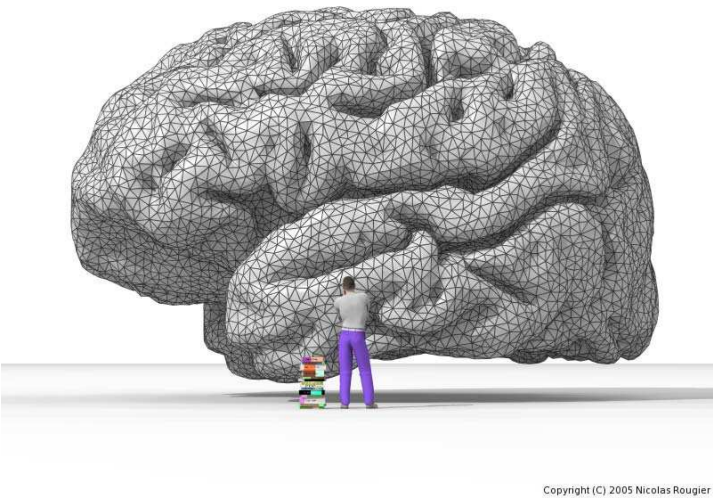
David Vernon

An Introduction to Computer Cognition



– Course Notes –

November 18, 2010

A cognitive system is an autonomous anti-entropy engine whose function is to preserve the system's autonomy.

Contents

Chapter 1
What is Cognition?

Cognition . . .the action or faculty of knowing, perceiving, conceiving . . .
f. L. *cognit-* ppl. stem of *cognoscere* [*co-* together + *(g)noscere* to apprehend]

Oxford English Dictionary

Cognition — understanding what's going on around you and acting in an appropriate way — has fascinated and perplexed people for centuries. It seems to be a peculiarly human trait, although there is a growing body of evidence that other species exhibit cognitive skills too. Cognition defies easy definition. It has been equated with rationality and reasoning, deliberation and abstract thought, and problem solving. While such concerns are clearly relevant, it is by no means clear that they necessarily form the essence of cognition. It all comes down to what we mean when we say we understand something and what we deem to be an appropriate way to behave. Does a central heating thermostat understand the need for warmth (and the consequences of not getting it) when it detects that the temperature of a room is too cold and switches on the heating? Certainly not in any meaningful way.[1] It seems that such a scenario is too trivial to be interesting. *Understanding* becomes interesting only if there are a lot of factors to be considered in assessing a situation and, especially, if they are complicated: they might conflict, there may be some essential information missing, they might be constantly changing, or they might simply be incorrect.[2] These complications mean that it is not sufficient for a cognitive system just to react to present circumstances, to how things are, to make some best-possible choice based on some or all of the available information, and then proceed on.

[1] The idea that a thermostat could legitimately be viewed as thinking and having beliefs (specifically, that the room is too hot, too cold, or ok) is due to John McCarthy of Stanford University and appears in a 1983 paper 'The Little Thoughts of Thinking Machines' in *Psychology Today* [135]; see `http://www.cse.msu.edu/ cse841/papers/McCarthy.pdf`. McCarthy is often referred to as the father of Artificial Intelligence.

[2] John McCarthy did extend his thermostat scenario to more complex situation where 'compromise' would be required. His essential point was that it helps for people to think about machines as having the capacity to think both when using them and when designing them.

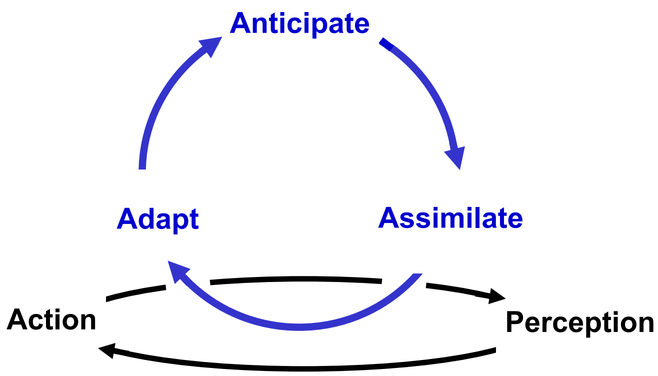


Fig. 1.1 Breaking the 'here-and-now barrier': cognition as a cycle of anticipation, assimilation, and adaptation, embedded in, contributing to, and benefi tting from a continuous process of action and perception.

Cognition implies an ability to understand how things might possibly be, not now but at some future time, and take this into consideration. Remembering what happened at some point in the past helps in anticipating future events, so memory is important too: using the past the predict the future[3] and then assimilating what does actually happen to adapt and improve the system's anticipatory ability in a virtuous cycle that is embedded in an on-going process of action and perception (see Figure 1.1).

Cognition breaks through the 'here-and-now barrier' and takes us into the future with the help of the past, in a way that allows the system to adapt and improve.

But what makes an action the right one to choose? Having broken through the here-and-now barrier, what type of behaviour does cognition enable? This opens up another dimension of the problem: what motivates cognition? How is perception guided? How are actions selected? And what makes cognition possible? Cognitive skills can improve, but what do you need to get started? What drives the developmental process? In other words, in addition to autonomous perception, action, anticipation, assimilation, and adaptation, there are the underlying motivations to consider. These motivations impact (or drive) perceptual attention, action selection, and system development, resulting in the long-term robust behaviour we seek from such systems.

---

[3] Berthoz puts it very succinctly: 'Memory is used primarily to predict the consequences of future action by recalling those of past action' [18].

Although the study of cognition in humans (and other species) is fascinating in its own right, this course is about *artificial* cognitive systems: computer cognition. We want to be able to build computer-based robotic systems that have this elusive cognitive capability. It is a course about synthesis: about design and implementation, about theory and models, and about their realization in working systems. However, it doesn't make a lot of sense to take on such an ambitious challenge in a vacuum, ignoring what we know about cognition in natural systems. On the contrary, we have a great deal to learn from such systems and we will embrace our knowledge of natural cognitive systems in psychology and in neuroscience and use it to inform our models of artificial cognitive systems.

Unfortunately, no one yet knows how to design and build an artificial cognitive system. There is no shortage of ideas and many alternative approaches have been proposed, but a complete convincing artificial cognitive system hasn't yet been developed. Understanding cognition, and modelling, designing, and building artificial cognitive systems are challenging long-term research problems.

The goal of this course is to provide you with a comprehensive overview of the many topics involved in cognition and cognitive systems and to do so from several perspectives. This presents a challenge for three reasons. First, the area is huge, as we will soon see. Second, it is not always easy to say where one should draw the boundary between cognition in natural systems and cognition in artificial computer-based systems. The boundary between these two fields is complicated. Sometimes, they form a symbiotic relationship, one learning from the other's successes and failures. Other times, they are antagonistic: limitations in our understanding of one field sometimes act as a brake on developments in the other field. Consequently, people sometimes decide to ignore these limitations and proceed with their own agenda in sometimes dangerous isolation. For some, the relationship between the two is strict and strong (even to the extent of being equivalent). For others, the relationship is loose, and ideas are borrowed and used at a conceptual level without concern for the plausibility of their implementation. The third reason why the goal of the course is challenging is that there is no universal agreement as to what cognition is, in the first place! Our aim here is to make sense of all this.

Given that cognitive systems research is such a big area, here is a disclaimer. This course is not a primer on artificial intelligence, psychology, neuroscience, non-linear dynamical systems theory, synergetics, autonomous systems theory, machine learning, pattern recognition, computer vision, haptic sensing, aural perception, cybernetics, neural networks, epistemology, philosophy, linguistics, semiotics, robotics, manipulation, or communication. It probably should be. All of these topics, and many others, impact in some way or other on cognitive systems and we will mention most of them at some point.

So, if the course doesn't attempting to introduce all of these disciplines in a substantive manner, even if we know it should, what exactly does it address? The answer to this question is provided by the diversity of constituent disciplines. We will try to identify the full scope of cognition and cognitive systems and we will try to provide a useful working definition, one that strikes a balance between being broad enough to do service to the many views that people have on cognition and deep enough to help in the formulation of theories and models. We will then present a summary of the many approaches that people adopt in researching and developing cognitive systems.

Ultimately, this course is intended to give you a clear understanding of the scope of the domain, its alternative approaches, and their underlying differences. Perhaps most important of all, it will give you a solid grasp of the issues that need to be addressed in striving for the goal of creating a true cognitive system. But this course does more than discuss what cognition is and what cognitive systems do. It also addresses *how* one can go about modelling, designing, and implementing cognitive systems. Finally, it discusses some important attempts that have been made to build cognitive systems.

1.1 Motivation for the Study of Cognitive Systems

Computer systems pervade many aspects of life today but they are not yet ubiquitous. Why? What stops computer systems being drummed into the service of society in every facet of life?[4] If we look at the applications in which computer systems have been most successful, we see that they are almost without exception the ones where the space of interaction are tightly constrained and unambiguously specified. In other words, we find computer systems wherever we can stipulate or predict exactly how they will be used, under what conditions, and with what data. Irrespective of the complexity of the processing they carry out, almost all computer systems, whether they are computer games, payroll applications, and even satellite navigation systems, share one common feature: well-defined inputs and outputs, and a well-specified space of interaction. To deal with circumstances where the space of interactions is poorly specified, we need to be able to build more robust, resilient, and adaptable computer systems. The hope is that we can do this by making them cognitive: by giving them the ability to learn, adapt, improve, and improvise to develop new strategies for interacting.

1.2 A Definition of Cognition

As we noted already, there is no universal consensus on what exactly cognition is. Nonetheless, to get things started, we will make an initial attempt to define the area of cognitive systems. We will expand on this later in the course.

A cognitive system exhibits effective — adaptive, anticipatory, and goal-directed — behaviour through perception, action, deliberation, communication, and through either individual or social interaction with the environment. The hallmark of a cogni-

[4] We don't take a position here on whether or not computer systems *should* be incorporated in every facet of life; we are only concerned about why, from a scientifi c and technological point of view, they haven't been.

tive system is that it can function effectively in circumstances that were not planned for explicitly when the system was designed. That is, it should have some degree of plasticity and be resilient in the face of the unexpected. The characteristic of anticipation, *i.e. prospective* behaviour, is crucial as it allows the system to operate across a variety of time-scales, in the here-and-now, but extending into the future. Thus, a cognitive system is capable of more than reactive stimulus-response behaviour, which might be quite complex in its own right.

To achieve this robust behaviour, cognitive systems anticipate, assimilate, and adapt. In doing so, they learn and develop [232]. That is, cgnitive systems anticipate future events when selecting actions, they subsequently learn from what actually happens when they do act, and thereby they modify subsequent expectations and, in the process, they change how the world is perceived and what actions are possible. Cognitive systems do all of this autonomously. The adaptive, anticipatory, autonomous viewpoint reflects the position of Freeman and Núñez who, in their book *Reclaiming Cognition* [60], assert the primacy of action, intention, and emotion in cognition. In the past, however, cognition was viewed in a very different light as a symbol-processing module of mind concerned with rational planning and reasoning. Today, however, this is changing and even proponents of these early approaches now see a much tighter relationship between perception, action, and cognition (e.g. see [6, 120]).

So, if cognitive systems anticipate, assimilate, and adapt, if they develop and learn, the question we must ask is *why* do they do this? There is a subsequent question – *how* do they do it? — but we leave this for later on. For the moment, the question *why* cognition is necessary provides a way of defining *what* cognition is.

The view of cognition taken in this course is that cognition is the process whereby an autonomous self-governing system acts effectively in the world in which it is embedded [133]. However, in natural systems, the latencies inherent in the neural processing of sense data are too great to allow effective action. This is one of the primary reasons a cognitive agent must anticipate future events: so that it can prepare the actions it may need to take. In addition, there are also limitations imposed by the environment and the cognitive system's body. To perform an action, one needs to have the relevant body part in a certain place at a certain time. In a dynamic environment that is constantly changing and with a body that takes time to move, this requires preparation and prediction. Furthermore, the world in which the agent is embedded is unconstrained and the sensory data which is available to the cognitive system is not only 'out-of-date' but it is also uncertain and incomplete. Consequently, it is not possible to encapsulate a priori all the knowledge required to deal successfully with the circumstances it will experience so that it must also be able to adapt, progressively increasing its space of possible actions as well as the time horizon of its prospective capabilities. It must do this, not as a reaction to external stimuli but as a self-generated process of proactive understanding. [5] This process is what we mean by development.

[5] As we will see later on, this is sometimes referred to as a process of 'sense-making'.

In summary, we define cognition as the process by which an autonomous self-governing agent acts effectively in the world in which it is embedded and that cognition has two functions: (1) to increase the agent'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.

Before concluding, it is worth noting that some authors in discussing the development of cognitive systems go even further than what we have discussed so far. For example, Brachman [22] defines a cognitive computer system as one which — in addition to being able to reason, to learn from experience, to improve its performance with time, and to respond intelligently to things it's never encountered before — would also be able to explain what it is doing and why it is doing it. This would enable it to identify potential problems in following a current approach to carrying out a task or to know when it needed new information in order to complete it. Hollnagel [94] suggests that a cognitive system is able to view a problem in more than one way and to use knowledge about itself and the environment so that it is able to plan and modify its actions on the basis of that knowledge. Thus, for some, cognition also entails a sense of self-reflection in addition to self-development. We see here cognition straying into the domain of consciousness. We won't say anything further in this course on the consciousness apart from remarking that computation modelling of consciousness is an active area of research.

1.3 Emulation or Simulation?

Before going any further, we need to be clear exactly what we are trying to do in attempting to develop an artificial cognitive system. Are we trying to develop an artifact that emulates the cognitive behaviour and capabilities of human beings, or are we trying to simulate the actual process by which a human being effects such behaviour and capabilities. The distinction, which is really the re-appearance of the complicated boundary between biological and computer cognition, is important because in the case of emulation it is only the end product — the system behaviour — that is crucial. On the other hand, in the case of simulation, it is necessary to be as faithful as possible to the human or biological process which underpin cognition. This doesn't mean that in the case of emulation we should ignore completely the biological systems. On the contrary, we need to look somewhere for inspiration in trying to deal with this very complex topic and biological systems are the only exemplars of cognition we have. Clearly, we should draw as much inspiration as possible from them and from what is known about biological cognition. In this course, though, we assume that our task is to emulate cognitive capability. That is, we want to create artificial cognitive systems — systems with all the desirable attributes set out above — but we don't necessarily want to make any strong claims that the resultant models are either biologically plausible or that they are viable models of human cognition. That said, we do look to the constraints of biological plausibility to provide some guidance in our search for a model of artificial cognition.

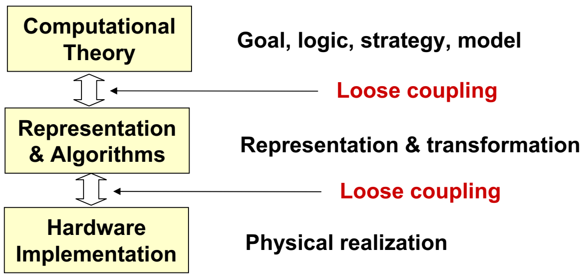


Fig. 1.2 The three levels at which a system should be modelled: the computational theory that formalizes the problem, the representational and algorithmic level that addresses the implementation of the theory, and the hardware level that phyically realizes the system (after [130]). The computational theory is primary and the system should be modelled at this level of abstration, although the representational and algorithmic level is often more intuitively accessible.

With that established, we need to decide at what level of abstraction we wish to draw on for our inspiration. That is, we need to find out the best level of abstraction of biological system models to use in developing the emulation (*i.e.* synthetic) model? There is some dissension in the scientific community about this.

For example, consider the classic work of David Marr [130] who advocated a three-level hierarchy of abstraction (see Figure 1.2). At the top level you have the computational theory. Below this there is the level of representation and algorithm. At the bottom there is the hardware implementation. The computational theory asks "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?". The representation and algorithm level asks "how can this computational theory be implemented? In particular, what is the representation for the input and output, and what is the algorithm for the transformation?" The hardware implementation asks "how can the representation and algorithm be realized physically?" Marr emphasized that these three levels are only loosely coupled. This dissociation is significant as it echoes the very strong assumptions made by proponents of a particular form of cognition (the cognitivist approach; see Chapter 2, Section 2.1). Marr states that, although the algorithm and representation levels are more accessible, it is the computational level that is critically important from an information processing perspective. In essence, he states that the problem can and should be modelled at the level of the computational theory without strong reference to the lower and less abstract levels. Marr illustrated this by pointing out that to understand bird flight, you need to address the theory of aerodynamics rather than the structure of feathers:

> "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

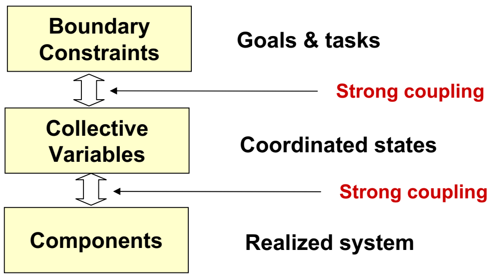


Fig. 1.3 The three levels at which a system should be modelled: a boundary constraint level that determines the task or goal, a collective variable level that characterizes coordinated states, and a component level which forms the realized system (after [105]). All three levels are equally important and should be considered together.

> have to understand aerodynamics; only then do the structure of feathers and the different shapes of birds' wings make sense"

Of course, you then have to decide how to realize the resulting computational model. Again, the point he was making is that you should decouple the different levels of abstraction, and begin your analysis at the highest level, allowing this subsequently to drive the decisions that need to be taken at the lower level when realizing the physical system.

In strong contrast, Kelso [105] argues that a system, specifically a non-linear dynamical systems of the type that may provide the basis for cognition, should be modelled simultaneously at three distinct levels. These are as follows (see Figure 1.3).

1. A boundary constraint level that determines the task or goals;
2. A collective variable level that characterizes coordinated states;
3. A component level which forms the realized system.

Kelso argues that the "boundary constraints, at least in complex biological systems, necessarily mean that the coordination dynamics are context or task dependent". Take away the context and you take away the basis for the model. Furthermore, the instantiation of the system has a direct role to play in the model itself (which is another way of saying that the system morphology matters and cannot be abstracted away). This is the essence of the perspective of an approach to modelling cognitive systems that has recently come to the fore. One cannot model the system in isolation from either the environmental context or its own physical instantiation. This issue will arise again later in the course when we look at the issue of embodiment in Chapter 3. We will also see these two different philosophies being reflected in the disparate positions on cognition to which we turn now in Chapter 2.