1
The Nature of Cognition

1.1 Motivation for Studying Artificial Cognitive Systems

When we set about building a machine or writing a software application, we usually have a clear idea of what we want it to do and the environment in which it will operate. To achieve reliable performance, we need to know about the operating conditions and the user's needs so that we can cater for them in the design. Normally, this isn't a problem. For example, it is straightforward to specify the software that controls a washing machine or tells you if the ball is out in a tennis match. But what do we do when the system we are designing has to work in conditions that aren't so well-defined, where we cannot guarantee that the information about the environment is reliable, possibly because the objects the system has to deal with might behave in an awkward or complicated way, or simply because unexpected things can happen?

Let's use an example to explain what we mean. Imagine we wanted to build a robot that could help someone do the laundry: load a washing machine with clothes from a laundry basket, match the clothes to the wash cycle, add the detergent and conditioner, start the wash, take the clothes out when the wash is finished, and hang them up to dry (see Figure 1.1). In a perfect world, the robot would also iron the clothes,[1] and put them back in the wardrobe. If someone had left a phone, a wallet, or something else in a pocket, the robot should either remove it before putting the garment in the wash or put the garment to

[1] The challenge of ironing clothes as a benchmark for robotics [1] was originally set by Maria Petrou [2]. It is a difficult task because clothes are flexible and unstructured, making them difficult to manipulate, and ironing requires careful use of a heavy tool and complex visual processing.

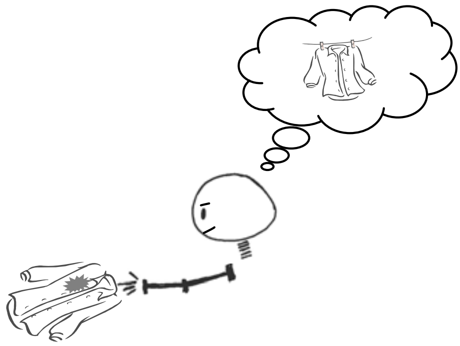

Figure 1.1: A cognitive robot would be able to see a dirty garment and figure out what needs to be done to wash and dry it.

one side to allow a human to deal with it later. This task is well beyond the capabilities of current robots[2] but it is something that humans do routinely. Why is this? It is because we have the ability to look at a situation, figure out what's needed to achieve some goal, anticipate the outcome, and take the appropriate actions, adapting them as necessary. We can determine which clothes are white (even if they are very dirty) and which are coloured, and wash them separately. Better still, we can also learn from experience and adapt our behaviour to get better at the job. If the whites are still dirty after being washed, we can apply some extra detergent and wash them again at a higher temperature. And best of all, we usually do this all on our own, autonomously, without any outside help (except maybe the first couple of times). Most people can work out how to operate a washing machine without reading the manual, we can all hang out damp clothes to dry without being told how to do it, and (almost) everyone can anticipate what will happen if you wash your smartphone.

[2] Some progress has been made recently in developing a robot that can fold clothes. For example, see the article "Cloth grasp point detection based on multiple-view geometric cues with application to robotic towel folding" by Jeremy Maitin-Shepard *et al.* [3] which describes how the PR2 robot built by Willow Garage [4] tackles the problem. However, the focus in this task is not so much the ill-defined nature of the job — how do you sort clothes into different batches for washing and, in the process, anticipate, adapt, and learn — as it is on the challenge of vision-directed manipulation of flexible materials.

We often refer to this human capacity for self-reliance, for being able to figure things out, for independent adaptive anticipatory action, as *cognition*. What we want is the ability to create machines and software systems with the same capacity, i.e., *artificial cognitive systems*. So, how do we do it? The first step would be to model cognition. And this first step is, unfortunately, where things get difficult because cognition means different things to different people. The issue turns on two key concerns: (a) the purpose of cognition — the role it plays in humans and other species, and by extension, the role it should play in artificial systems — and (b) the mechanisms by which the cognitive system fulfils that purpose and achieves its cognitive ability. Regrettably, there's huge scope for disagreement here and one of the main goals of this book is to introduce you to the different perspectives on cognition, to explain the disagreements, and to tease out their differences. Without understanding these issues, it isn't possible to begin the challenging task of developing artificial cognitive systems. So, let's get started.

1.2 *Aspects of Modelling Cognitive Systems*

There are four aspects which we need to consider when modelling cognitive systems:[3] how much inspiration we take from natural systems, how faithful we try to be in copying them, how important we think the system's physical structure is, and how we separate the identification of cognitive capability from the way we eventually decide to implement it. Let's look at each of these in turn.

[3] For an alternative view that focusses on assessing the contributions made by particular models, especially computational and robotic models, see Anthony Morse's and Tom Ziemke's paper "On the role(s) of modelling in cognitive science" [5].

To replicate the cognitive capabilities we see in humans and some other species, we can either invent a completely new solution or draw inspiration from human psychology and neuroscience. Since the most powerful tools we have today are computers and sophisticated software, the first option will probably be some form of computational system. On the other hand, psychology and neuroscience reflect our understanding of biological life-forms and so we refer to the second option as a bio-inspired system. More often than not, we try to blend the two together. This balance of pure computation and bio-inspiration is the first aspect of modelling cognitive systems.

Unfortunately, there is an unavoidable complication with the bio-inspired approach: we first have to understand how the biological system works. In essence, this means we must come up with a model of the operation of the biological system and then use this model to inspire the design of the artificial system. Since biological systems are very complex, we need to choose the level

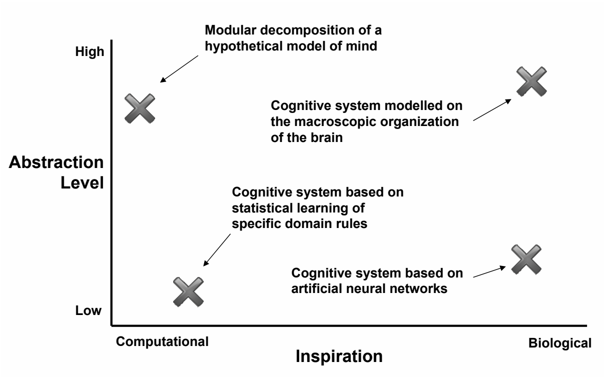

Figure 1.2: Attempts to build an artificial cognitive system can be positioned in a two-dimensional space, with one axis defining a spectrum running from purely computational techniques to techniques strongly inpired by biological models, and with another axis defining the level of abstraction of the biological model.

of abstraction at which we study them. For example, assuming for the moment that the centre of cognitive function is the brain (this might seem a very safe assumption to make but, as we'll see, there's a little more to it than this), then you might attempt to replicate cognitive capacity by emulating the brain at a very high level of abstraction, e.g. by studying the broad functions of different regions in the brain. Alternatively, you might opt for a low level of abstraction by trying to model the exact electrochemical way that the neurons in these regions actually operate. The choice of abstraction level plays an important role in any attempt to model a bio-inspired artificial cognitive system and must be made with care. That's the second aspect of modelling cognitive systems.

Taking both aspects together — bio-inspiration and level of abstraction — we can position the design of an artificial cognitive system in a two-dimensional space spanned by a computational / bio-inspired axis and an abstraction-level axis; see Figure 1.2. Most attempts today occupy a position not too far from the centre, and the trend is to move towards the biological side of the computational / bio-inspired spectrum and to cover several levels of abstraction.

In adopting a bio-inspired approach at any level of abstraction it would be a mistake to simply replicate brain mechanisms in complete isolation in an attempt to replicate cognition. Why? Because the brain and its associated cognitive capacity is the result

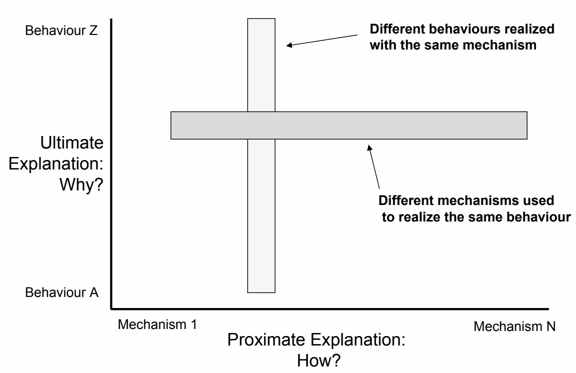

Figure 1.3: The ultimate-proximate distinction. Ultimate explanations deal with *why* a given behaviour exists in a system, while proximate explanations address the specific mechanisms by which these behaviours are realized. As shown here, different mechanisms could be used to achieve the same behaviour or different behaviours might be realized with the same mechanism. What's important is to understand that identifying the behaviours you want in a cognitive system and finding suitable mechanisms to realize them are two separate issues.

of evolution and the brain evolved for some purpose. Also, the brain and the body evolved together and so you can't divorce one from the other without running the risk of missing part of the overall picture. Furthermore, this brain-body evolution took place in particular environmental circumstances so that the cognitive capacity produced by the embodied brain supports the biological system in a specific ecological niche. Thus, a complete picture may really require you to adopt a perspective that views the brain and body as a complete system that operates in a specific environmental context. While the environment may be uncertain and unknown, it almost always has some in-built regularities which are exploited by brain-body system through its cognitive capacities in the context of the body's characteristics and peculiarities. In fact, the whole purpose of cognition in a biological system is to equip it to deal with this uncertainty and the unknown nature of the system's environment. This, then, is the third aspect of modelling cognitive systems: the extent to which the brain, body, and environment depend on one another.[4]

[4] We return to the relationship between the brain, body, and environment in Chapter 5 on embodiment.

Finally, we must address the two concerns we raised in the opening section, i.e., the purpose of cognition and the mechanisms by which the cognitive system fulfils that purpose and achieves its cognitive ability. That is, in drawing on bio-inspiration, we need to factor in two complementary issues: what cognition is for and how it is achieved. Technically, this is known as the *ultimate-proximate distinction* in evolutionary psychology; see Figure 1.3. Ultimate explanations deal with questions concerned with *why* a given behaviour exists in a system or is selected through evolution, while proximate explanations address the specific mechanisms by which these behaviours are realized. To build a complete picture of cognition, we must address both explanations. We must also be careful not to get the two issues mixed up, as they very often are.[5] Thus, when we want to build machines which are able to work outside known operating conditions just like humans can — to replicate the cognitive characteristics of smart people — we must remember that this smartness may have arisen for reasons other than the ones in which it is being deployed in the current task-at-hand. Our brains and bodies certainly didn't evolve so that we could load and unload a washing machine with ease, but we're able to do it nonetheless. In attempting to use bio-inspired cognitive capabilites to perform utilitarian tasks, we may well be just piggy-backing on a deeper and quite possibly quite different functional capacity. The core problem then is to ensure that this *system* functional capacity matches the ones we need to get *our* job done. Understanding this, and keeping the complementary issues of the purpose and mechanisms of cognition distinct, allows us to keep to the forefront the important issue of how one can get an artificial cognitive system (and a biological one, too, for that matter) to do what we want it to do. If we are having trouble doing this, the problem may not be the operation of the specific (proximate) mechanisms of the cognitive model but the (ultimate) selection of the cognitive behaviours and their fitness for the given purpose in the context of the brain-body-mind relationship.

[5] The importance of the ultimate-proximate distinction is highlighted by Scott-Phillips *et al.* in a recent article [6]. This article also points out that ultimate and proximate explanations of phenomena are often confused with one another so we end up discussing proximate concerns when we really should be discussing ultimate ones. This is very often the case with artificial cognitive systems where there is a tendency to focus on the proximate issues of *how* cognitive mechanisms work, often neglecting the equally important issue of *what* purpose cognition is serving in the first place. These are two complementary views and both are needed. See [7] and [8] for more details on the ultimate-proximate distinction.

To sum up, in preparing ourselves to study artificial cognitive systems, we must keep in mind four important aspects when modelling cognitive systems:

1. The computational / bio-inspired spectrum;

2. The level of abstraction in the biological model;

3. The mutual dependence of brain, body, and environment;

4. The ultimate-proximate distinction (why *vs.* how).

Understanding the importance of these four aspects will help us make sense of the different traditions in cognitive science, artificial intelligence, and cybernetics (among other disciplines) and the relative emphasis they place on the mechanisms and the purpose of cognition. More importantly, it will ensure we are addressing the right questions in the right context in our efforts to design and build artificial cognitive systems.

1.3 So, What Is Cognition Anyway?

It should be clear from what we have said so far that in asking "what is cognition?" we are posing a badly-framed question: what cognition *is* depends on what cognition is *for* and *how* cognition is realized in physical systems — the ultimate and proximate aspects of cognition, respectively. In other words, the answer to the question depends on the context — on the relationship between brain, body, and environment — and is heavily coloured by which cognitive science tradition informs that answer. We devote all of Chapter 2 to these concerns. However, before diving into a deep discussion of these issues, we'll spend a little more time here setting the scene. In particular, we'll provide a generic characterization of cognition as a preliminary answer to the question "what is cognition?", mainly to identify the principal issues at stake in designing artificial cognitive systems and always mindful of the need to explain how a given system addresses the four aspects of modelling identified above. Now, let's cut to the chase and answer the question.

Cognition implies an ability to make inferences about events in the world around you. These events include those that involve the cognitive agent itself, its actions, and the consequences of those actions. To make these inferences, it helps to remember what happened in the past since knowing about past events helps to anticipate future ones.[6] Cognition, then, involves predicting the future based on memories of the past, perceptions of the present, and in particular anticipation of the behaviour[7] of the world around you and, especially, the effects of your actions in it. Notice we say actions, not movement of motions. Actions usually involve movement or motion but an action also involves

[6] We discuss the forward-looking role of memory in anticipating events in Chapter 7.

[7] Inanimate objects don't behave but animate ones do, as do inanimate objects being controlled by animate ones (e.g. cars in traffic). So agency, direct or indirect, is implied by behaviour.

something else. This is the *goal* of the action: the desired outcome, typically some change in the world. Since predictions are rarely perfect, a cognitive system must also learn by observing what does actually happen, assimilate it into its understanding, and then adapt the way it subsequently does things. This forms a continuous cycle of self-improvement in the system's ability to anticipate future events. The cycle of anticipation, assimilation, and adaptation supports — and is supported by — an on-going process of action and perception; see Figure 1.4.

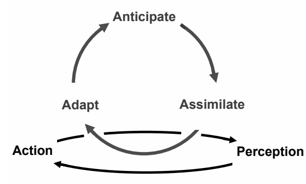

Figure 1.4: Cognition as a cycle of anticipation, assimilation, and adaptation: embedded in, contributing to, and benefitting from a continuous process of action and perception.

We are now ready for our preliminary definition.

> Cognition is the process by which an autonomous system perceives its environment, learns from experience, anticipates the outcome of events, acts to pursue goals, and adapts to changing circumstances.[8]

We will take this as our preliminary definition of cognition and, depending on the approach we are discussing, we will adjust it accordingly in later chapters.

While definitions are convenient, the problem with them is that they have to be continuously amended as we learn more about the thing they define.[9] So, with that in mind, we won't become too attached to the definition and we'll use it as a memory aid to remind us that cognition involved at least six attributes of autonomy, perception, learning, anticipation, action, and adaptation.

For many people, cognition is really an umbrella term that covers a collection of skills and capabilities possessed by an agent.[10] These include being able to do the following.

- Take on goals, formulate predictive strategies to achieve them, and put those strategies into effect;
- Operate with varying degrees of autonomy;
- Interact — cooperate, collaborate, communicate — with other agents;
- Read the intentions of other agents and anticipate their actions;
- Sense and interpret expected and unexpected events;

[8] These six attributes of cognition — autonomy, perception, learning, anticipation, action, adaptation — are taken from the author's definition of cognitive systems in the Springer *Encyclopedia of Computer Vision* [9]

[9] The Nobel laureate, Peter Medawar, has this to say about definitions: "My experience as a scientist has taught me that the comfort brought by a satisfying and well-worded definition is only short-lived, because it is certain to need modification and qualification as our experience and understanding increase; it is explanations and descriptions that are needed" [10]. Hopefully, you will find understandable explanations in the pages that follow.

[10] We frequently use the term *agent* in this book. It means any system that displays a cognitive capacity, whether it's a human, or (potentially, at least) a cognitive robot, or some other artificial cognitive entity. We will use *agent* interchangably with *artifical cognitive system*.

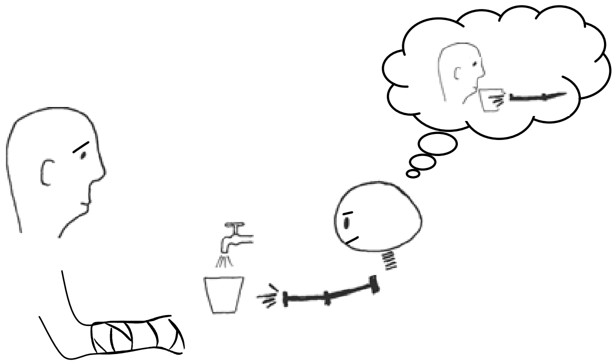

Figure 1.5: Another aspect of cognition: effective interaction. Here the robot anticipates someone's needs (see Chapter 9, Section 9.4 *Instrumental Helping*).

- Anticipate the need for actions and predict the outcome of its own actions and those of others;
- Select a course of action, carry it out, and then assess the outcome;
- Adapt to changing circumstances, in real-time, by adjusting current and anticipated actions;
- Learn from experience: adjust the way actions are selected and performed in the future;
- Notice when performance is degrading, identify the reason for the degradation, and take corrective action.

These capabilities focus on what the agent should do: its functional attributes. Equally important are the effectiveness and the quality of its operation: its non-functional characteristics (or, perhaps more accurately, its meta-functional characteristics): its dependability, reliability, usability, versatility, robustness, fault-tolerance, and safety, among others.[11]

These meta-functional characteristics are linked to the functional attributes through system capabilities that focus not on carrying out tasks but on maintaining the integrity of the agent.[12] Why are these capabilities relevant to artificial agents? They are relevant — and critically so — because artificial agents such as a robot that is deployed outside the carefully-configured environments typical of many factory floors have to deal with a

[11] The "non-" part of "non-functional" is misleading as it suggests a lesser value compared to functional characteristics whereas, in reality, these characteristics are equally important but complementary to functionality when designing a system. For that reason, we sometimes refer to them as meta-functional attributes; see [11] for a more extensive list and discussion of meta-functionional attributes.

[12] We will come back to the issue of maintaining integrity several times in this book, briefly in the next section, and more at length in the next chapter. For the moment, we will just remark that the processes by which integrity is maintained are known as *autonomic* processes.

world that is only partially known. It has to work with incomplete information, uncertainty, and change. The agent can only cope with this by exhibiting some degree of cognition. When you factor interaction with people into the requirements, cognition becomes even more important. Why? Because people are cognitive and they behave in a cognitive manner. Consequently, any agent that interacts with a human needs to be cognitive to some degree for that interaction to be useful or helpful. People have their own needs and goals and we would like our artificial agent to be able to anticipate these (see Figure 1.5). That's the job of cognition.

So, in summary, cognition is not to be seen as some module in the brain of a person or the software of a robot — a planning module or a reasoning module, for example — but as a system-wide process that integrates all of the capabilities of the agent to endow it with the six attributes we mentioned in our memory-aid definition: autonomy, perception, learning, anticipation, action, and adaptation.

1.3.1 *Why Autonomy?*

Notice that we included autonomy in our definition. We need to be careful about this. As we will see in Chapter 4, the concept of autonomy is a difficult one. It means different things to different people, ranging from the fairly innocent, such as being able to operate without too much help or assistance from others, to the more controversial, which sees cognition as one of the central processes by which advanced biological systems preserve their autonomy. From this perspective, cognitive development has two primary functions: (1) to increase the system's repertoire of effective actions, and (2) to extend the time-horizon of its ability to anticipate the need for and outcome of future actions.[13]

Without wishing to preempt the discussion in Chapter 4, because there is a tight relationship between cognition and autonomy — or not, depending on who you ask — we will pause here just a while to consider autonomy a little more.

From a biological perspective, autonomy is an organizational characteristic of living creatures that enables them to use their

[13] The increase of action capabilities and the extension anticipation capabilities as the primary focus of cognition is the central message conveyed in *A Roadmap for Cognitive Development in Humanoid Robots* [12], a multi-disciplinary book co-written by the author, Claes von Hofsten, and Luciano Fadiga.

own capacities to manage their interactions with the world in order to remain viable, i.e., to stay alive. To a very large extent, autonomy is concerned with the system maintaining itself: self-maintenance, for short.[14] This means that the system is entirely self-governing and self-regulating. It is not controlled by any outside agency and this allows it to stand apart from the rest of the environment and assert an identity of its own. That's not to say that the system isn't influenced by the world around it, but rather that these influences are brought about through interactions that must not threaten the autonomous operation of the system.[15]

If a system is autonomous, its most important goal is to preserve its autonomy. Indeed, it must act to preserve it since the world it inhabits that may not be very friendly. This is where cognition comes in. From this (biological) perspective, cognition *is* the process whereby an autonomous self-governing system acts effectively in the world in which it is embedded in order to maintain its autonomy.[16] To act effectively, the cognitive system must sense what is going on around it. However, in biological agents, the systems responsible for sensing and interpretation of sensory data, as well as those responsible for getting the motor systems ready to act, are actually quite slow and there is often a delay between when something happens and when an autonomous biological agent comprehends what has happened. This delay is called *latency* and it is often too great to allow the agent to act effectively: by time you have realized that a predator is about to attack, it may be too late to escape. This is one of the primary reasons a cognitive system must anticipate future events: so that it can prepare the actions it may need to take in advance of actually sensing that these actions are needed.

In addition to sensory latencies, there are also limitations imposed by the environment and the cognitive system's body. To perform an action, and specifically to accomplish the goal associated with an action, you need to have the relevant part of your body in a certain place at a certain time. It takes time to move, so, again, you need to be able to predict what might happen and prepare to act. For example, if you have to catch an object, you need to start moving your hand before the object arrives and

[14] The concepts of self-maintenance and recursive self-maintenance in self-organizing autonomous system was introduced by Mark Bickhard [13]. We will discuss them in more detail in Chapter 2. The key idea is that self-maintenant systems make active contributions to their own persistence but do not contribute to the maintenance of the conditions for persistence. On the other hand, recursive self-maintenant systems do contribute actively to the conditions for persistence.

[15] When an influence on a system isn't directly controlling it but nonetheless has some impact on the behaviour of the system, we refer to it as a *perturbation*.

[16] The idea of cognition being concerned with *effective action*, i.e. action that helps preserve the system's autonomy, is due primarily to Francisco Varela and Humberto Maturana [14]. These two scientists have had a major impact on the world of cognitive science through their work on biological autonomy and the organizational principles which underpin autonomous systems. Together, they provided the foundations for a new approach to cognitive science called *Enaction*. We will discuss enaction and enactive systems is more detail in Chapter 2.

sometimes even before it has been thrown. Also, the world in which the system is embedded is constantly changing and is outside the control of the system. Consequently, the sensory data which is available to the cognitive system may not only be late in arriving but critical information may also be missing. Filling in these gaps is another of the primary functions of a cognitive system. Paradoxically, it is also often the case that there is too much information for the system to deal with and it has to ignore some of it.[17]

[17] The problem of ignoring information is related to two problems in cogitive science: the *Frame Problem* and *Attention*. We will take up these issues again later in the book.

Now, while these capabilities derive directly from the biological autonomy-preserving view of cognition, it should be fairly clear that they would also be of great use to artificial cognitive systems, whether they are autonomous or not. However, before moving on to the next section which elaborates a little more on the relationship between biological and artificial cognitive systems, it is worth noting that some people consider that cognition should involve even more than what we have discussed so far. For example, an artificial cognitive system might also be able to explain what it is doing and why it is doing it.[18] This would enable the system to identify potential problems which could appear when carrying out a task and to know when it needed new information in order to complete it. Taking this to the next level, a cognitive system would be able to view a problem or situation in several different ways and to look at alternative ways of tackling it. In a sense, this is similar to the attribute we discussed above about cognition involving an ability to anticipate the need for actions and their outcomes. The difference in this case is that the cognitive system is considering not just one but *many* possible sets of needs and outcomes. There is also a case to be made that cognition should involve a sense of self-reflection:[19] an ability on the part of the system to think about itself and its own thoughts. We see here cognition straying into the domain of consciousness. We won't say anything more in this book on that subject apart from remarking that computational modelling of consciousness is an active area of research in which the study of cognition plays an important part.

[18] The ability not simply to act but to explain the reasons for an action was proposed by Ron Brachman in an article entitled "Systems that know what they're doing" [15].

[19] Self-reflection, often referred to as meta-cognition, is emphasized by some people, e.g. Aaron Sloman [16] and Ron Sun [17], as an important aspect of advanced cognition.

1.4 Levels of Abstraction in Modelling Cognitive Systems

All systems can be viewed at different levels of abstraction, successively removing specific details at higher levels and keeping just the general essence of what is important for a useful model of the system. For example, if we wanted to model a physical structure, such as a suspension bridge, we could do so by specifying each component of the bridge — the concrete foundations, the suspension cables, the cable anchors, the road surface, and the traffic that uses it — and the way they all fit together and influence one another. This approach models the problem at a very low level of abstraction, dealing directly with the materials from which the bridge will be built, and we would really only know after we built it whether or not the bridge will stay up. Alternatively, we could describe the forces at work in each member of the structure and analyze them to find out if they are strong enough to bear the required loads with an acceptable level of movement, typically as a function of different patterns of traffic flow, wind conditions, and tidal forces. This approach models the problem at a high level of abstraction and allows the architect to established whether or not his or her design is viable before it is constructed. For this type of physical system, the idea is usually to use an abstract model to validate the design and then realize it as a physical system. However, deciding on the best level of abstraction is not always straightforward. Other types of system — biological ones for example — don't yield easily to this top-down approach. When it comes to modelling cognitive systems, it will come as no surprise that there is some disagreement in the scientific community about what level of abstraction one should use and how they should relate to one another. We consider here two contrasting approaches to illustrate their differences and their relative merits in the context of modelling and designing artificial cognitive systems.

As part of his influential work on modelling the human visual system, David Marr[20] advocated a three-level hierarchy of abstraction;[21] see Figure 1.6. At the top level, there is the computational theory. Below this, there is the level of representation and algorithm. At the bottom, there is the hardware implementation.

[20] David Marr was a pioneer in the field of computer vision. He started out as a neuroscientist but shifted to computational modelling to try to establish a deeper understanding of the human visual system. His seminal book *Vision* [18] was published posthumously in 1982.

[21] Marr's three-level hierarchy is sometimes known as the *Levels of Understanding* framework.

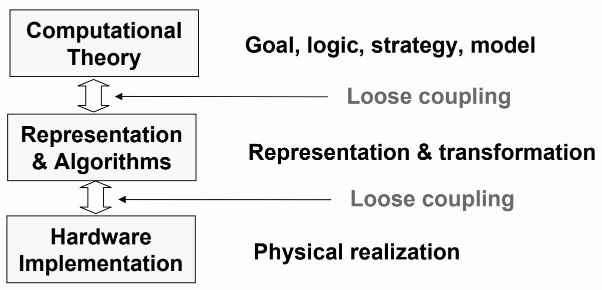

Figure 1.6: The three levels at which a system should be understood and modelled: the computational theory that formalizes the problem, the representational and algorithmic level that addresses the implementation of the theory, and the hardware level that phyically realizes the system (after David Marr [18]). The computational theory is primary and the system should be understood and modelled first at this level of abstraction, although the representational and algorithmic level is often more intuitively accessible.

At the level of the computational theory, you need to answer questions such as "what is the goal of the computation, why is it appropriate, and what is the logic of the strategy by which it is carried out?" At the level of representation and algorithm, the questions are different: "how can this computational theory be applied? In particular, what is the representation for the input and output, and what is the algorithm for the transformation?" Finally, the question at the level of hardware implementation is "how can the representation and algorithm be physically realized?" In other words, how can we build the physical system? Marr emphasized that these three levels are only loosely coupled: you can — and, according to Marr, you should — think about one level without necessarily paying any attention to those below it. Thus, you begin modelling at the computational level, ideally described in some mathematical formalism, moving on to representations and algorithms once the model is complete, and finally you can decide how to implement these representations and algorithms to realize the working system. Marr's point is that, although the algorithm and representation levels are more accessible, it is the computational or theoretical level that is critically important from an information processing perspective. In essence, he states that the problem can and should first be modelled at the abstract level of the computational theory without strong reference to the lower and less abstract levels.[22] Since many people believe that cognitive systems — both biological and artificial — are effectively information processors, Marr's hierarchy of abstraction is very useful.

[22] Tomaso Poggio recently proposed a revision of Marr's three-level hierarchy in which he advocates greater emphasis on the connections between the levels and an extension of the range of levels, adding *Learning and Development* on top of the computational theory level (specifically hierarchical learning), and *Evolution* on top of that [19]. Tomaso Poggio co-authored the original paper [20] on which David Marr based his more famous treatment in his 1982 book *Vision* [18].

Marr illustrated his argument succinctly by comparing the problem of understanding vision (Marr's own goal) to the problem of understanding the mechanics of flight.

> "Trying to understand perception by studying only neurons is like trying to understand bird flight by studying only feathers: it just cannot be done. In order to understand bird flight, we have to understand aerodynamics; only then do the structure of feathers and the different shapes of birds' wings make sense"

Objects with different cross-sectional profiles give rise to different pressure patterns on the object when they move through a fluid such as air (or when a fluid flows around an object). If you choose the right cross-section then there is more pressure on the bottom than on the top, resulting in a lifting force that counters the force of gravity and allows the object to fly. It isn't until you know this that you can begin to understand the problem in a way that will yield a solution for your specific needs.

Of course, you eventually have to decide how to realize a computational model but this comes later. The point he was making is that you should decouple the different levels of abstraction and begin your analysis at the highest level, avoiding consideration of implementation issues until the computational or theoretical model is complete. When it is, it can then subsequently drive the decisions that need to be taken at the lower level when realizing the physical system.

Marr's dissociation of the different levels of abstraction is significant because it provides an elegant way to build a complex system by addressing it in sequential stages of decreasing abstraction. It is a very general approach and can be applied successfully to modelling, designing, and building many different systems that depend on the ability to process information. It also echoes the assumptions made by proponents of a particular paradigm of cognition — *cognitivism* — which we will meet in the next chapter.[23]

Not everyone agrees with Marr's approach, mainly because they think that the physical implementation has a direct role to play in understanding the computational theory. This is particularly so in the emergent paradigm of embodied cognition which we will meet in the next chapter, the embodiment reflecting the physical implementation. Scott Kelso,[24] makes a case for a com-

[23] The cognitivist approach to cognition proposes an abstract model of cognition which doesn't require you to consider the final realization. In other words, cognitivist models can be applied to any platform that supports the required computations and this platform could be a computer or a brain. See Chapter 2, Section 2.1, for more details.

[24] Over the last 25 years, Scott Kelso, the founder of the Center for Complex Systems and Brain Sciences at Florida Atlantic University, has developed a theory of *Coordination Dynamics*. This theory, grounded in the concepts of self-organization and the tools of coupled nonlinear dynamics, incorporates essential aspects of cognitive function, including anticipation, intention, attention, multimodal integration, and learning. His book, *Dynamic Patterns – The Self-Organization of Brain and Behaviour* [21], has influenced research in cognitive science world-wide.

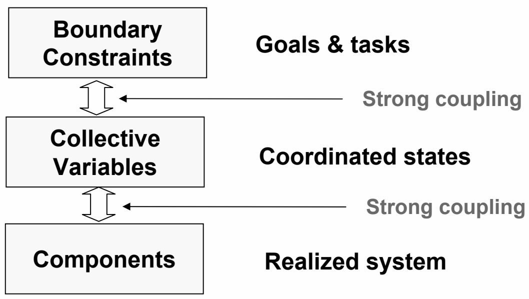

Figure 1.7: Another three levels at which a system should be modelled: a boundary constraint level that determines the task or goal, a collective variable level that characterizes coordinated states, and a component level which forms the realized system (after Scott Kelso [21]). All three levels are equally important and should be considered together.

pletely different way of modelling systems, especially non-linear dynamical types of systems that he believes may provide the true basis for cognition and brain dynamics. He argues that these types of system should be modelled at three distinct levels of abstraction, but at the same time. These three levels are a boundary constraint level, a collective variables level, and a components level. The boundary constraint level determines the goals of the system. The collective variable[25] level characterizes the behaviour of the system. The component level forms the realized physical system. Kelso's point is that the specification of these three levels of model abstraction are tightly coupled and mutually dependent. For example, the environmental context of the system often determines what behaviours are feasible and useful. At the same time, the properties of the physical system may simplify the necessary behaviour. Paraphrasing Rolf Pfeifer,[26] "morphology matters": the properties of the physical shape or the forced needed for required movements may actually simplify the computational problem. In other words, the realization of the system and its particular shape or morphology cannot be ignored and should not be abstracted away when modelling the system. This idea that you cannot model the system in isolation from either the system's environmental context or the system's ultimate physical realization is linked directly to the relationship between brain, body, and environment. We will meet it again later in the book when we discuss enaction in Chapter 2 and when we consider the issue of embodiment in Chapter 5.

The mutual dependence of system realization and system

[25] Collective variables, also referred to as order parameters, are so called because they are responsible for the system's overall collective behaviour. In dynamical systems theory, collective variables are a small subset of the system's many degrees of freedom but they govern the transitions between the states that the system can exhibit and hence its global behaviour.

[26] Rolf Pfeifer, University of Zurich, has long been a champion of the tight relationship between a system's embodiment and its cognitive behaviour, a relationship set out in his book *How the body shapes the way we think: A new view of intelligence* [22], co-authored by Josh Bongard.

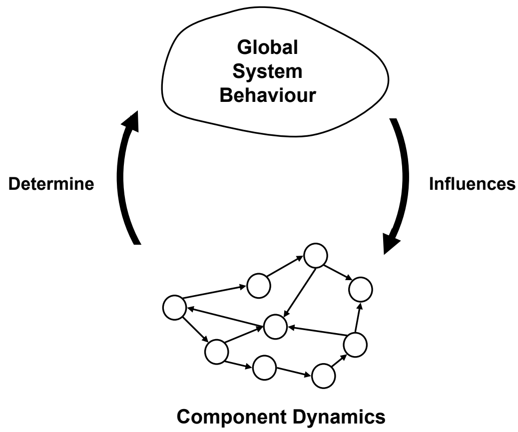

Figure 1.8: Circular causality — sometimes referred to as continuous reciprocal causation or recursive self-maintenance — refers to the situation where global system behaviour somehow influences the local behaviour of the system components and yet it is the local interaction between the components that determines the global behaviour. This phenomenon appears to be one of the pivotal mechanisms in autonomous cognitive systems.

modelling presents us with a difficulty, however. If we look carefully, we see a circularity, with everything depending on something else. It's not easy to see how you break into the modelling circle. This is one of the attractions of Marr's approach: there is a clear place to get started. This circularity crops up repeatedly in cognition and it does so in many forms. All we will say for the moment is that circular causality[27] — where global system behaviour somehow influences the local behaviour of the system components and yet it is the local interaction between the components that determines the global behaviour; see Figure 1.8 — appears to be one of the key mechanisms of cognition. We will return again to this point later in the book. For the moment, we'll simply remark that the two constrasting approaches to system modelling mirror two opposing paradigms of cognitive science. It is to these that we now turn in Chapter 2 to study the foundations that underpin our understanding of natural and artificial cognitive systems.

[27] Scott Kelso uses the term "circular causality" to describe the situation in dynamical systems where the cooperation of the individual parts of the system determine the global system behaviour which, in turn, governs the behaviour of these individual parts [21]. This is related to Andy Clark's concept of continuous reciprocal causation (CRC) [23] which "occurs when some system S is both continuously affecting and simultaneously being affected by, activity in some other system O" [24]. These ideas are also echoed in Mark Bickhard's concept of recursive self-maintenance [13]. We will say more about these matters in Chapter 4.