward the front of the robot, a point of light will move at the same speed across all of the tubes, no matter whether it is in front or to the side of the robot; the motion parallax is therefore compensated away by this particular morphology.

pair at the same rate. Indeed, after about five hours, the robot managed to solve the problem: the resulting arrangement of the facets was similar to the one found in biological insects, with most clustered near the front and fewer arranged along the robot's side.

Morphological Computation, Cheap and Diverse Locomotion: Stumpy

At just about the start of the twenty-first century, Raja Dravid, a physicist, engineer, and self-made man who runs an "inventor's cooperative" in Zurich—together with the engineers and computer scientists Chandana Paul and Fumiya Iida—had an ingenious idea: they developed a very simple robot capable of many behaviors like walking, dancing, hopping, and turning. But rather than building a robot with legs and actuating them, they decided to actuate only the upper body.

Stumpy's lower body is made up of an inverted T mounted on wide, springy "feet." The upper body is an upright T connected to the lower body by a joint that can move back and forth, the "waist" joint: with this joint, Stumpy can move the upper body left and right, but cannot turn it (see figure 4.6). This upper horizontal beam is connected to the vertical beam by a second joint that can rotate left and right, providing an additional degree of freedom, the "shoulder" joint. So, Stumpy has two degrees of freedom: it can move its upper body left and right, and it can rotate its shoulder left and right, but it cannot bend forward and back. The horizontal beam at the top of the robot has weights attached to the ends in order to increase the effect of its movements. Since the first Stumpy, a whole series of Stumpies with somewhat different designs, morphologies, and materials have been built in order to explore the different ways in which simple bodies can give rise to lots of different behaviors.

Although Stumpy has no real legs or feet, it can move around in many ways: it can move forward in a straight or curved line, it has different gait patterns, it can move sideways, and it can turn on the spot. Interestingly, all this can be achieved by actuating only the two joints. In other words, control is extremely simple—the robot is virtually "brainless." The reason this works is because the dynamics, determined by its morphology, its materials (elastic, springlike materials, the surface properties of the feet), and the way it is actuated, are exploited in clever ways. Stumpy's many appealing and entertaining ways of moving arise not just from actuation of the two joints in particular ways, but because Stumpy is built in a specific manner (for more detail, see Iida et al., 2002 and Paul et al., 2002); if its morphology were different, it would exhibit less behavioral diversity, as illustrated in figures 4.6a and b.

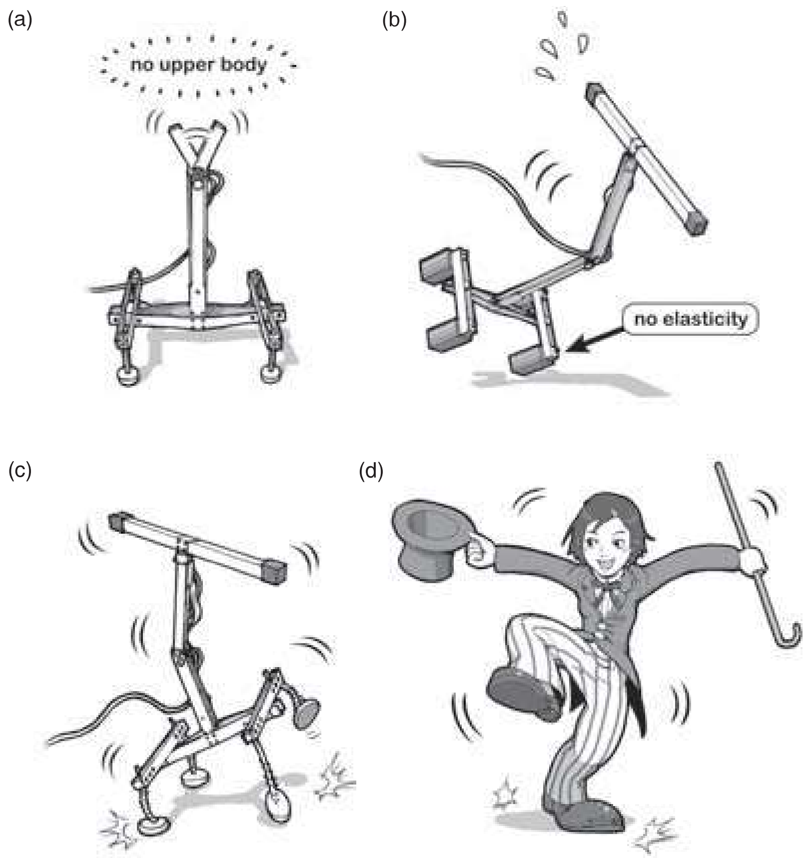

Figure 4.6
Ecological balance: morphological computation through shape and materials. Three morphologies are shown, two that do not work properly and one that achieves the desired dancing behavior. (*a*) A robot without a heavy enough upper body cannot generate enough momentum to get its feet off the ground. (*b*) A robot with no elasticity in its feet will not move properly or will fall over because the forces are not adequately propagated through the robot to the ground for locomotion. (*c*) Stumpy has the right morphology (an upper body) and the right materials (elastic feet) so that it can perform a large variety of interesting behaviors. (*d*) The biological system that is modeled by Stumpy: we use our upper body and the elasticity in our legs to move in interesting ways.

Before moving on to the next principle, let us briefly summarize the ideas concerning ecological balance, i.e., the interplay of morphology, materials, interaction with the environment, and control. First, given a particular task environment, the (physical) dynamics of the agent can be exploited which leads not only to a natural behavior of the agent, but also to greater energy efficiency. Second, when the dynamics of the agent is exploited, control can often be significantly simplified while a certain level of behavioral diversity is maintained. Third, materials have intrinsic control properties (e.g., stiffness, elasticity, and damping). And fourth, because ecological balance is exploited, agents like Stumpy can display surprisingly diverse behavior. In this sense, Stumpy also illustrates diversity-compliance: on the one hand, it exploits the physical dynamics in interesting ways and on the other it displays high behavioral diversity.

4.9 Agent Design Principle 7: Parallel, Loosely Coupled Processes

The principle of parallel, loosely coupled processes states that intelligence is emergent from a large number of parallel processes that are often coordinated through embodiment, in particular via the embodied interaction with the environment.

The way we like to view ourselves, and the way we usually conceptualize intelligence, is in terms of hierarchical organizations: there is the "I" that perceives an event in the outside world and maps the event onto an internal representation (e.g., a coffee cup standing on my desk), uses this representation to plan an action (drinking from the cup), and finally executes the action (reaching for the cup, grasping it, and drinking from it). This way of viewing behavior, also called the sense-think-act model, has proved inappropriate in the real world because (1) it is a one-way model, assuming that sensory stimulation comes first and leads to internal representation, and (2) because of real-time constraints, this way of functioning would simply not be fast enough. Recall our discussions of sensory-motor coordination and running (for additional arguments, see for example Pfeifer and Scheier, 1999). In reaction to this kind of thinking, in the mid-1980s Rodney Brooks of MIT suggested an alternative way of viewing intelligence, namely as a collection of parallel, asynchronous processes that are only loosely coupled. In this view intelligent behavior is, in essence, emergent from a large number of such processes. As discussed earlier, it was really Brooks who finally triggered the embodied turn in artificial intelligence. In a paper with the innocuous title "A Robust Layered Control System for a Mobile Robot," published in 1986, Brooks presented a radical alternative to designing control systems, the famous subsumption architecture (Brooks, 1986). The principle of parallel, loosely coupled processes is, in essence, a general way of interpreting the subsumption architecture. As outlined in chapter 2, the original publication was complemented later by the more provocatively titled papers "Intelligence Without Reason" and "Intelligence Without Representation." The debate on whether such architectures are suitable to achieve high-level intelligence is still open. We will return to this point later.

The term *loosely coupled* is used in contrast to hierarchically coupled processes. In the latter there is a control program (the "I") that calls the subroutines (e.g., for perception), and the calling program then has to wait for the subroutine (the perceptual act) to complete its task before it can continue (and go on to the action planning phase and then the action phase). This hierarchical control corresponds to very strong coupling; there is a very tight control regime between the calling and the called routines. But of course, in a complete agent there is strong coupling between processes simply because the system is embodied: for instance two joints such as the shoulder and the elbow, connected by a physical link (the upper arm), are very strongly coupled.

"Loosely coupled" also refers to the coupling of subsystems of an agent through its interaction with the environment, as we have seen in our discussion of leg coordination in insect walking, where the individual leg controllers were coupled through the interaction with the environment via the angle sensors in the joints of the legs. The coupling is called "loose" because the global coordination is achieved indirectly—through the environment—and not directly through the neural system. In grasping a coffee cup, the movement of the head, the eyes, the arms, and the fingers are also coupled through the interaction with the environment, so sensory-motor coordination always implies this kind of organization. Put another way, there is loose coupling between parallel processes, which in this case are the different sensory and motor processes involved in the grasping task: foveation—looking at the object—reaching, touching, and finally grasping. Note that in order to coordinate these processes, little internal neural processing is required: the coordination comes about through the environment.

Parallel, loosely coupled processes also play a role in social interaction. The social interaction robot Kismet, with gremlin-like features, which the robotics researcher Cynthia Breazeal developed while at the

Artificial Intelligence Laboratory at MIT, is another beautiful illustration of this design principle. Kismet is in fact simply a head, but by actuating various parts of its head—turning its head, focusing its eyes, or uttering sounds—it can engage an observer in seemingly complex social interactions. Rather than getting into the details of how Kismet functions, here we ask what we can learn from Breazeal's experiments, and provide our take on the question.

When watching Kismet interact with a person, one cannot help but attribute high social competence to this robot. It is essentially controlled by a collection of relatively simple reflexes that work in parallel. One reflex focuses on salient objects, i.e., objects that attract the robot's attention. A salient object might be one that has just appeared in the visual field, is moving rapidly, or is very bright. The object-tracking reflex causes the robot to follow slowly moving objects with its head and eyes, and a third reflex performs sound localization, turning the head in the direction of loud noises. There is also a habituation reflex, meaning that if the robot has been engaged in the same activity for some time it will get "bored," and look for something else to do. Note the anthropomorphic vocabulary that we are using, and remember to keep the frame of reference in mind: Kismet does not actually get bored (or does it?), but an observer may attribute boredom to Kismet based on its interactions with the environment. Despite the sophistication of Kismet, what matters for our discussion is that there are processes that work more or less independently of each other but are loosely coupled, i.e., they are coordinated through the interaction with the environment. Also, our simple description does not do justice to Kismet; for example, there is in fact a sophisticated model of emotion underlying Kismet's facial expressions that we will not discuss here (for more detail, see Breazeal, 2002).

Imagine now that I am talking to the robot so that it focuses on my face. If a door to the side opens with a noise and a person enters the room, the robot will turn its head toward the door (sound localization), it will track the human who has entered the room for a bit (following slowly moving objects), then it will get bored (habituation), and if I talk to Kismet again it will turn its head back toward me (sound localization) and continue our interaction. This kind of behavior is precisely what you would expect from a socially competent individual: someone new enters the room; you turn your head, perhaps briefly follow the person, and then turn back to your previous activity. One of the amazing things about Kismet is that it demonstrates that sophisticated algorithms or complex reasoning are not necessary to achieve this behavior. This leaves us with a deep philosophical question about human nature: perhaps we are much more driven by low-level reflexes rather than by our high-level rational thoughts. For some people, this idea is decidedly disconcerting, especially those with a Cartesian attitude: that is, people who believe there is a clear distinction between body and abstract thought, and that we can rationally decide what we want to do. Others might be relieved, because if our social abilities are indeed to a high degree controlled by reflexes and these reflexes are automatic, we do not have to think or worry about them: they take care of themselves. The latter is more related to the "Zen" attitude to being in the world. We surmise that this is why Rodney Brooks's term "the Zen of robot programming" has become a catchphrase among artificial intelligence researchers interested in embodiment.

4.10 Agent Design Principle 8: Value

The value principle states that intelligent agents are equipped with a value system which constitutes a basic set of assumptions about what is good for the agent.

The value principle is on the one hand very important because it deals with the fundamental issue of what is good for the agent, which then leads to the question of what the agent will or should do in a particular situation. On the other, the value principle is also extremely vague, and there is no consensus in the vast literature about how to approach it, neither in biology and psychology, nor in robotics and artificial intelligence. So, we cannot provide a satisfactory answer. All we can do, in contrast to the other design principles, is raise a number of issues for discussion. The question of value is certainly one of the open questions in intelligence research. We will start in this chapter and follow some of the points up in chapters 5 and 6.

Let us first talk about value in the context of designing and building artificial systems. The value principle states that intelligent agents are to be equipped with a value system which constitutes a basic set of assumptions about what is good for the agent. And once these assumptions have been made, they are no longer questioned—at least for a certain period of time, typically the lifetime of the agent. When designing, for instance, a companion robot (see chapter 11), the assumption is that anything that enables and helps the robot to perform its tasks—entertaining humans, serving coffee, mowing the lawn, performing household chores, looking after the kids, shopping—constitutes value. Thus, the set of design decisions make up the value system: cameras, microphones, wireless LAN, legs, arms and hands, mechanisms for walking, for manipulating objects, and for deciding what to do in a particular situation, etc. The more fully the agent conforms to the design principles outlined earlier, the more value it will be able to get from its setup (for example, it may be able to run more quickly if it exploits the elasticity in the artificial muscles). But we have to mind the frame-of-reference problem here: To the designer, these decisions are explicit, but once they are implemented on the robot, its behavior is emergent from a combination of all the components and mechanisms. So, the value is in the head of the designer rather than the head of the robot.

Let us now turn to a more specific question: given a particular agent, how does it decide what to do in a particular situation? This is especially important if the agents are to be autonomous and self-sufficient like the Fungus Eaters, which always have to achieve a number of tasks in order to keep functioning. Often, so-called action selection schemes are used: given a particular situation—e.g., the children have come home from school, there is no ice cream in the fridge, and the vacuum cleaner is broken—there are a number of actions the robot can take: buy cookies, take the vacuum cleaner to the repair shop, play with the children, etc. From these alternatives one is chosen based on an analysis of the current circumstances and an evaluation of the alternatives. This kind of approach is often employed in real-world applications where the objective is to build a working robot. But how much can we learn about intelligent behavior from this approach, which essentially implements how we as designers feel decision making is best done? We can learn about how well robots programmed in this way can function in dynamic complex environments such as people's homes, but this may in fact bear little relation to how "decisions" are taken in biological systems such as humans.

Let us briefly illustrate this point here with an example from psychology, the famous "A not B error," originally studied by Piaget. Imagine an experimenter at a table across from a baby sitting on his mother's lap. There are two holes in the table, A and B, each covered with a lid. The experimenter takes a toy, shakes it in front of the baby to attract his attention and puts the toy into hole A, and repeats this procedure a few times. It turns out that in most cases the baby will reach for hole A and take off that lid. Then, again in front of the eyes of the baby after shaking the toy back and forth, the experimenter puts it into hole B. Surprisingly, the baby will reach for lid A. This effect, called the "A not B error" has been shown to occur in babies aged seven to twelve months. Most of the literature tries to explain this phenomenon in terms of the cognitive processes of the babies. By contrast, Thelen and colleagues (2001) hypothesized that, rather than being the result of cognitive processes, this behavior might be emergent from a dynamical system. And indeed, if the physical dynamics of the system (the reaching system of the baby) is changed, the baby no longer makes the error. For example, when, after the training phase, the position of the baby is changed from sitting to upright, or when weights are attached to the baby's arms—both measures that change the physical dynamics of the reaching system—the baby no longer makes the error. The explanation is that through the various trials in the experiment, the babies, viewed as dynamical systems, get "stuck" in a particular attractor state from which they cannot escape unless the dynamics of the system is changed. At a later age, the external stimulus of the experimenter who puts the toy into hole B is sufficient to change attractors, and the babies do not make the error any more. Thus, something that looks very much like action selection, or a cognitive decision process, might in fact be emergent from a dynamical system.

This relates to the general issue of how to conceptualize the behavior of biological agents in complex situations when trying to explain their motivation, which is, in essence, the question of value. Without going into the details—there is a substantial literature on this issue—we have a strong tendency to attribute goals and decision processes to other humans (and even to animals and robots), which is in line with a Cartesian mind-set: we have a goal, and then we plan and execute our actions to achieve the goal. Alas, it seems that goals are more like post hoc rationalizations, attributed to give the behavior the flavor of coherence, than the actual causes of behavior (for a review of these issues, see McFarland and Bösser, 1993; Pfeifer and Scheier, 1999; or the collection of articles in Montefiore and Noble, 1989). One of the key insights from the embodied approach has been that often much simpler explanations can be given and that there is no need to attribute sophisticated goal hierarchies or decision processes to the agent. An instructive example is Kismet, whose behavior, in essence, is emergent from a number of reflexes. And in the "A not B" experiment, the apparent decision behavior is emergent from a dynamical system, the baby's reaching system. These insights might provide valuable intuitions for the design of artificial agents.

To conclude our (admittedly somewhat superficial) discussion of the value principle, let us briefly discuss the time frames. What we have been saying so far applies mostly to the "here-and-now" perspective, where the designer decides what will be of value for the robot to achieve its tasks. In chapter 5, we will provide the details on value from a developmental perspective. One of the deep and largely unresolved questions there is why an agent should learn anything in the first place. In other words, how is learning related to value? Why continue to acquire more and more sophisticated skills and not be happy with what you have? Chapter 6 will discuss the evolutionary perspective on value, which raises the conundrum of why organisms become more complex during the process of evolution—that is, of how increased complexity is linked to value.

4.11 Summary and Conclusions

In this chapter we have outlined a set of principles that, on the one hand, characterize biological systems and on the other can be employed as heuristics for designing and building artificial ones. Although we are convinced that these principles are essential and capture the major insights into the intricacies of how intelligent behavior comes about, they constitute a preliminary set that will eventually need to be extended and revised. The basic set outlined in this chapter will be complemented in the subsequent three chapters by a number of additional principles for development, evolution, and collective intelligence. We have tried our best to boil down the principles to the bare minimum while maintaining comprehensibility: for a more detailed, but perhaps somewhat less up-to-date elaboration, see Pfeifer and Scheier (1999). A summary of all the design principles from chapters 4, 5, 6, and 7 will be given in the concluding chapter of the book.