Chapter 1

Cognitive Architecture

Newell's Ultimate Scientific Question

On December 4, 1991 Allen Newell delivered his last lecture, knowing he was dying. Fortunately it was recorded[1]. I recommend it to anyone who wants to hear a great scientist explaining the simple but deep truths about his life as a scientist. For different people, different gems stand out from that talk, but the thing that stuck with me was his statement of the question that drove him. He set the context:

"You need to realize, if you haven't before, that there is this collection of ultimate scientific questions, and if you are lucky to get grabbed by one of these, that will just do you for the rest of your life. Why does the universe exist? When did it start? What's the nature of life? All of these are questions of a depth about the nature of our universe that they can hold you for an entire life and you are just a little ways into them."

Within this context, he announced that he had been so blessed by such a scientific question:

[1] *Desires and diversions* / Allen Newell; Carnegie Mellon University, School of Computer Science. Stanford, CA: University Video Communications, c1993. The portion of the lecture in question is available at our web site (act-r.psy.cmu.edu) The entire lecture is available at: http://wean1.ulib.org/cgi-bin/meta-vid.pl?target=Lectures/Distinguished%20Lectures/1991

"The question for me is, how can the human mind occur in the physical universe? We now know that the world is governed by physics. We now understand the way biology nestles comfortably within that. The issue is, how will the mind do that as well? The answer must have the details. I have got to know how the gears clank and how the pistons go and all the rest of that detail. My question leads me down to worry about the architecture."

When I heard these remarks from Newell, I heard what drove me as a cognitive scientist stated more clearly than I had ever been able to articulate myself. As Newell said, this question can hold you for a lifetime and you can only progress a little way towards the answer, but it is a fabulous journey. While Newell did much in his lifetime to make progress on the answer, I think he would be surprised by the developments since his death. For instance, we are now in a position where biology can really begin to inform our understanding of the mind. I can just see that enormous smile consuming his face if he had learned about these details. The purpose of this book is to report on some of the progress that has come from taking a variety of perspectives, including the biological.

While Newell did not come up with a final answer to his question, he was at the center of developing an understanding of what that answer would be like. The answer would be a specification of a **cognitive architecture** – "how the gears clank and how the pistons go and all the rest of that detail." The idea of a cognitive architecture did not exist when Newell entered the field, but it was well appreciated by the time he died. Because

Newell did more than anyone else to develop it, it is really his idea. It constitutes a great idea of science commensurate to the ultimate question of science that it addresses.

The purpose of this chapter is to describe what a cognitive architecture is, how the idea came to be, what the (failed) alternatives to it are, and to introduce the cognitive architecture around which the discussion in the remaining chapters will be organized.

What is a Cognitive Architecture?

"Cognitive Architecture" is a term used with some frequency in modern cognitive science (it is one of the official topics in the *Cognitive Science* Journal), but that does not mean that what it implies is obvious to everyone. Newell introduced the term cognitive architecture into cognitive science through an analogy to computer architecture (Bell and Newell, 1971), which Fred Brooks[2] (1962) introduced into computer science through an analogy to the architecture of buildings.

When acting in his or her craft, the architect neither builds nor lives in the house, but rather is concerned with how the structure (the domain of the builder) achieves the function (the domain of the dweller). Architecture is the art of specifying the structure of the building at a level of abstraction sufficient to assure that the builder will achieve the functions desired by the user. As his remarks at the beginning of his chapter "Architectural Philosophy" in *Planning a*

[2] Brooks managed the development of the IBM 360, which was a revolution at the time in the computer world. His perspective on computer architecture came from his experiences at IBM leading up to this.

Computer System indicate, this seems to be the idea that Brooks had in mind

"Computer architecture, like other architecture, is the art of determining the needs of the user of a structure and then designing to meet those needs as effectively as possible within economic and technological constraints." (p. 5)

In this passage, Brooks is using "architecture" to denote the activity of design, which in general usage probably constitutes its primary sense. However, computer architecture has come to mean the product of the design instead of the activity of it.. This was the sense used by Bell and Newell. It was in this sense that Newell introduced the term "cognitive architecture" into cognitive science, as can be seen in his 1990 definition: "the fixed (or slowly varying) structure that forms the framework for the immediate processes of cognitive performance and learning" (p. 111).[3]

This conception of cognitive architecture is found in a number of other definitions in the field:

Pylyshyn (1984): "The functional architecture includes the basic operations provided by the biological substrate, say, for storing and retrieving symbols, comparing them, treating them differently" (p. 30)

[3] Elsewhere, reflecting the history that led to this definition, Newell describes it as "what is fixed mechanism (hardware) and what is content (software) at the symbol level is described by the description of the system at the register-transfer level…. To state the matter in general: given a symbol level, the architecture is the description of the system in whatever system-description scheme exists next below the symbol level." (p.81)

Or my own rather meager definition:

Anderson (1983): "a theory of the basic principles of operation built into the cognitive system."[4] (p. ix)

It is worth reflecting on the relationship between the original sense of architecture involving buildings and this sense involving cognition. Figure 1.1 illustrates that relationship. Both sense of architecture involve relating a structure to a function:

Structure. The building's structure involves its physical components – its posts, fixtures, etc. None of the above definitions of cognitive architecture actually mentions its physical component – the brain – although Pyslyhyn's hints at it. While it would be strange to talk about a building's architecture at such a level of abstraction that one ignored its physical reality, one frequently finds discussions of cognitive architecture that simply do not mention the brain, . The definition at the end of this section, however, will make explicit reference to the brain.

Function: The function of building architecture is to enable the habitation, and the function of cognitive architectures is to enable cognition. Both habitation and cognition are behaviors of beings, but there is a difference in how they relate to the structure. In the case of a building, its function involves another agent: the dweller. In the case of cognitive architecture (or computer architecture), the structure is the agent[5]. Thus, there

[4] While my quoted definition predates the Newell dated definition, I know I got the term from discussions with him.

[5] One could get Platonic here and argue that "knowledge" is the agent occupying the cognitive architecture; then the analogy to physical architecture would be even closer.

is a functional shift from construction being designed to enable the activity of another to construction enabling its own activity. Except for this shift, however, there is still the same structure-function relationship. In both cases, an important measure of function is the success of the resulting behavior –building architecture is constrained to achieve successful habitation; cognitive architecture is constrained to achieve successful cognition[6].

Before the idea of cognitive architecture emerged, a scientist interested in cognition seemed to have two options -- either focus on structure and get lost in the endless details of the human brain (a structure of approximately 100 billion neurons), or focus on function and get lost in the endless details of human behavior. To understand the mind, we need an abstraction that gets at its essence. The cognitive architecture movement reflects the realization that this abstraction lies in understanding the relationship between structure and function rather than focusing on either individually. Of course, just stating the category of the answer in this way does not give the answer. There are major debates in cognitive science about what the best abstractions are for specifying a cognitive architecture.

With all this in mind, here is a definition of cognitive architecture for the purposes of this book:

A cognitive architecture is a specification of the structure of the brain at a level of abstraction that explains how it achieves the function of the mind.

[6] Although in one case the constraint is created by the marketplace and in the other case by evolution. I am aware that this discussion ignores aesthetic issues that influence the architecture of buildings.

Like any definition, this one relates one term, in this case cognitive architecture, to other terms. I suspect readers are going to wonder more about what the term "function of the mind" is in this definition than what the term "structure of the brain" is. The goal of a cognitive architecture is to provide the explanatory structure for better understanding both of these terms. However, before specifying such an architecture – and as some protection against misunderstanding – I'll note here that the "function of the mind" can be roughly interpreted as referring to human cognition in all of its complexity.

Alternatives to Cognitive Architectures

The type of architectural program that I have in mind requires paying attention to three things: brain, mind (functional cognition), and the architectural abstractions that link them. The history of cognitive science since the cognitive revolution has seen a number of approaches that tried to do with less; they can be viewed as shortcuts to understanding. This chapter will examine three of the more prominent instances of such shortcuts, discuss what they can accomplish, and note where they fall short of being capable of answering Newell's question. By looking at these shortcuts and what their problems are, we can better appreciate what the cognitive architecture program contributes when it attends to all three components.

Shortcut 1. Classic Information-Processing Psychology: Ignore the Brain

The first shortcut is the classic information-processing psychology[7] that ignored the brain. It was strongly associated with Newell and Simon, and one can argue that Newell never fully appreciated the importance of the brain in an architectural specification. In the decades immediately after cognitive psychology broke off from behaviorism, many argued that the answers it provided were at a level of abstraction that allowed it to ignore the brain. Rather than cite someone else for this bias, I will quote myself, although I was just parroting the standard party line:

"Why not simply inspect people's brains and determine what goes on there when they are solving mathematics problems? Serious technical obstacles must be overcome, however, before the physiological basis of behavior could be studied in this way. But, even assuming that these obstacles could be properly handled, the level of analysis is simply too detailed to be useful. The brain is composed of more than 10 billion nerve cells[8]. Millions are involved in solving a mathematics problem. Suppose we had a listing that explained the role of each cell in solving the problem. Since the listing would have to describe the behavior of individual cells, it would not offer a very satisfactory explanation for how the problem was solved. A neural explanation is too complex and detailed to adequately describe sophisticated human behavior. We need a level of analysis that is more abstract." (Anderson, 1980, pp 10 & 11)

The problem with this classic information-processing account is that it is like a specification of a building's architecture that ignores what the building is made out of.

[7] The modifier "classic" is being appended because "information processing" is used in many different senses in the field and I do not want it to appear as if this characterization applies to all senses of the term.
[8] This number has also come in for some revision.

Nonetheless, this type of account was very successful during the 1960s and 1970s. For example, the Sternberg task, and Saul Sternberg's (1966) model, was held up to my generation of graduate students as the prototype of a successful information-processing approach. In the prototypical **Sternberg paradigm,** participants are shown a small number of digits, such as "3 9 7," that they must keep in mind. They are then asked to answer – as quickly as they can – whether a particular probe digit is in this memory set. Sternberg varied the number of digits in the memory set and looked at the speed with which participants could make this judgment. Figure 1.2a illustrates his results. He found a nearly linear relationship between the size of the memory set and the judgment time, with each additional item adding 35-40 ms to the time. Sternberg also developed a very influential model of how participants made these judgments that exemplifies what an abstract information-processing model is like. Sternberg assumed that when participants saw a probe stimulus such as a 9, they went through the series of information-processing stages that are illustrated in Figure 1.2b. The stimulus first has to be encoded, then be compared to each digit in the memory set. He assumed that it took 35-40 ms to complete each of these comparisons. Sternberg was able to show that this model accounted for the millisecond behavior of participants under a variety of manipulations. Like many of those who created the early information-processing theories, Sternberg reached for the computer metaphor to help motivate his theory:

"When the scanner is being operated by the central processor it delivers memory representations to the comparator. If and when a match occurs a signal is delivered to the match register…." (p. 444)

From its inception, there were expressions of discontent with the classic information-processing doctrine. With respect to the Sternberg model itself, James Anderson wrote a 1973 *Psychological Review* article protesting that this model was biologically implausible in assuming that comparisons could be completed in 35 ms. It became increasing apparent that the computer-inspired model of discrete serial search failed to capture many of the nuances of the data (e.g., Glass, 1984; Van Zandt and Townsend, 1993). Such criticisms, however, were largely ignored until connectionism arose in the 1980s. Connectionism's proponents added many examples bolstering Anderson's general claim that processing in the brain is very different from processing in the typical computer. The connectionists argued that processing was different in brains and computers because a brain consists of millions of units operating in parallel, but slowly, whereas the typical computer rapidly executes a sequence of actions, and computers are discrete in their actions whereas neurons in the brain are continuous. The early connectionist successes, such as the Rumelhart and McClelland past-tense model, which will be described shortly, illustrated how much insight could be gained from taking brain processing seriously.

The rise of neural imaging in the 1990s has added more force to the importance of understanding the brain as the structure underlying cognition. Initially, researchers were simply fascinated by their newfound ability to see where cognition played out in the brain. More recently, however, brain-imaging research has strongly influenced theories of cognitive architecture. I will describe a number of examples of this influence. It has become increasingly apparent that cognition is not so abstract that our understanding of it can be totally divorced from our understanding of its physical reality.

Shortcut 2. Eliminative Connectionism: Ignore the Mind

As noted, one reason for dissatisfaction with the information-processing approach was the rise of connectionism and its success in accounting for human cognition through paying attention to the brain. Eliminative connectionism[9] is a type of connectionism that holds that all we have to do is pay attention to the brain -- just describe what is happening in the brain at some level of abstraction. Why include mental function as a constraint; why not just describe structure of the brain? Of course, that brain structure will generate the behavior of humans, and that behavior is functional. However, maybe it is just enough to describe the brain and get functional behavior for free from that description.

Eliminative connectionism is like claiming that we can understand a house just in terms of boards and bricks without understanding the function of its parts. This approach seems unlikely to yield useful results, approach and other metaphors reinforce skepticism – trying to understand what a computer is doing solely in terms of the activity of its circuitry without trying to understand the program that the circuitry is implementing, or indeed, trying to understand the other parts of the body just in terms of the properties of their cells without trying to understand their function. Despite the reasons for skepticism, this is just the approach of eliminative connectionism. Its goal is to come up with an abstract description of the computational properties of the brain – so-called "neurally inspired" computation – and then apply this description to explain various behavioral

[9] A term introduced by Pinker and Prince (1988) to describe connectionist efforts that eliminate symbols as useful explanations of cognitive processes, although here I am really using it to refer to efforts that ignore functional organization (how the pieces are put together).

phenomena. It is not concerned with how the system might be organized to achieve functional cognition. Rather, it assumes that cognition is whatever emerges from the brain's responses to the tasks it is presented and that any functionality comes for free – the house is what results from the boards and the carpenters, and if we can live in it, so much the better.

Eliminative connectionism has enjoyed many notable successes over the past two decades. The past tense model of Rumelhart and McClelland (1986) is one such success; I will describe it here as an exemplary case. Children show an interesting history (Brown, 1973) in dealing with irregular past tenses. For instance, the past tense of *sing* is *sang*. First, children will use the irregular correctly, generating *sang*; then they will overgeneralize the past-tense rule and generate *singed*; finally, they will get it right for good and return to *sang*. The existence of this intermediate stage of overgeneralization has been used to argue for the existence of rules, since it is argued that the child could not have learned from direct experience to inflect *sing* with *ed*. Rather, children must be overgeneralizing a rule that has been learned. Until Rumelhart and McClelland, this was the conventional wisdom (e.g., Brown, 1973), but it was a bit of a "just so story" as no one produced a running model that worked in this way.[10]

Rumelhart and McClelland (1986) not only challenged the conventional wisdom, they implemented a system that approximated the empirical phenomena by simulating a neural

[10] Actually, this statement is a bit ungenerous to me. I produced a simulation model that embodied this conventional wisdom in Anderson (1983), but it was in no way put into serious correspondence with the data. Although the subsequent past tense models are still deficient in various aspects of their empirical support, they do reflect a more serious attempt to ground the theories in empirical facts.

network illustrated in Figure 1.3 that learned the past tenses of verbs. In the network, one inputs the root form of a verb (e.g., kick, sing) as an activated set of feature units. After passing through a number of layers of association, the past-tense form (e.g., kicked, sang) should appear as another activated set of feature units. Their computer model was trained with a set of 420 pairs of root verbs with their past tenses. A simple neural learning system was used to learn the mapping between the feature representation of the root and the feature representation of the past tense. Thus, their model might learn (momentarily, incorrectly) that words beginning with "s" are associated with past tense endings of "ed," thus leading to the "singed" overgeneralization (but things can be much more complex in such neural nets). The model mirrored the standard developmental sequence of children: first generating correct irregulars, then overgeneralizing, and finally getting it right. It went through the intermediate stage of generating past-tense forms such as *singed* because of generalization from regular past-tense forms. With enough practice, the model, in effect, memorized the past-tense forms and was not using generalization. Rumelhart and McClelland concluded:

"We have, we believe, provided a distinct alternative to the view that children learn the rules of English past-tense formation in any explicit sense. We have shown that a reasonable account of the acquisition of past tense can be provided without recourse to the notion of a "rule" as anything more than a description of the language. We have shown that, for this case, there is no induction problem. The child need not figure out what the rules are, nor even that there are rules." (1986, p. 267)

Thus, they claim to have achieved the function of a rule without ever having to consider rules in their explanation. The argument is that one can understand function by just studying structure[11] and not constraining that structure to achieve the function. This original model is 20 years old and had shortcomings that were largely repaired by more adequate models that have been developed since (e.g., Plunkett and Joula, 1999; Plunkett and Marchand, 1993). Many of these later models are still quite true to the spirit of the original. This is still an area of lively debate, and Chapter 4 will describe our contribution to that debate.

However, the whole enterprise rests on a sleight of hand. This is not often noted, perhaps because many other models in cognitive science depend on this same slight of hand.[12] The sleight of hand becomes apparent if we register what the model is actually doing: mapping activation patterns onto activation patterns. It is not in fact engaged in anything resembling human speech production. Viewed in a quite generous light, the model is just a system that blurts out past tenses whenever it hears present tenses, which is not a common human behavior. That is, the model does not explain how, in a functioning system, the activation-input patterns get there, or what happens to the output patterns to yield parts of coherent speech. The same system could have been tasked with mapping past tenses onto present tenses – which might be useful, but for a different function. The model only seems to work because we are able to imagine how it could

[11] "Structure" here refers to more than just the network of connections; it also includes the neural computations and learning mechanisms that operate on this network.

[12] Our own ACT-R model of past tense (Taatgen and Anderson, 2002) is guilty of the same sleight of hand. It is possible to build such ACT-R simulations that are not end-to-end simulations but simply models of a step along the way. However, such fragmentary models are becoming less common in the ACT-R community.

serve a useful function in a larger system, or because we hook it into a larger system that actually does something useful. In either case, the functionality is not achieved by a connectionist system; it is achieved by our generous imaginations or by an ancillary system we have provided. So, basically in either case, it is we who have provide the function for the model, but we are not there to provide the function for the child. The child's mind must put together the various pieces required for a functioning cognitive system.

The criticism above is not a criticism of connectionist modeling per se, but rather a criticism of modeling efforts that ignore the overall architecture and its function. Connectionism is more prone to this error because its more fine-grained focus can lead to myopic approaches. Nonetheless, there are connectionist efforts that are concerned with full functioning systems (Smolensky and Legendre, 2006), striving to capture more of the overall flow of information processing in the brain (O'Reilly and Munakata, 2000). Especially, in the Smolensky and Legendre case, this reflects a conscious decision not to ignore function.

Shortcut 3. Rational Analysis: Ignore the Architecture

Another shortcut starts from the observation that a constraint on how the brain achieves the mind is that both the brain and the mind have to survive in the real world. Rather than focus on architecture as the key abstraction, focus on adaptation to the environment. I have called this approach rational analysis when I tried practicing it (Anderson 1990), but

it has been called other things when practiced by such notables as Egon Brunswik (1955 – probabilistic functionalism), James Gibson (1966 – ecological psychology), David Marr (1982 – computation level), and Roger Shepard (1984, 1987 – evolutionary psychology). More recent research in this spirit includes that of Nick Chater and Mike Oaksford (1999), Gerd Gigerenzer (Gigerenzer and Todd, 1999), and Josh Tenenbaum (Tenenbaum and Griffiths, 2001). My application of this approach was basically Bayesian, and more recent approaches have become even more Bayesian. Indeed, the Bayesian statistical methodology that accompanies much of this research has almost become a new Zeitgeist for understanding human cognition. Briefly, the Bayesian approach claims that

1. We have a set of prior constraints about the nature of the world we occupy. These priors reflect the statistical regularities in the world that we have acquired either through evolution or experience. For instance, physical objects in the universe tend to have certain shapes, reflectance properties, and paths of motion, and our visual system has these priors built into it.
2. Given various experiences, one can calculate the posterior probability that various states of the world gave rise to them. For instance, we can calculate the conditional probability of what falls on our retina given different states of affairs in the world.
3. Given the input, one can calculate the posterior probabilities from the priors (1) and conditional probabilities (2) . For instance, one can calculate what state of affairs in the world most likely corresponds to what falls on our retina.

4. After making this calculation, one engages in Bayesian decision-making and takes the action that optimizes our expected utilities (or minimizes our expected costs). For instance, we might duck if we detect information that is consistent with an object coming at our head. Anderson (1990) suggested that at this stage, knowledge of the structure of the brain could come into play in computing the biological costs of doing something.

The Bayesian argument claims neither that people explicitly know the priors or the conditional probabilities nor that they do the math explicitly. Rather, we don't have to worry about how people do it; we can predict their cognition and behavior just from knowing that they do it somehow. Thus, the Bayesian calculus comes to take the place of the cognitive architecture.

I regard the work I did with Lael Schooler on memory as one of the success stories of this approach (Anderson and Schooler, 1991; Schooler and Anderson, 1997). We looked at how various statistics about the appearance of information in the environment predicted whether we would need to know the information in the future. Figure 1.4 shows an example related to the retention function (the probability and speed of remembering something). Part (a) of that figure shows how the probability that I will receive an email message from someone on a day varies as a function of how long it has been since I last received an email from that person. So, for example, if I received an email message from someone yesterday, the probability is about 30% that I will receive one from him or her on today. However, if it has been 100 days since I last received an email message from

that person, the probability is only about 1% that I will receive one from him or her today. Figure 1.4a shows a rapid drop off indicating that if I have not heard from someone for a while, it becomes very unlikely that I will again. Anderson and Schooler found this same sort of function showing up for repetition of information in all sorts of environments. It reflects the demand that the world makes on our memory. For instance, when I receive an email message, it is a demand on my memory to remember the person who sent it.

If the brain chose which memories to make most available, it would make sense to choose the memories that are most likely to be needed. Figure 1.4a indicates that time since a memory was last used is an important determinant of whether the memory will be needed now. Anderson and Schooler did the Bayesian math to show that this temporal determinant implied that retention functions should show the same form as environment functions such as Figure 1.4a, and they do, as Figure 1.4b shows in the classic retention function obtained by Ebbinghaus. Thus, a memory for something diminishes in proportion to how likely people are to need that memory. We showed that this was true not only for retention functions, but also for practice functions, for the interaction between practice and retention, for spacing effects, for associative priming effects, and so on. Human memory turned out to mirror the statistical relationship in the environment in every case. As described in Chapter 3, we discovered a relationship between retention and priming in the environment that had never been tested in human memory. Schooler did the experiment and, sure enough, it was true of human memory (Schooler and Anderson, 1997). Thus, the argument goes, one does not need a description of how

memory works, which is what an architecture gives; rather, one needs to focus on how memory solves the problems it encounters. Similar analyses have been applied to vision (Karlin and Lewicki, 2005), categorization (Anderson, 1991b; Tenenbaum, 1997; Sanborn, Griffiths, and Navarro, 2006), causal inference (Griffiths and Tenenbaum, 1995), language (Pickering and Crocker, 1996), decision making (Bogacz et al., 2006), and reasoning (Oaksford and Chater, 1994).

While I was an advocate of this approach, I started to realize (e.g., Anderson, 1991a) that it would never answer the question of how the human mind can occur in the physical universe. This is because the human mind is not just the sum of core competences such as memory, or categorization, or reasoning. It is about how all these pieces and other pieces work together to produce cognition. All the pieces might be adapted to the regularities in the world, but understanding their pattern of adaptation does not address how they are put together.

In many cases, the rational analyses (e.g., vision, memory, categorization, causal inference) have characterized features of the environment that all primates (and perhaps all mammals) experience.[13] Actually, many of these adaptive analyses were inspired by research on optimal foraging theory (Stephens and Krebs, 1986), which is explicitly pan-species in its approach. The universal nature of these features raises the question of what enables the human mind in particular.[14] Humans share much with other creatures

[13] Schooler has done unpublished analyses of primate environments.
[14] While there have been some interesting analyses of how the statistics of the language affect language learning and language use (e. g., Newport and Aslin, 2004), exposing a non-human primate to these statistics does not result in language processing capability.

(primates in particular), so these analyses have much to contribute to understanding humans, but something is missing if we stop with them. There is a great cognitive gulf between humans and other species, and we need to understand the nature of that gulf. What distinguishes humans is their ability to bring the pieces together, and this unique ability is just what adaptive analyses do not address, and just what a cognitive architecture is about. As Newell said, you have to know how the gears clank and how the pistons go and all the rest of that detail.

ACT-R: A Cognitive Architecture

It was basically a rhetorical ploy to have postponed giving an instance of a cognitive architecture until now. Many instances of cognitive architecture exist, including connectionist architectures[15]. Newell was very committed to an architecture called Soar, which has continued to evolve and grow since his death (Newell, 1990 – see http://sitemaker.umich.edu/soar for current developments in Soar).

A different book could have included a comparison of different cognitive architectures, but such comparisons are already abundant in the literature (e.g., National Research Council, 1998; Ritter et al., 2003; Taatgen and Anderson, in press). The goal of this book is not to split hairs about the differences among architectures, but to use one to try to convey what we know that is true about the human mind. For this purpose, I will use the ACT-R architecture (Anderson et al, 2004) because I know it best. However, this book

[15] Just do a search on "connectionist architecture" in Google.

is not about ACT-R; rather, I am using ACT-R as a tool to describe the mind. Like the architect's drawings are tools to connect structure and function, the ACT-R models in this book are being used to connect brain and mind. We may be proud of our ACT-R models and think they are better than others in just the way architects are proud of their specifications, but we try not to loose track of the fact that they are just a way of describing what is really of interest.

ACT-R has a history (discussed in the Appendix) going back 30 years to the HAM theory and early ACT theories. ACT-R emerged in 1993 (Anderson, 1993) when I realized the inadequacy of rational analysis, but the R stands for "rational" to reflect the influence of rational analysis. Today ACT-R is the product of a community of researchers who use it to theorize about cognitive processes. There is an ACT-R web site (http://act-r.psy.cmu.edu/) that you can visit to read about example models or to consult the users manual and tutorial for the simulation system that specify the details of the architecture. (A computer simulation of the architecture has been developed that allows us to work out precisely what ACT-R models predict about human cognition.) . Having this documentation on the web allows me to focus here on core ideas about human cognition, and to develop them in detail in later chapters. The goals of the remainder of this chapter are to briefly describe ACT-R as an illustration of a cognitive architecture, to show how an architecture can be connected to the results of brain imaging. and to use ACT-R as a context for discussing contentious issues in cognitive science regarding the status of symbols.

ACT-R'S Modular Organization

Figure 1.5 illustrates the ACT-R architecture as it appeared in Anderson (2005). In this architecture, cognition emerges through the interaction of a number of independent modules. Anderson (2005) was concerned with how the ACT-R system applied to the learning of a small fragment of algebra. The five modules in Figure 1.5 were those[16] used in the model I developed of algebra learning:

1. A visual module that might hold the representation of an equation such as "3x – 5 = 7".
2. A problem state module (sometimes called an imaginal module) that holds a current mental representation of the problem. For example, the student might have converted the original equation into a mental image of "3x =12."
3. A control module (sometimes called a goal module) that keeps track of one's current intentions in solving the problem. For example, one might be trying to perform an algebraic transformation.
4. A declarative module that retrieves critical information from declarative memory, such as that 7 + 5 = 12.
5. A manual module that programs the output, such as "x = 4"

Each of these modules is associated with specific brain regions; ACT-R contains elaborate theories about the internal processes of these modules. Later chapters explore the specifics of some of these modules, which must communicate among each other; they do so by placing information in small-capacity buffers associated with them. A central procedural system (a sixth module) can recognize patterns of information in the buffers

[16] The next chapter will discuss all eight modules that are currently part of ACT-R.

and respond by sending requests to the modules. These recognize-act tendencies of the central procedural module are characterized by production rules. E.g., the following is a description of a possible production rule in the context of solving algebraic equations such as 3x –5 = 7:

If the goal is to solve an equation
    And the equation is of the form "expression - number1 = number2"
THEN write "expression = number2 + number1"[17]

where the first line refers to the goal buffer, the second line to the visual buffer, and the third line to a manual action.

Anderson (2005) describes a detailed model of learning to solve simple linear equations (such as 3x –5 = 7) that was used to understand the data from an experiment (Qin et al, 2004) involving children aged 11-14. They were proficient in the middle-school prerequisites for algebra, but they had never before solved equations. During the experiment, they practiced solving such equations for 1 hour per day for 6 days. The first day (Day 0) they were given private tutoring on solving equations; on the remaining 5 days, they practiced solving three classes of equations on a computer:

0-step: e.g., 1x+0=4

1-step: e.g., 3x+0=12, 1x+8=12

2-step: e.g., 7x+1=29

Figure 1.6 shows how the time required by the children to process these equations diminished over the course of the experiment.

[17] This rule is hypothetical, used for illustration; consult Anderson (2005) for more accurate details.

Figure 1.6 also illustrates the predictions of a model implemented in the ACT-R architecture. The model is not programmed to do the task; instead it starts with declarative representations of the instructions that the children receive and has general production rules for following any set of instructions. It also has a virtue that can be achieved by a system built in a full cognitive architecture -- it does the entire task transparently, from the appearance of the equation on the screen to the pressing of the keystroke (unlike past tense models that model a small fraction of the task and leave to the imagination how that fraction results in functional behavior). We sometimes call this a model of **end-to-end behavior**.

The model, like the participants, took longer with more complex equations because it had to go through more cognitive steps. More interestingly, it improved gradually in task performance at the same rate as participants: the effect of 6 days of practice was to make a 2-step equation like a 1-step equation in terms of difficulty (as measured by solution time) and a 1-step equation like a 0-step equation; Anderson (2005) describes the detailed processing. The critical factors in learning to solve equations will be considered in Chapter 5. However, for current purposes, Figure 1.7 (taken from Anderson (2005)) illustrates the detailed processing involved in solving the 2-step equation $7x + 3 = 38$ on the first (Figure 1.7a) and fifth (Figure 1.7b) days of the experiment. In the figure, the passage of time moves from top to bottom and different columns represent the points in time at which different modules were active. This can be seen as just a great elaboration of the Sternberg stage model (Figure 1b) in which stages include activities in multiple

modules that can be simultaneously active. The primary reason the model requires less time on day 5 than on day 1 is a reduction in the amount of information the declarative module is called upon to retrieve. This is clear in the comparison between the amounts of activity in the retrieval columns in parts (a) and (b) of Figure 1.7. As will be elaborated on in Chapters 3 and 4, this is due to increased speed of individual retrievals and because retrieval of instructions is replaced by production rules specific to algebra.

Brain Imaging Data and the Problem of Identifiability

The complexity of Figure 1.7 relative to the simplicity of the behavioral data in Figure 1.6 reflects a deep problem that has seriously handicapped efforts to develop cognitive architectures. A very complicated set of information-processing steps is required to go from instruction on algebra and the presentation of an algebraic equation to the actual execution of an answer. No matter how one tries to do it, if the attempt is detailed and faithful to the task, the resulting picture is complicated like Figure 1.7. However, while we know the process is complicated, it does not necessarily follow that those complicated steps are anything like Figure 1.7 in terms of the modules involved or exact sequences of operations. Working with standard behavioral data, the only way a cognitive modeler had to tell whether his or her model was correct was whether it matched data such as Figure 1.6's. But such data do not justify all of this detail.

In Anderson (1990) I showed that given any set and any amount of behavioral data, there would always be multiple different theories of internal process that produced that data. I

concluded, “it is just not possible to use behavioral data to develop a theory of the implementation level in the concrete and specific terms to which we have aspired” (p. 24). This was part of my motivation for developing the rational approach. In 1990 a diagram such as Figure 1.7 would be as much my fantasy about what was going on as it would be fact. However, I did acknowledge that physiological data would get us out of this identifiability dilemma. I claimed that “the right kind of physiological data to obtain is that which traces out the states of computation of the brain,” because this would provide us with “one-to-one tracing of the implementation level.” I noted the progress that the pioneers of brain imaging had already made by 1990.

While the field is not altogether there yet in 2007, it is much closer to having what is needed to base a diagram such as Figure 1.7 on fact rather than fantasy. In our lab we have been mainly working with fMRI (functional magnet resonance imagery) brain imaging data. The next chapter will include an up-to-date report of the connections we have made between modules of ACT-R and activity in specific brain regions, but this chapter provides a taste of material illustrating that it is possible to map some of the detail in Figure 1.7 onto precise predictions about brain regions.

The children whose behavioral data were reported in Figure 1.6 were scanned on days 1 and 5 in an fMRI scanner. The details of the study and derivation of predictions from Figure 1.7 are available in Anderson (2005); Figure 1.8 summarizes the predictions and results for 5 brain regions. These regions are not cherry-picked for this one study; they

are the same regions examined in study after study because they are associated with specific modules in the ACT-R theory.

Predicting the BOLD Response in Different Brain Regions

Figure 1.8a illustrates the simplest case, which is the motor module. The representation of the hand along the motor strip is well known, and there is just a single use of this module on each trial to program the response. The x-axis presents time from the onset of the trial[18]. The data in Figure 1.8 show the increase from base line in the BOLD (blood oxygen level dependent) response in this region. The top graph shows the BOLD response for different numbers of operations (averaging over days). The three BOLD functions are lagged about 2 seconds apart, just as the actual motor responses are in the three conditions. However, as typical of BOLD functions, they slowly rise and fall, reaching a peak 4 to 5 seconds after the key press. The bottom graph in the figure compares the BOLD response on days 1 and 5 (averaging over the number of transformations). Basically, the response shifts a little forward in time from day 1 to day 5, reflecting the speed increase. The predictions are displayed as solid lines in the figure and provide a good match to the data. As detailed in the next chapter, these predictions are generated according to when the module is active. Whenever a module is active, it creates extra metabolic demand in its associated brain region, which drives a larger BOLD signal. In the case of the manual module, the activity and metabolic demand

[18] The first 1.2 seconds involved presentation of a warning signal before the equation was presented. The data in Figure 1.6 are from presentation of the equation.

happen at the end of the charts in Figure 1.7. Figure 1.8a illustrates the ability of this methodology to track one component in an overall task

Unlike the motor module, the other modules are used sporadically through the solution of the problem rather than just at the end (see Figure 1.7). Because the BOLD response tends to smear close-by events together, it is not possible in this experiment to track the timing of a specific step in these other modules. Nonetheless, we can generate and test distinct predictions for these regions.

We have associated a prefrontal region (see Figure 1.8b) with retrieval from the declarative module. In contrast to the motor region, in this prefrontal region there are very different magnitudes of response for different numbers of operations in the top graph. This is as predicted, since more transformations mean that more instructions and mathematical facts need to be retrieved to solve the equation. A distinguishing feature of this region is the very weak response it generates in the case of 0 steps. According to the model, this case involves some brief retrievals of instructions but no retrieval of number facts, which is why the response is so weak. As noted earlier, the major reason for the speed increase across days is that the number of retrievals decreases and the time per retrieval speeds up. Therefore, the reduction is predicted in the BOLD response in going from Day 1 to Day 5 in the bottom graph in Figure 1.8b.

We have associated a region of the anterior cingulate cortex (see Figure 1.8c) with the control function of the goal module. As in the prefrontal region, there is a large effect of

number of operations in the top graph because the model has to go through more control states when there are more transformations. In contrast to the prefrontal region, however, in the anterior cingulate cortex there is a robust response even in the 0 step case, because it is still necessary to go through the control states governing the encoding of the equation and the generation of the response. The striking feature of the anterior cingulate is that there is almost no effect of learning in the bottom graph. The effect of practice is largely to move the model more rapidly through the same state changes, and so there is little change in the number of control states. Therefore, little effect is predicted for number of days.

For the sake of brevity, I will skip discussion of the other two regions (the parietal in Figure 1.8d associated with the imaginal module, and the caudate in Figure 1.8e associated with the procedural module), except to note that they display a pattern similar to one another but different from any of the other regions. Details can again be found in Anderson (2005), as well as evidence of just how good the match up is between prediction and data. The fact that we can obtain and predict four different patterns of activation across the same conditions shows the power of imaging to go beyond the latency data displayed in Figure 1.6. And if the suspicious reader is wondering just how good the match up is between prediction and data, please go to Anderson (2005).

The rest of the book will be concerned in great detail with the properties of these specific regions and their associations with ACT-R modules. We will discuss the similarities and

differences between the ACT-R interpretation of these regions and other interpretations in the literature. Unless you are quite familiar with this research, the similarities among the theories will seem much greater than the differences. There is convergence in the literature on the interpretation of the functions of these various brain regions.

Summary

For the purposes of this chapter, consider how the ACT-R architecture avoids the pitfalls of the shortcuts that have been reviewed:

1. Unlike the classic information-processing approach, the architecture is directly concerned with data about the brain. While brain imaging data have played a particularly important role in my laboratory, data about the brain more generally have been influential in the development of ACT-R.
2. Unlike eliminative connectionism, an architectural approach is also focused on how a fully functioning system can be achieved. Within the ACT-R community, the primary functional concern has been with the mathematical-technical competences that define modern society.[19] The final chapter of this book will elaborate extensively on what algebra problem solving reveals as unique in the human mind

[19] However, the reader should not think this is all that has been worked on. The ACT-R web site displays the full range of topics on which ACT-R models have been developed.

3. Unlike the rational approach and some connectionist approaches, ACT-R does not ignore issues about how the components of the architecture are integrated. Indeed, ACT-R is more a theory about that integration than anything else.

Symbols versus Connections in a Cognitive Architecture

The Debate

There is a great debate in cognitive science between architectures that are called symbolic and architectures that are called connectionist, and ACT-R has been reluctantly placed on one side of this debate. I would rather skip that to get on with the story, but the debate is too notorious to just ignore.[20] . While it is not a commonly held characterization among members of the ACT-R community, many members of the larger cognitive science community tend to regard ACT-R as an instance of a symbolic architecture.[21] The connectionist past-tense models described earlier did not gather so much attention simply by accomplishing what had not been done before. Rather, they were magnets for attention in the cognitive science community because of statements like the one quoted earlier that claimed to have done away with symbols. These connectionist efforts claimed to have shown fundamental inadequacies in "symbolic" architectures such as ACT-R. There has been no lack of people willing to join the debate on the symbolic side (e.g., Fodor and Pylyshyn, 1988; Pinker and Prince, 1988; Marcus, 2001). It was a particular virtue of Newell that he never engaged in this debate, even though others had placed him on the symbolic side of the world (and he certainly did believe in symbols).

[20] Perhaps this is why I put this off to the last topic in the chapter.
[21] I received the Rumelhart prize in 2005 as "the leading proponent of the symbolic modeling framework". While I was very honored by the prize, I have to confess the characterization left a craw in my throat.

Some fraction of the controversy is really a debate about the language to describe cognition, rather than about scientific claims. This debate turns on the word "symbol" – a word that enjoyed a happy existence in the English language until the advent of cognitive science. In the good old days, symbols were physical objects (usually visual representations -- for instance, the cross as the symbol for Christianity) that were used to stand for or designate something else. There were good symbols and bad symbols (in many senses of the words "good" and "bad"), but nobody would think to debate whether symbols, per se, were good or bad. Among these symbols were the symbols of mathematicians and logicians. Among these mathematicians and logicians were people such as Church, Turing, Goedel, Post, and von Neumann who noted that computation could be achieved by operations on such symbols – hence the emergence of the idea of symbol manipulation. With the appearance of real computers, individuals such as McCarthy who were heavily influenced by this logical background created symbol manipulation languages such as LISP that formed the backbone of early artificial intelligence. By this time the "symbol" in cognitive science had only a loose connection to its original meaning.

There is a lack of consensus about whether symbols in cognitive science maintain the referential feature of original symbols – i.e., they stand for something. Newell and Simon (1976) explicitly state that symbols designate other things. Nonetheless, they extend the notion of symbols to pointers in data structures, which can have no reference to anything external to the data structure itself. Pointers really derive their meaning from

the structures and processes in which they participate; they do not have external reference. Nonetheless, the idea that symbols have reference continues in discussions. For instance, Vera and Simon (1993) assert that "we call patterns symbols when they can designate or denote" (p. 9). On the other hand, one finds people such as Searle (1980) and Lakoff (1988) talking about "meaningless symbol manipulation." Searle, focusing on their physical appearance, refers to them as "meaningless squiggles." Harnad (1990), in describing what he calls the symbol-grounding problem**,** asks, **"**How can the meanings of the meaningless symbol tokens, manipulated solely on the basis of their (arbitrary) shapes, be grounded in anything but other meaningless symbols?"

Given this lack of agreement on what symbols are, it should come as no surprise that there is no consensus about what role symbols play in an explanation of the mind and how they should be coordinated with knowledge of brain processing. The positions can be classified according to whether they give an explanatory role to symbols or connections. These are enumerated below with a "+" to indicate an explanatory role and a "-" a non-explanatory role.

1. +Symbols, - Connections: The Classic Symbol Manipulation Position holds that the principles by which the mind operates involve transformations of structural properties of symbolic representations. This is the position that symbols are like the symbols that appear in LISP (which are basically pointers and, as noted, can be almost devoid of any sense of external reference). The claim is that, while the mind is not a LISP program, symbols play the same critical role in the explanation of mind as they do in a LISP

program. There are two sub-traditions -- the linguistic tradition, represented by Chomsky and Fodor, and the information-processing tradition, represented by Newell and Simon. This position has threads in common with the information-processing shortcut described earlier and tends to regard the physical processes that realize these symbols as unimportant.

2. -Symbols, +Connections: Eliminative Connectionism views symbols much like elements in explicitly stated rules ("If the verb ends in d or t, add ed") and regards such assertions about the mind as, at best, good approximate descriptions of brain computations and, at worst, misleading. This position is called eliminative connectionism because it seeks to eliminate symbols in the explanation of cognition. This position sees no explanatory role for symbols, just as the classic position sees no explanatory role for the brain.

3. +Symbols, + Connections: Implementational Connectionism believes that connectionist computations are organized to achieve symbolic results and that both connectionist and symbolic characterizations play an important explanatory role (e.g., Shastri and Ajjanagadde, 1993; Smolensky, 1995). One way or another, this view assumes that connectionist computations implement symbolic computations. For instance, in Smolensky and Legendre's (2006) Integrated Connectionist/Symbolic Architecture (ICS), connectionist calculations can serve to enforce a hierarchy of symbolic constraints on grammatical selections. For Smolensky and Legendre, with their emphasis on linguistic applications, the symbols are basically the kinds of terms that

appear in classic linguistic models such as "verb phrase" or "stressed."

4. **-Symbols, -Connections:** Some researchers have rejected both symbols and connections as explanations. In their place, other explanatory devices are offered, or the possibility of explaining the human mind is simply rejected. Historically, functionalism and some varieties of behaviorism, such as that of Skinner, had this characteristic. More recently, some versions of adaptive explanations (see earlier discussion of rational analysis) have emphasized the explanation as totally residing in the environment. Differing slightly in their emphasis, some versions of situated cognition (e.g., Lave, 1988; Lave and Wenger, 1991; Greeno, Smith, and Moore, 1992) have also emphasized that the explanation resides in what is outside the human.[22]

In my opinion, debates among these positions have the character of jousting with windmills. Because there is not even agreement about what symbols mean, these debates are a waste of time.

The Symbolic-Subsymbolic Distinction

However, I cannot simply reject all discussion of symbols and use the ACT-R architecture, because that architecture makes a distinction between what it calls "symbolic" and "subsymbolic" levels. [23] These bear only partial relationships to the

[22] One might also include dynamical systems (e.g., Thelen and Smith, 1994; van Gelder, 1998) in this category as Clark (1997) suggests, but at least some practioners (e.g., Smith and Samuelson, 2003) of this approach have argued that their battle with the greater common enemy (the classic symbol manipulation approach) means that the connectionist and dynamic systems approach are really complementary.

[23] Some apology is in order for having introduced these terms into the theory, I suppose. It happened as we attempted to describe an important distinction in a way that we thought would be meaningful to the

terms of the debate about symbols versus connections. The symbolic level in ACT-R is an abstract characterization of how brain structures encode knowledge. The subsymbolic level is an abstract characterization of the role of neural computation in making that knowledge available. The following discussion of symbols by Newell captures the essence of the symbolic level as we use it in ACT-R and sets the context for also understanding ACT-R's subsymbolic level:

"Symbols provide distal access to knowledge-bearing structures that are located physically elsewhere within the system. The requirement for distal access is a constraint on computing systems that arises from action always being physically local, coupled with only a finite amount of knowledge being encodable within a finite volume of space, coupled with the human mind's containing vast amounts of knowledge. Hence encoded knowledge must be spread out in space, whence it must be continually transported from where it is stored to where processing requires it. Symbols are the means that accomplish the required distal access." (Newell, 1990, p. 427)

Newell identifies the critical role of symbols as knowledge access; there is no mention in this quote of the popular image of symbol manipulation with its juggling of symbols, nor is there any commitment to whether symbols refer. He notes that most computation is local (true of the brain with its hypercolumns and the like), but information must be brought from other locations to influence the local processing (again true of the brain with its fiber tracks). Symbols for Newell provide this distal access. This is exactly what

---

cognitive science community. We were not thinking deeply about what the words meant to us or what they really meant (or did not mean) in the cognitive science community.

they do in ACT-R; one might identify them with fiber tracks in the brain.

While symbols provide distal access so that information can be brought from one location to another, there is the question of just what information will be brought and how quickly that information will appear. This is what the subsymbolic level is about. Symbolic structures have subsymbolic quantities associated with them that control how fast they are processed and which units get processed at choice points.[24] This symbolic-subsymbolc relationship reflects a very general theoretical approach in science to postulate objects with real-value quantities – habits with strengths in Hull's theory, units with activations and link strengths in connectionism, or electrons with energy levels.

The symbolic-subsymbolic distinction has been developed extensively for two modules in ACT-R, the declarative and procedural modules.

The Symbolic-Subsymbolic Distinction in the Declarative Module

With respect to the declarative module at the symbolic level, ACT-R has networks of knowledge encoded in what we call **chunks**. Figure 1.9 illustrates a declarative chunk encoding a fact from the Berry and Broadbent (1984) sugar factory task. This structure connects an event in that task: a factory had produced 10,000 tons of sugar in the previous month, 800 workers were assigned to the factory in the current month, and

[24] However, be aware that this ACT-R use of "subsymbolic" to designate the numbers under the symbols is not the same as the more standard use of "subsymbolic" to refer to the connectionist elements, which are at a finer grain size than symbolic units. The "sub" in the more common usage can be read as "pieces of symbols," whereas in our usage it is the numbers "under the symbols."

7,000 tons of sugar was produced in the current month. Figure 1.9 illustrates the connections that provide Newell's distal access. Thus, a query such as, "If the past production was 10,000 tons and I use 800 workers, how many tons will I get now?" can make contact with the answer of 7,000 tons.

However, what if there were multiple chunks stored with different current output associated with 10,000 tons in the past and 800 workers? What if there was no chunk with the answer for this exact query? One needs to specify the neural processes by which an appropriate chunk is selected as an answer. As Chapter 3 will develop, chunks have **activations** at the subsymbolic level. The most active chunk will be the one retrieved, and its activation value will determine how it is retrieved. The activation values of chunks are determined by computations that attempt to abstract the impact of neural Hebbian-like learning and spread of activation among neurons. Chapter 3 will review some of the successes of this mechanism in capturing many aspects of human cognition, including performance in the Berry and Broadbent sugar factory task.

The Symbolic-Subsymbolic Distinction in the Procedural Module

As already noted, the procedural module consists of production rules.[25] Figure 1.10

[25] While there is a widely felt discontent with "symbols" and their connotations, there is an evenly more widely felt discontent with 'rules" and their connotations. I have encountered it not only from connectionists, but also from many mathematics educators. I have been advised that ACT-R would have greater appeal if I just did not use the phrase "production rule" but instead something like "action selection." Perhaps such a name switch would more accurately reflect what the rules (or the mappings) do, but I fear such

illustrates a production rule that might apply in solving the equation 3 + x = 8. Part (a) of Figure 1.10 is an instantiation of the rule for this specific equation. The rule responds to a pattern that appears in a set of modules – in this case, to the encoding of the equation by the visual module and the setting of the control state in the goal module to solve that equation. An action is selected that requests the retrieval from declarative memory of the difference between 8 and 3 and sets the control state to note a subtraction is occurring. As we will discuss throughout the book, it is generally thought that the basal ganglia play a critical role in achieving this pattern recognition, action selection, and action execution.

Part (b) of Figure 1.10 illustrates the general rule that is behind the instance in part (a). The rule is not specific to the numbers 3 and 8. Whatever number appears in the arg1 slot of the visual buffer is copied to the arg2 slot on the declarative retrieval request. Similarly, whatever number appears in the arg2 slot of the visual buffer is copied to the arg1 slot of the retrieval request. Thus, this production is a pattern that specifies how information is to be moved from one location to a distal location. This is symbolic exactly in the distal access sense of the Newell quote above.

There are situations (developed in Chapter 4) where multiple production rules might apply and the decisions about which rule to apply are determined at the subsymbolic level, where production rules have **utilities** and the production with the highest utility is chosen. The utilities of productions are determined by computations that are designed to abstract the essential aspects of the neural reinforcement learning that determines action

---

a name switch now would engender new confusions even greater than the ones it might eliminate.

selection.

Final Reflections on the Symbolic-Subsymbolic distinction

While the structures in Figures 1.9 and 1.10 are symbolic in the Newell sense, it is hard to see how anything in them is symbolic in the sense of "meaningless squiggles" or "ungrounded meaningless symbols" or in the sense of "denoting something." Nothing in the production rule in Figure 1.10 is fundamentally different than the pattern-matching capabilities of standard connectionist networks, and indeed we created a connectionist implementation of an early version of ACT-R (Lebiere and Anderson, 1993). The links in Figures 1.9 and 1.10 simply represent the kinds of connections seen in any neural model, albeit at a higher level of abstraction.

It is true that when one looks at the actual code that specifies a model for purposes of simulating it, one will see things that look like the cognitive science stereotype of a symbol as a piece of text. Consider the specification of a set of chunks in Table 1.1a for the ACT-R simulation program, and compare this with the specification of a connectionist network in Table 1.1b. There is the tendency to confuse the notation of either specification with "symbols." They are perhaps symbols for the simulation program, but they are not the symbols of the ACT-R architecture or the connectionist network.[26] The ACT-R specification uses the word "workers" and the connectionist specification uses the word "digits," but in both cases these are just mnemonic labels to

[26] Actually they are largely not manipulated by the simulation program either, but are notation about how to compile the simulation into code that "just does it."

help the person read the code. Neither model's behavior would change if some random sequence of letters were substituted instead. Much of the debate about symbols reflects confusion between notation and theory. Of course, the graphic representations in Figures 1.9 and 1.10 are just notations too. However, this book will tend to use such graphic notations because they tend to better convey the theoretical claims.

The reader may still feel there is some significant difference between the ACT-R specification in Table 1.1a and the connectionist specification in Table 1.1b. There is, and it is a difference in the level of abstraction at which the theory is specified. It is a strategic decision in science as to what is the best level of abstraction for developing a theory. In the case of connectionist elements or symbolic structures in ACT-R, the question is which level will provide the best bridge between brain and mind and thus answer Newell's question. In both cases, the units are a significant abstraction from neurons and real brain processes, but the gap is probably smaller from the connectionist units to the brain. Similarly, in both cases the units are a significant distance from functions of the mind, but probably the gap is smaller in the case of ACT-R units. In both cases, the units are being proposed to provide a useful island in building a bridge from brain to mind. The same level of description might not be best for all applications. Connectionist models have enjoyed their greatest success in describing perceptual processing, while ACT-R models have enjoyed their greatest success in describing higher-level processes such as equation solving.

To return to the title of this chapter and the book, the function of a cognitive architecture is to find a specification of the structure of the brain at a level of abstraction that explains how it achieves the function of the mind. I believe ACT-R has found the best level of abstraction for understanding those aspects of the human mind that separate it from the minds of other species. The rest of the book will develop the key aspects of this architecture. Chapter 5, in particular, will address the question of how human minds can occur.

Appendix: A Short History of ACT-R

Figure 1.11 provides the history of the ideas that are part of the current ACT-R. The origins of ACT-R can be traced back to two books published in 1973. The first was *Human Associative Memory*, which I wrote with Gordon Bower, describing the HAM theory of memory. HAM was one of several then-new efforts to create a rigorous theory of complex human cognition by specifying the theory with sufficient precision that it could be simulated on a computer. Another aspect of this effort that has carried over to modern ACT-R is the idea of a symbolic representation for declarative memory. The proposal in HAM was for a specific propositional representation similar to the proposals of Norman and Rumelhart (1975) and Kintsch (1974). Propositional representations did not generalize well in many applications, so over time the declarative representation has devolved into a more general relational representation.

The second 1973 book was the Carnegie Symposium volume edited in 1973 by Bill Chase that contained two landmark papers by Newell. The first was his famous "You can't play 20 questions with nature and win" paper, in which he lamented the tendency of cognitive psychology to divide the world into little paradigms, each with its own set of questions and logic. In his second paper, Newell (1973b) introduced his answer to this dilemma by describing his first production system theory of human cognition. This single system to perform the diverse set of tasks that occupied cognitive psychology provided the missing ingredient to convert the inert declarative representation of HAM into a functional theory of human cognition.

I combined HAM's declarative system and Newell's procedural system into the first version of the ACT theory (Anderson, 1976), which went beyond either earlier proposal in assuming that there were subsymbolic quantities that controlled access to the declarative and procedural elements. For declarative memory, activation-based quantities were used, inspired by the spreading activation model of Collins and Quillian (1972). For the procedural system, a strength quantity was proposed, based on ideas in psychology that have their origins in behaviorist theories. Both of these concepts evolved as we later considered neural realizations of these quantities and their role in enabling adaptation to the environment.

In 1983 I published a book describing the ACT* system. In it, the subsymbolic computations were changed to be more consistent with the emerging ideas of connectionism. The source I most often referenced was the McClelland and Rumelhart (1981) Interactive Activation Model. There were two other things that ACT* contained that are part of the modern ACT-R theory. One was goal-directed processing – a top-down control to cognition currently served by ACT-R's goal module. The other was a set of ideas for production learning. Among these were ideas for proceduralization and compositionthat are the basic ideas behind the modern production compilation mechanism (see Chapter 4).

I had called the 1983 theory ACT* (pronounced "act star") in loose analogy to the Kleene star to reflect my belief that it was "the final major reformulation within the ACT framework" (Anderson, 1983, p. 18). I said, "my plan for future research is to try to apply this theory wide and far, to eventually gather enough evidence to permanently break the theory and develop a better one" (p. 19). As it turned out, I spent much of the period from 1983 to 1993 engaged in

two activities. One of these was the development of a version of intelligent tutoring systems called cognitive tutors (for a review of those years see Anderson, Corbett, Koedinger, and Pelletier, 1995). This work, while initially motivated to test the ACT* theory and successful in many ways, actually had little direct influence on the theory. The main outcome for ACT-R of that effort was a better technical understanding of how to build production systems. The other effort was already mentioned work on rational analysis of cognition (Anderson, 1990) . While it was started with the intention of abandoning the architectural approach to human cognition, it actually wound up establishing an additional theoretical foundation for the subsymbolic level in ACT-R.

ACT-R came into being in 1993 with the publication of a new book that was an effort to summarize the theoretical progress made on skill acquisition in the intervening 10 years (e.g., Singley and Anderson, 1989) and tune the subsymbolic level of ACT-R with the insights of the rational analysis of cognition. The R in ACT-R was to denote the influence of rational analysis. Accompanying that book was a computer disk containing the first comprehensive implementation of the theory. The fact that we could produce this implementation reflected both our growing understanding (derived from all the production system implementations we had produced) and the fact that LISP, the implementation language of these theories, had become standardized.

The appearance of generally available, fully functioning code set off a series of events that was hardly planned. The catalyst for this was the emergence of a user community. Starting in 1994 on the suggestion of Werner Tack and the insistence of Christian Lebiere, we began holding

summer schools and workshops. The creation of that user community resulted in a whole new dynamic to the theory. One dimension of change was to the language of the theory. The theory became a language spoken among all members of the community, rather than a language spoken by authors of the theory to readers of the theory. This forced a greater standardization and consistency and made it possible for a wide range of researchers to contribute to development of the theory.

1998 saw the publication of the last book in the ACT series until this one. The 1998 book described ACT-R 4.0, which was a much more mature system than the 1993 ACT-R 2.0. Past books in the series had been planned as writing exercises concurrent with the development of the theory and intended to stimulate and discipline that development, whereas ACT-R 4.0 was already basically in place when the book was being written. It was written to display a number of running models built by different researchers, all working in this architecture. There were two notable changes in the architecture by this time. First, reflecting the effort of Lebiere and Anderson (1993) to create a connectionist simulation of ACT-R, we became aware of the need for a pattern matcher that was both more but flexible also more limited in its assumptions about the power of the processes that went into matching a single production. The pattern matcher implemented in ACT-R 4.0 represented a serious claim about what could be recognized in 50 ms of cognition, and this in turn meant that we could take our production rules more seriously. Second, we began producing what I have come to call "end-to-end" simulations that interact with the same (typically computer-based) environment that human participants do, and that actually do the task. This prevented us from making hidden assumptions about linkage to the external world that can protect a theory from disconfirmation.

To enable such end-to-end simulations, we had already began creating perceptual and motor interfaces. As one of these efforts, Mike Byrne had implemented many of the perceptual and motor modules from Meyer and Kieras's (1997) EPIC system into a system called ACT-R/PM – the PM standing for perceptual-motor. It grounded ACT-R in serious models of human perception and action and thus enabled the creation of "embodied" ACT-R models. It became apparent that understanding the perceptual-motor aspects of even abstract tasks like algebra was essential. We decided that these perceptual-motor aspects should be fully integrated into the theory rather than mere add-ons. EPIC also had a modular organization; this strongly influenced our movement to a modular structure.

Another development pushing ACT-R to a modular organization was our entry into fMRI brain imaging research. Slowly, there emerged the mapping between brain regions and modules that is seen in this chapter and throughout the book. This work has had influence on many aspects of ACT-R. For instance, as described in Anderson (2005), it led to the separation of the imaginal and goal modules, which had previously been combined in a single goal module.

Another important event since 1998 was the development of a successful theory of production compilation (described in detail in Chapter 4) with Niels Taatgen. This brought ideas from the 1983 ACT* into the modern ACT-R world. With the theory of production compilation, ACT-R now has a theory of procedural learning to match the successful theory of declarative learning. In addition, we began developing a theory of how such productions could be learned from instruction. This plays a significant role in the model in Figure 1.7 and will be expanded upon

in Chapter 5. An important feature of this is that ACT-R now has a mechanistic explanation of how subjects go from the instruction for a task to performance. (Previously, we just programmed in task-specific productions). One of the last remnants of magic had been eliminated from the theory.

This brings us pretty much up to date. The current simulation version of ACT-R is 6.0, written and maintained by Dan Bothell. In part, its creation was motivated by the desire to better represent the modular structure in the software and to facilitate the development of new modules. That is the history. I will speculate on the future of ACT-R in the Appendix to the last chapter of this book.

Table 1.1

(a) Specifying ACT-R Chunks

```
(add-dm
   (Fact1
   isa addition-fact
   past 10K
   workers 800
   present 7K)
(Fact2
   isa addition-fact
   past 9K
   workers 900
   present 9K)
(Fact3
   isa addition-fact
   past 8K
   workers 1000
   present 11K))
```

(b) Specifying a Connectionist Network

```
set hiddenSize 20
addNet   digits.$hiddenSize
addGroup input      20          INPUT
addGroup hidden     $hiddenSize
addGroup "hidden 2" $hiddenSize OUT_NOISE COSINE_COST
addGroup output     3           OUTPUT
connectGroups input hidden -p RANDOM -s 0.5
connectGroups hidden {"hidden 2"} output
loadExamples digits.ex  -s "clean set"
loadExamples digits2.ex -s "noisy set"
setObj learningRate 0.1
setObj input.numColumns 4
autoPlot
viewUnits
graphObject
```

Figure Captions

Figure 1.1 An illustration of the analogy between physical architecture and cognitive architecture. (Thanks to Andrea Stocco).

Figure 1.2 (a) The results from a Sternberg experiment and the predictions of the model; (a) Sternberg's analysis of the sequence of information-processing stages in his task that generate the predictions in part a. (Sternberg, 1969)

Figure 1.3. The Rumelhart and McClelland (1986) model for past tense generation. The phonological representation of the root is converted into a distributed feature representation. This representation is converted into a distributed feature representation of the past tense, which is then mapped into a phonological representation of the past tense.

Figure 1.4. (a) Probability that a mail message is sent from a source as a function of the number of days since a message was received from that source (Anderson and Schooler, 1991); (b) Saving in relearning as a function of delay (Ebbinghaus, 1885).

Figure 1.5. The interconnections among modules in ACT-R 5.0. From Anderson (2005).

Figure 1.6. Mean solution times (and predictions of the ACT-R model) for the three types of equations as a function of delay. Although the data were not collected, the predicted times are presented for the practice session of the experiment (Day 0).

Figure 1.7. Comparison of the module activity in ACT-R during the solution of a 2-step equation on Day 1 (part a) with a 2-step equation on Day 5 (part b). In both cases the equation being solved is 7*x+3=38

Figure 1.8. Use of module behavior to predict BOLD response in various regions: (a) Manual module predicts motor region; (b) Declarative Module predicts prefrontal region; (c) Control/Goal module predicts anterior cingulate region; (d) Imaginal/Problem State module predicts parietal region; (e) Procedural module (production system) predicts caudate region. The top graph in each figure shows the effect of number of operations averaging over days and the bottom graph shows the effect of days averaging over operations. The actual data are connected by dotted lines and the predictions are the solid lines.

Figure 1.9. Representation of a declarative chunk encoding a fact from the Berry and Broadbent (1984) sugar factory task.

Figure 1.10. Illustration of a production rule in ACT-R. Part (a) illustrates the buffer contents might operate upon in a specific case while part (b) illustrates the general pattern encoded in the rule that would apply to this case.

Figure 1.11 An illustration of the source of the ideas and practices in current ACT-R.

Figure 1.1

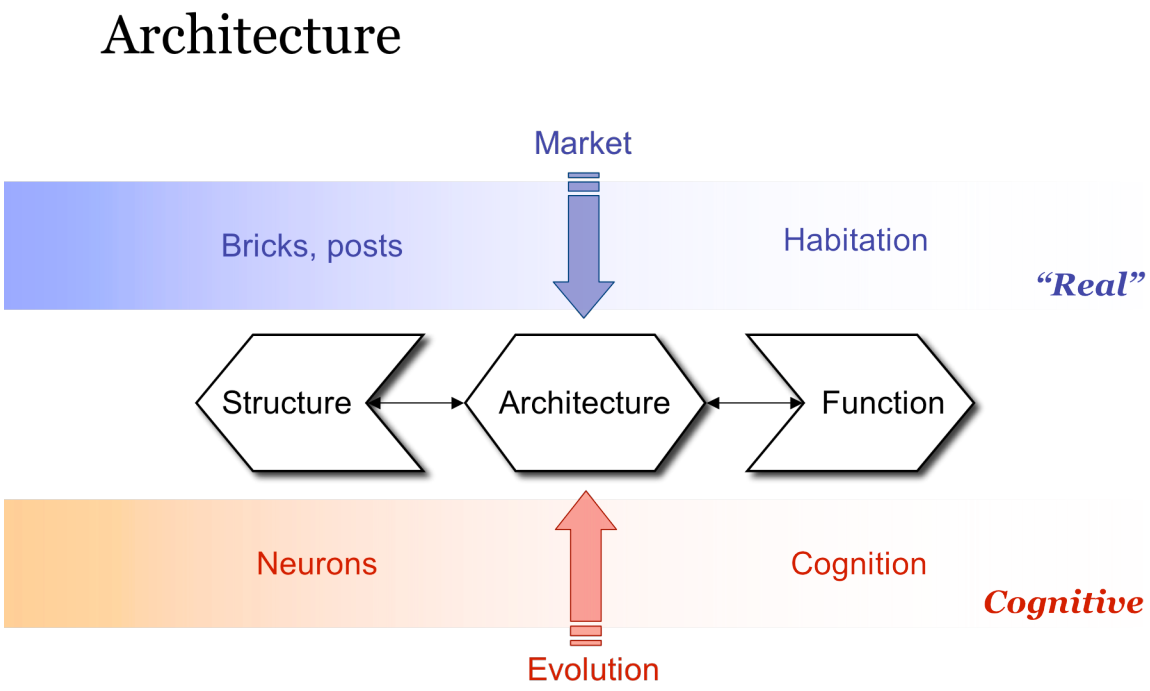
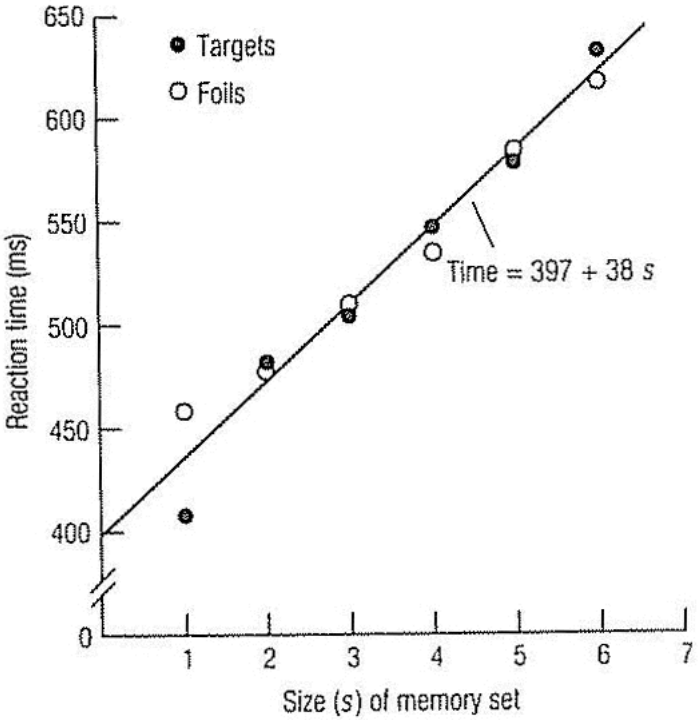

Architecture
Market
Bricks, posts
Habitation
"Real"
Structure
Architecture
Function
Neurons
Cognition
Cognitive
Evolution

Figure 1.2

(a)


650
600
550
500
450
400
0
Reaction time (ms)
Targets
Foils
Time = 397 + 38 s
1
2
3
4
5
6
7
Size (s) of memory set

(b)

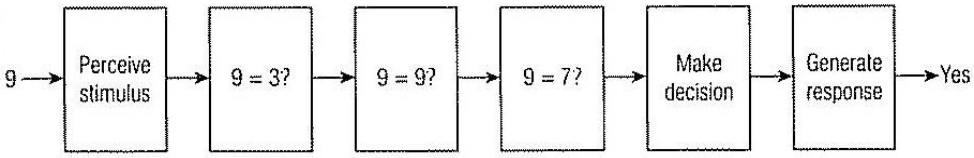

9
Perceive stimulus
9 = 3?
9 = 9?
9 = 7?
Make decision
Generate response
Yes

Figure 1.3

Past tense network

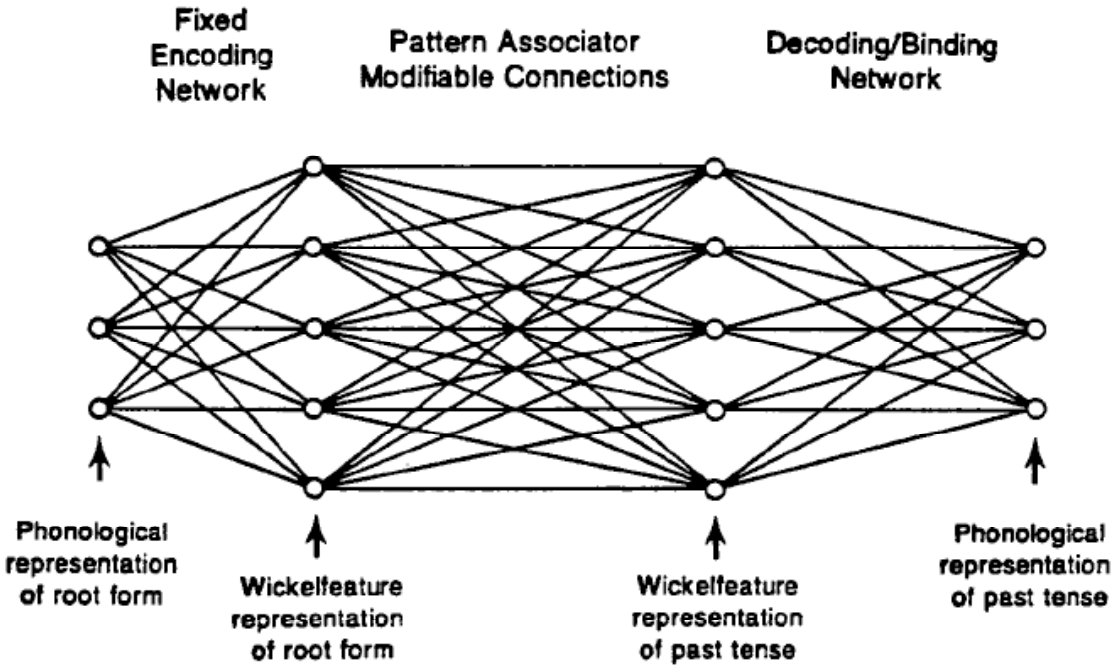
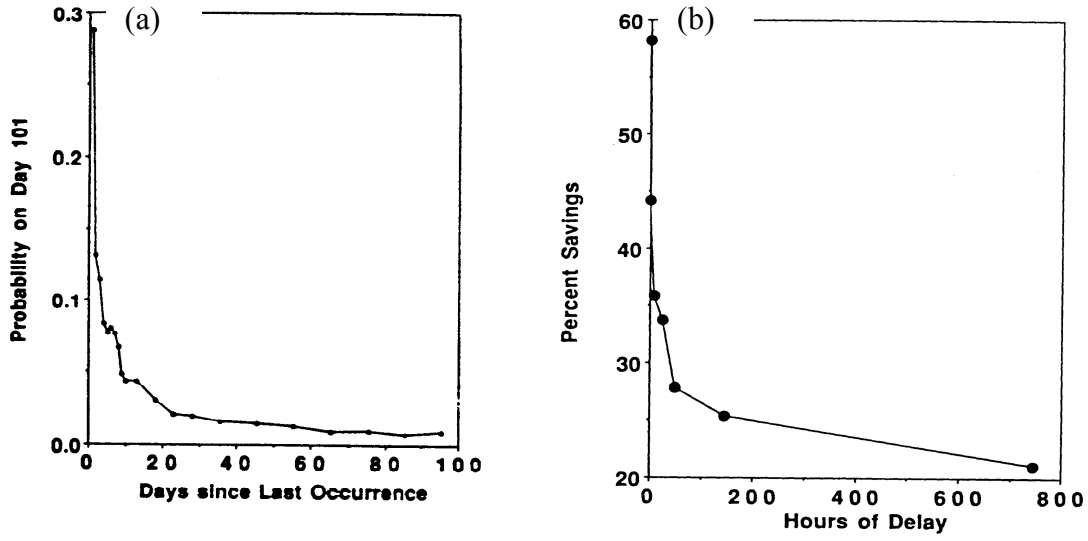


Figure 1.4


(a)
0.3
0.2
0.1
0.0
Probability on Day 101
0
20
40
60
80
100
Days since Last Occurrence
(b)
60
50
40
30
20
Percent Savings
0
200
400
600
800
Hours of Delay

d

Figure 1.5

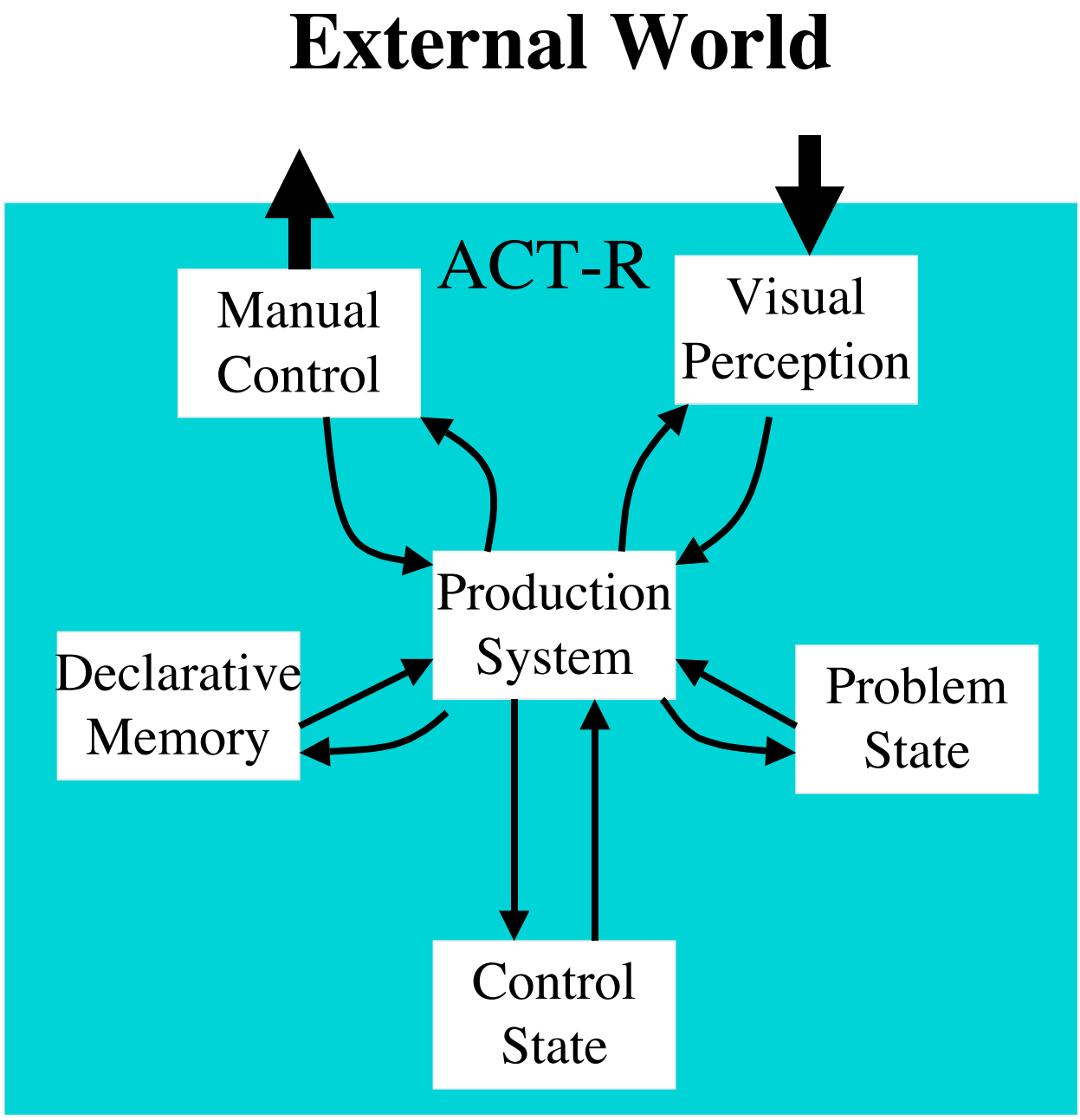

External World
ACT-R
Manual Control
Visual Perception
Production System
Declarative Memory
Problem State
Control State

Figure 1.6

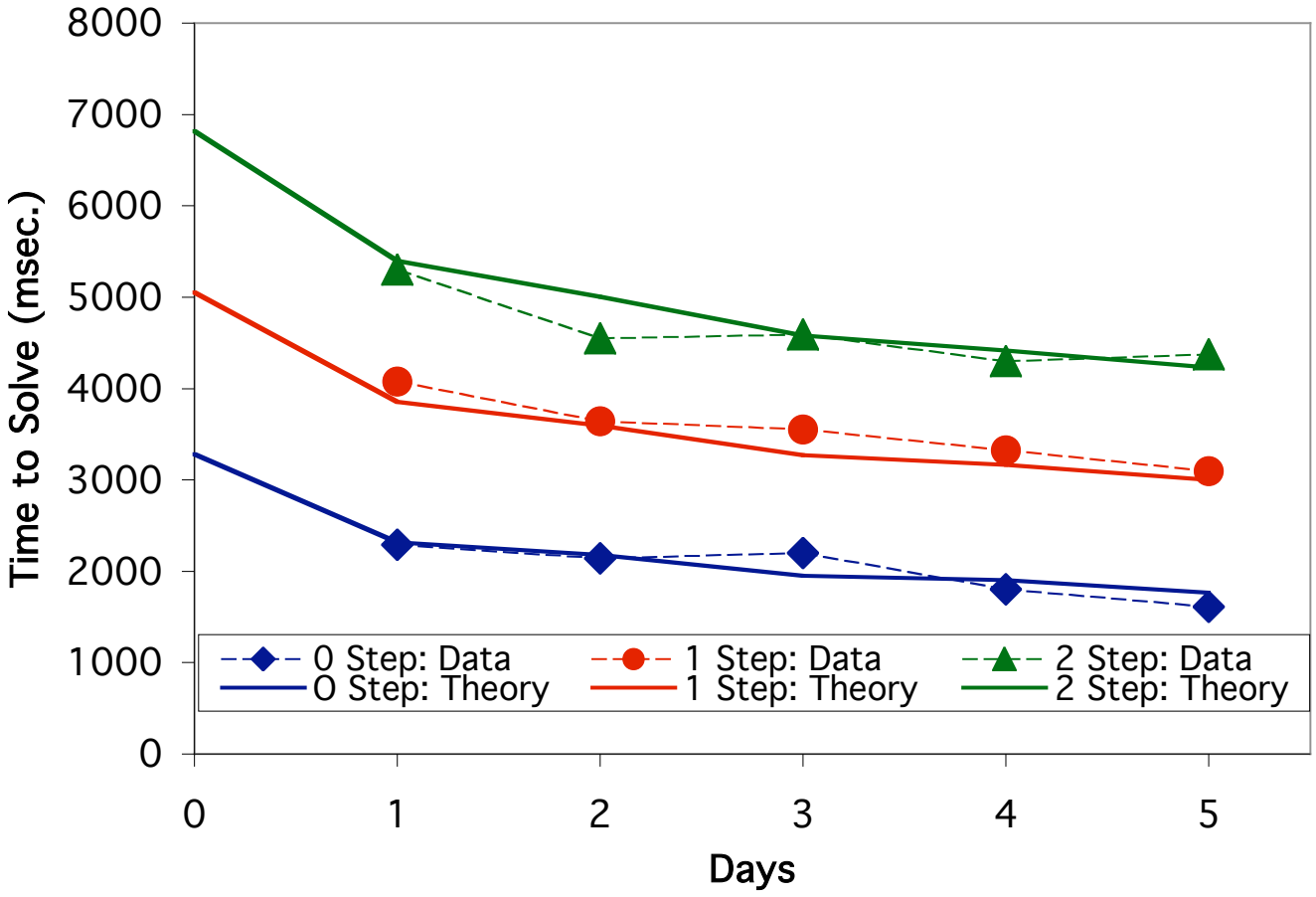

8000
7000
6000
5000
4000
3000
2000
1000
0
Time to Solve (msec.)
0
1
2
3
4
5
Days
0 Step: Data
0 Step: Theory
1 Step: Data
1 Step: Theory
2 Step: Data
2 Step: Theory

Figure 1.7

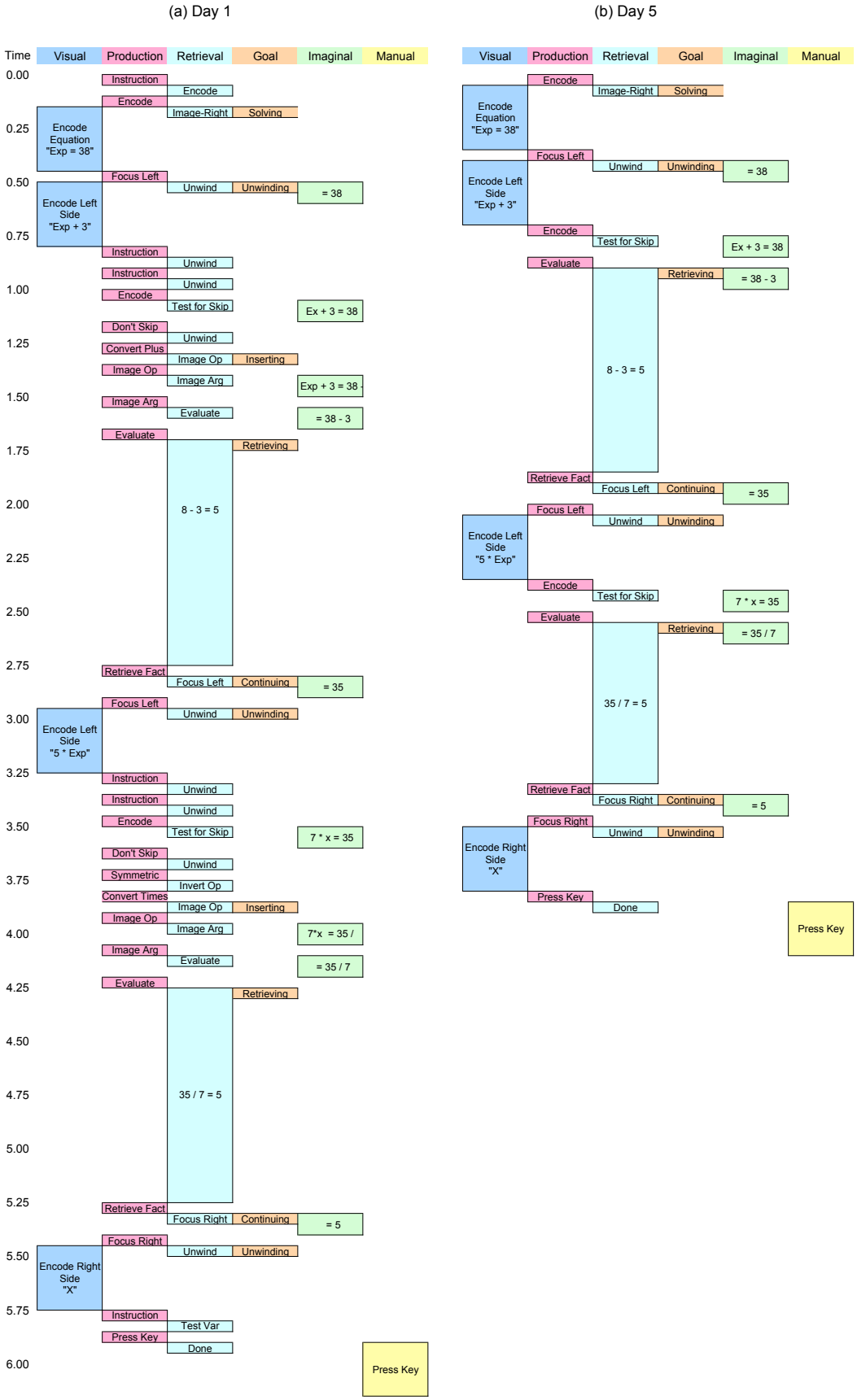

(a) Day 1
Time
Visual
Production
Retrieval
Goal
Imaginal
Manual
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
2.25
2.50
2.75
3.00
3.25
3.50
3.75
4.00
4.25
4.50
4.75
5.00
5.25
5.50
5.75
6.00
Instruction
Encode
Encode
Image-Right
Solving
Encode Equation "Exp = 38"
Focus Left
Unwind
Unwinding
= 38
Encode Left Side "Exp + 3"
Instruction
Unwind
Instruction
Unwind
Encode
Test for Skip
Ex + 3 = 38
Don't Skip
Unwind
Convert Plus
Image Op
Inserting
Image Op
Image Arg
Exp + 3 = 38
Image Arg
Evaluate
= 38 - 3
Evaluate
Retrieving
8 - 3 = 5
Retrieve Fact
Focus Left
Continuing
= 35
Focus Left
Unwind
Unwinding
Encode Left Side "5 * Exp"
Instruction
Unwind
Instruction
Unwind
Encode
Test for Skip
7 * x = 35
Don't Skip
Unwind
Symmetric
Invert Op
Convert Times
Image Op
Inserting
Image Op
Image Arg
7*x = 35 /
Image Arg
Evaluate
= 35 / 7
Evaluate
Retrieving
35 / 7 = 5
Retrieve Fact
Focus Right
Continuing
= 5
Focus Right
Unwind
Unwinding
Encode Right Side "X"
Instruction
Test Var
Press Key
Done
Press Key
(b) Day 5
Visual
Production
Retrieval
Goal
Imaginal
Manual
Encode
Image-Right
Solving
Encode Equation "Exp = 38"
Focus Left
Unwind
Unwinding
= 38
Encode Left Side "Exp + 3"
Encode
Test for Skip
Ex + 3 = 38
Evaluate
Retrieving
= 38 - 3
8 - 3 = 5
Retrieve Fact
Focus Left
Continuing
= 35
Focus Left
Unwind
Unwinding
Encode Left Side "5 * Exp"
Encode
Test for Skip
7 * x = 35
Evaluate
Retrieving
= 35 / 7
35 / 7 = 5
Retrieve Fact
Focus Right
Continuing
= 5
Focus Right
Unwind
Unwinding
Encode Right Side "X"
Press Key
Done
Press Key

Figure 1.8a

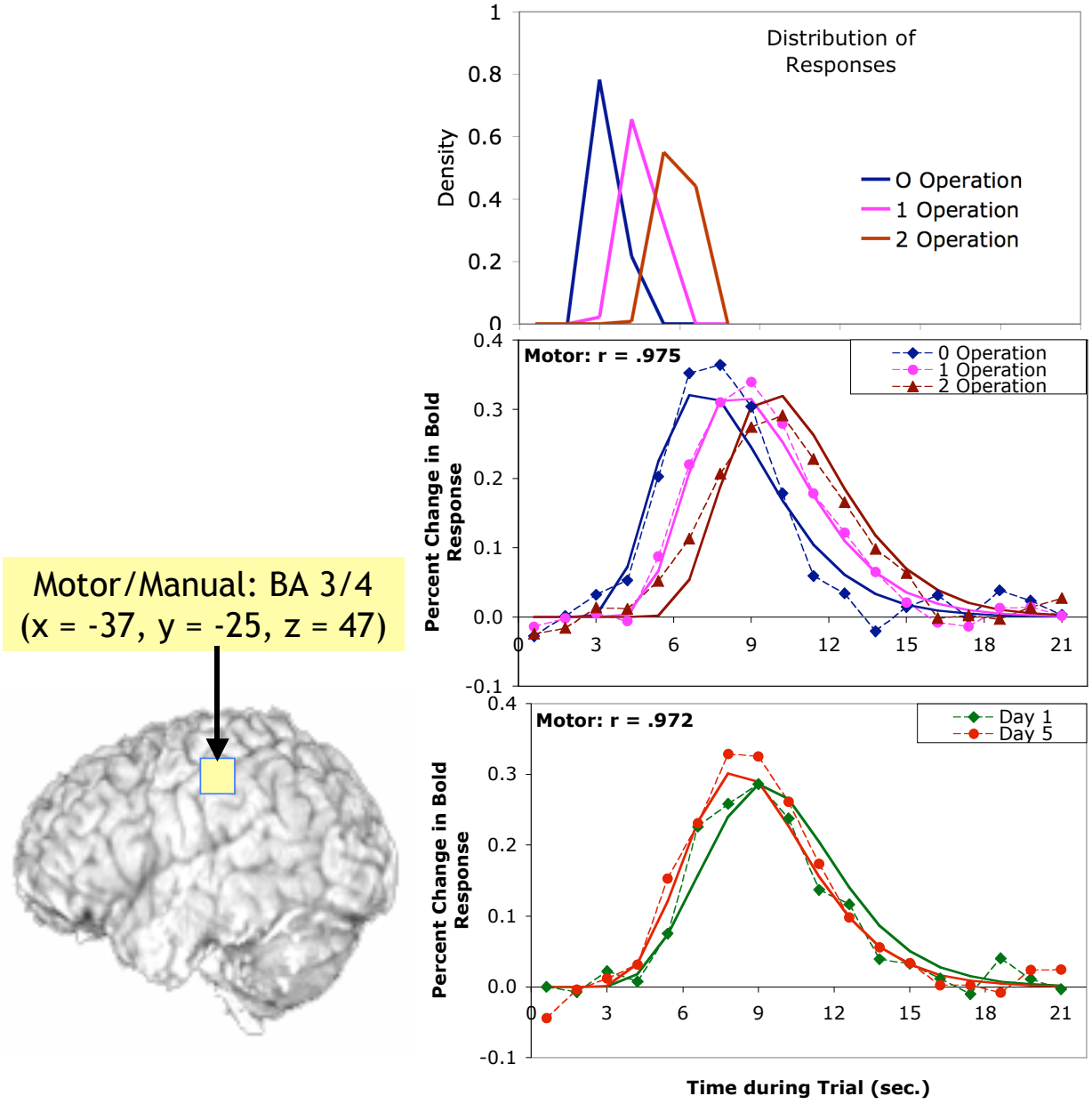

Motor/Manual: BA 3/4
(x = -37, y = -25, z = 47)
Distribution of Responses
Density
0 Operation
1 Operation
2 Operation
Motor: r = .975
Percent Change in Bold Response
Motor: r = .972
Day 1
Day 5
Time during Trial (sec.)

Figure 1.8b

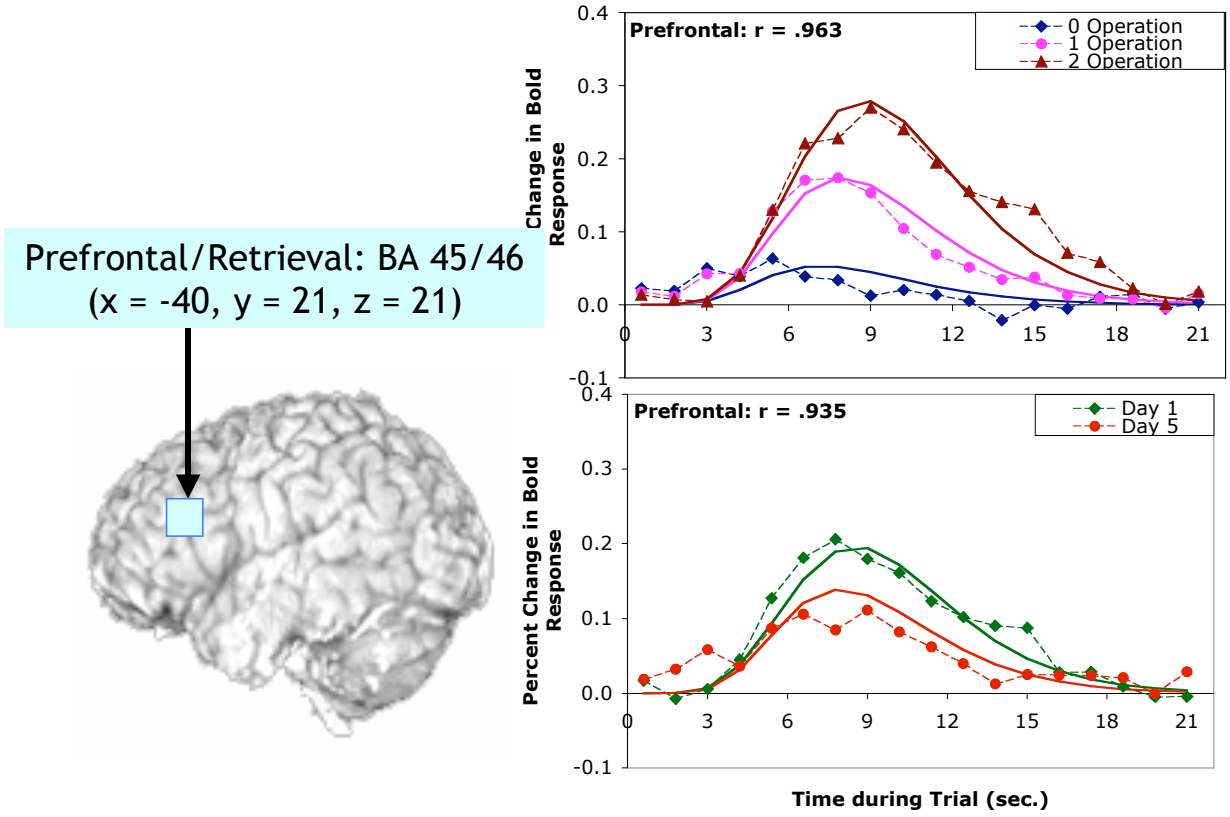

Prefrontal/Retrieval: BA 45/46
(x = -40, y = 21, z = 21)
Prefrontal: r = .963
0 Operation
1 Operation
2 Operation
Change in Bold Response
Prefrontal: r = .935
Day 1
Day 5
Percent Change in Bold Response
Time during Trial (sec.)

Figure 1.8c

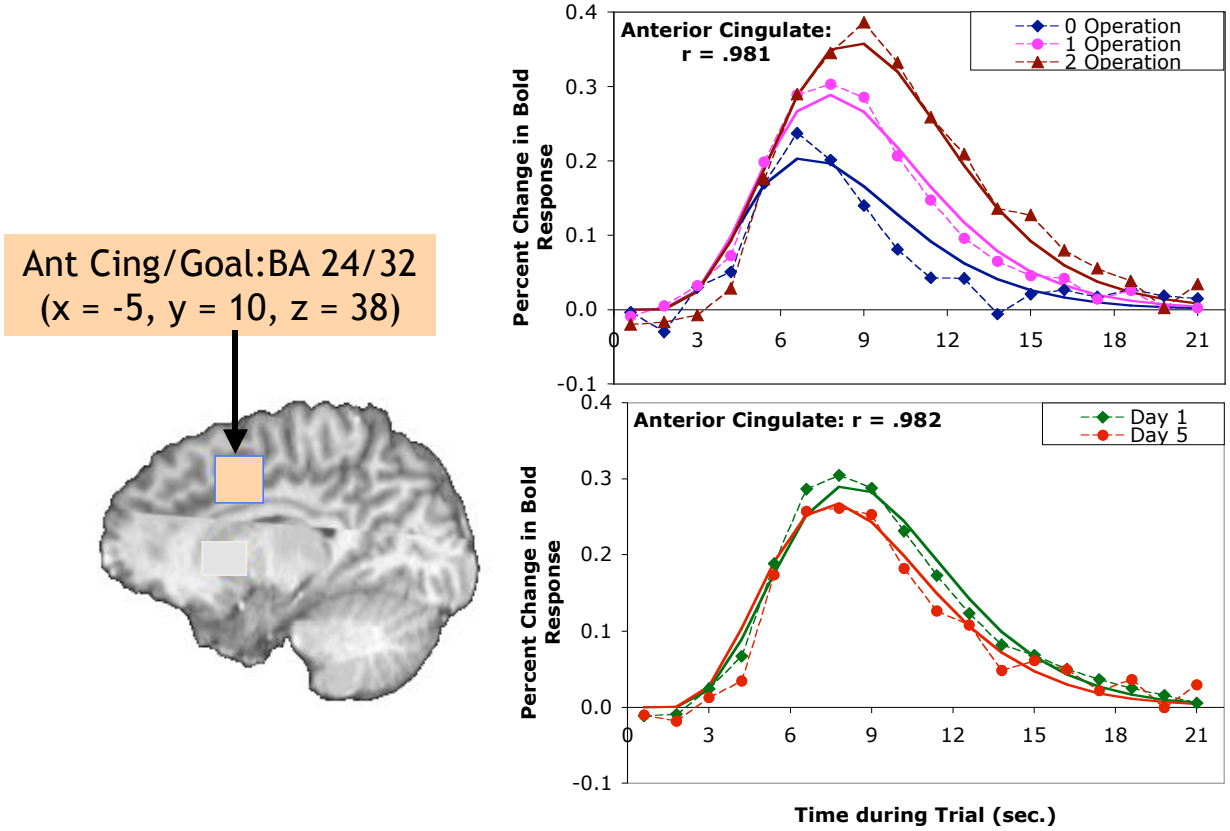

Ant Cing/Goal:BA 24/32
(x = -5, y = 10, z = 38)
Anterior Cingulate:
r = .981
0 Operation
1 Operation
2 Operation
Percent Change in Bold Response
Anterior Cingulate: r = .982
Day 1
Day 5
Percent Change in Bold Response
Time during Trial (sec.)

Figure 1.8d

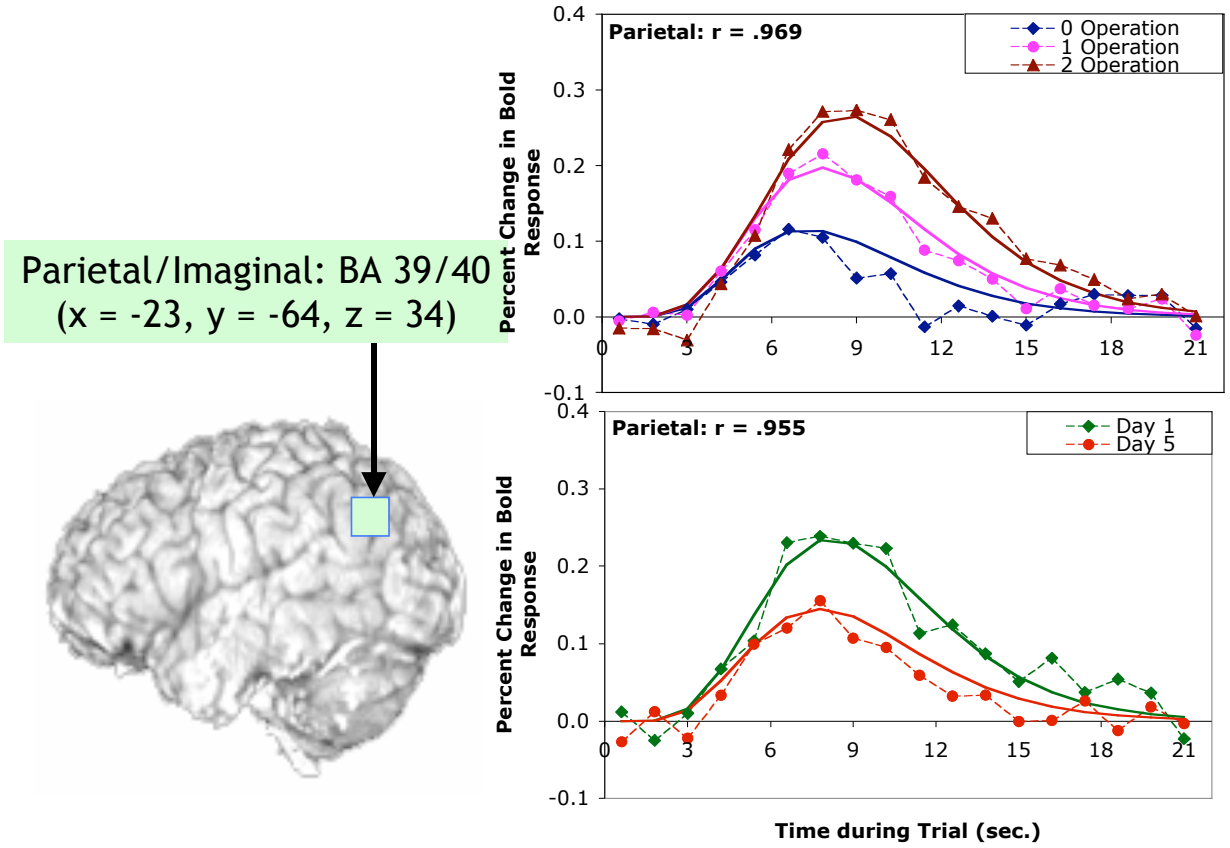

Parietal/Imaginal: BA 39/40
(x = -23, y = -64, z = 34)
Parietal: r = .969
0 Operation
1 Operation
2 Operation
Percent Change in Bold Response
Parietal: r = .955
Day 1
Day 5
Percent Change in Bold Response
Time during Trial (sec.)

Figure 1.8e

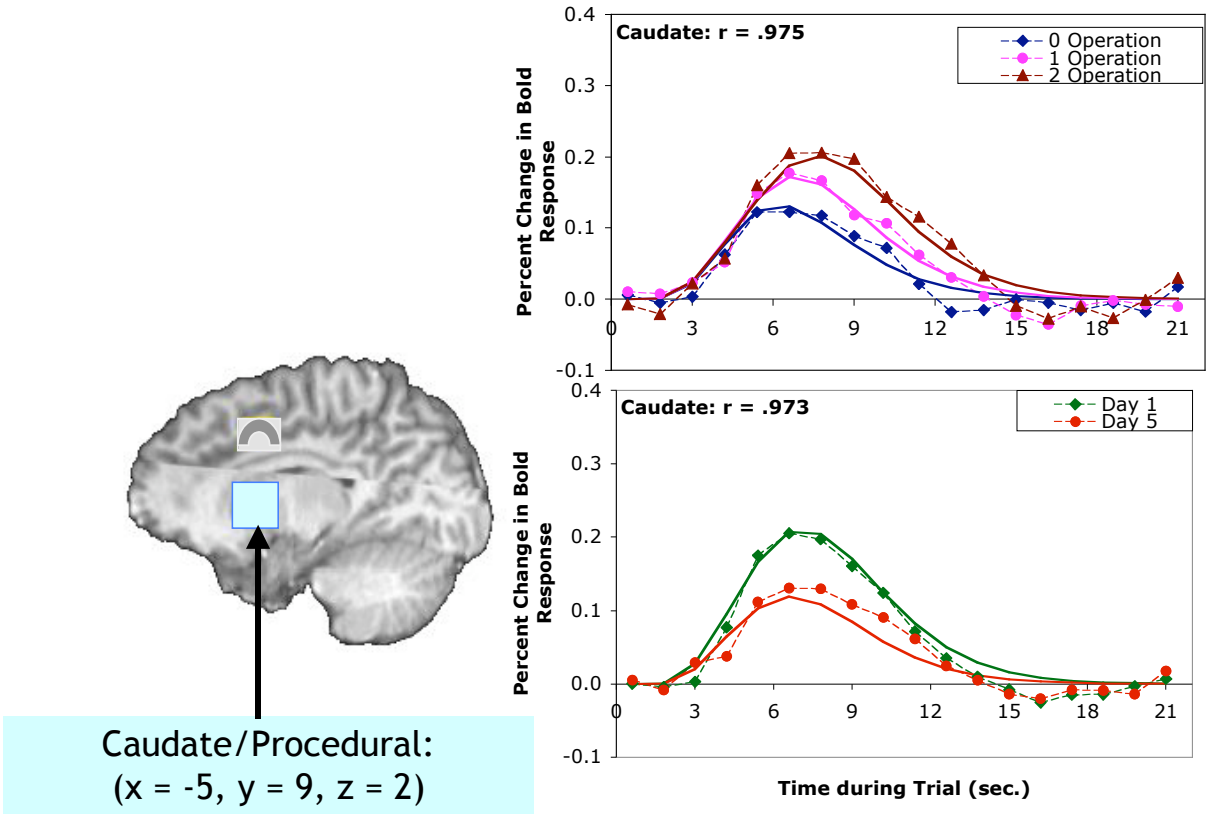

Caudate: r = .975
0 Operation
1 Operation
2 Operation
Percent Change in Bold Response
0.4
0.3
0.2
0.1
0.0
-0.1
0
3
6
9
12
15
18
21
Caudate: r = .973
Day 1
Day 5
Percent Change in Bold Response
Time during Trial (sec.)
Caudate/Procedural:
(x = -5, y = 9, z = 2)

Figure 1.9

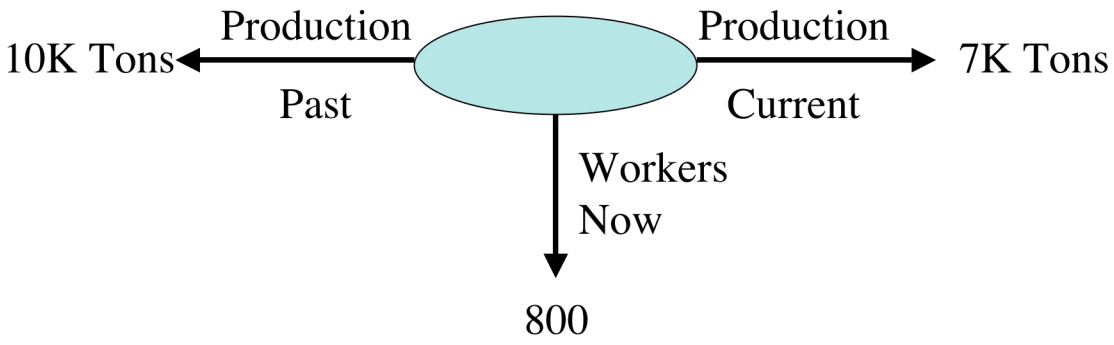

Production
10K Tons
Past
Production
7K Tons
Current
Workers
Now
800

Figure 1.10a

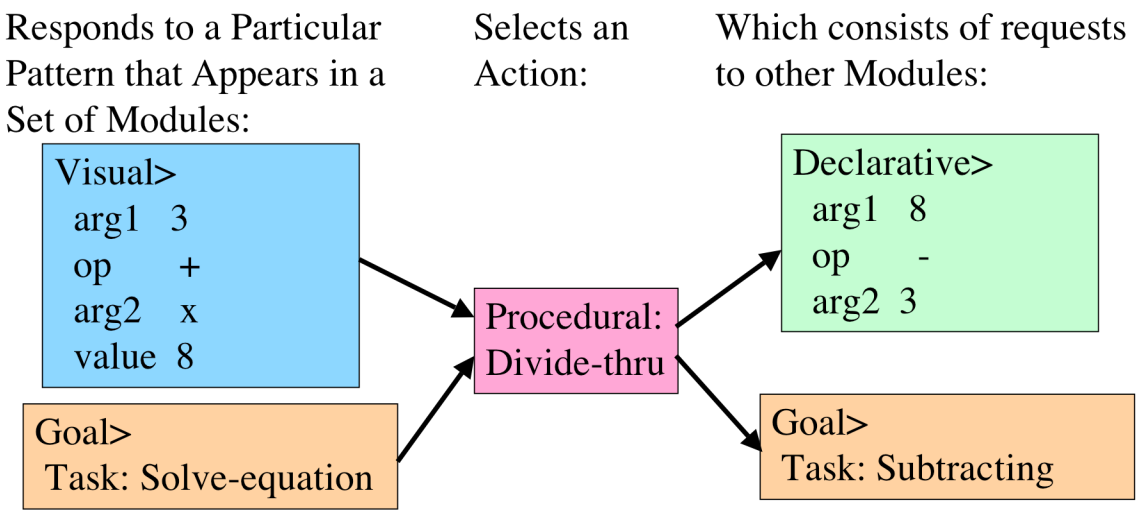


Figure 1.10b

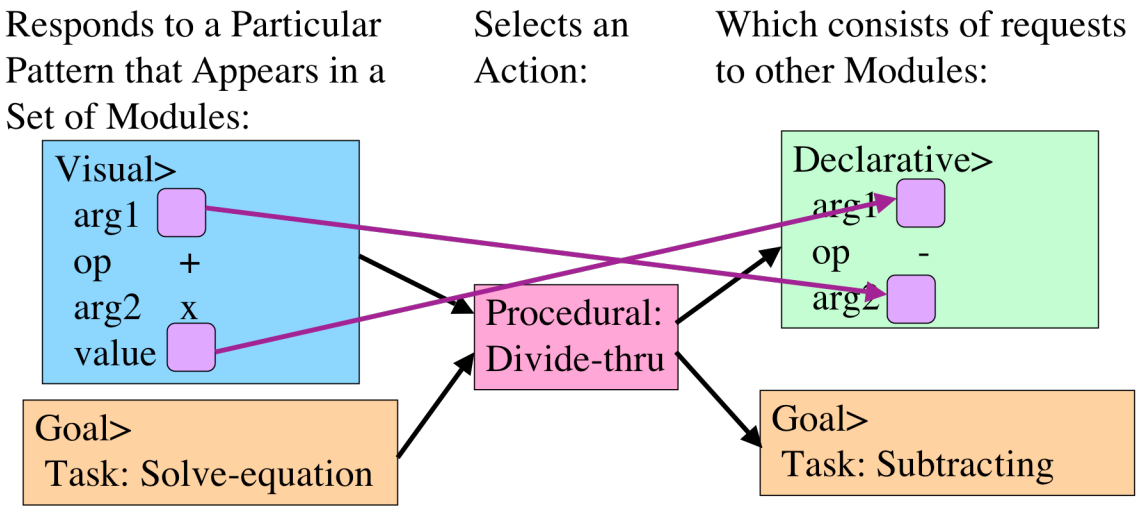
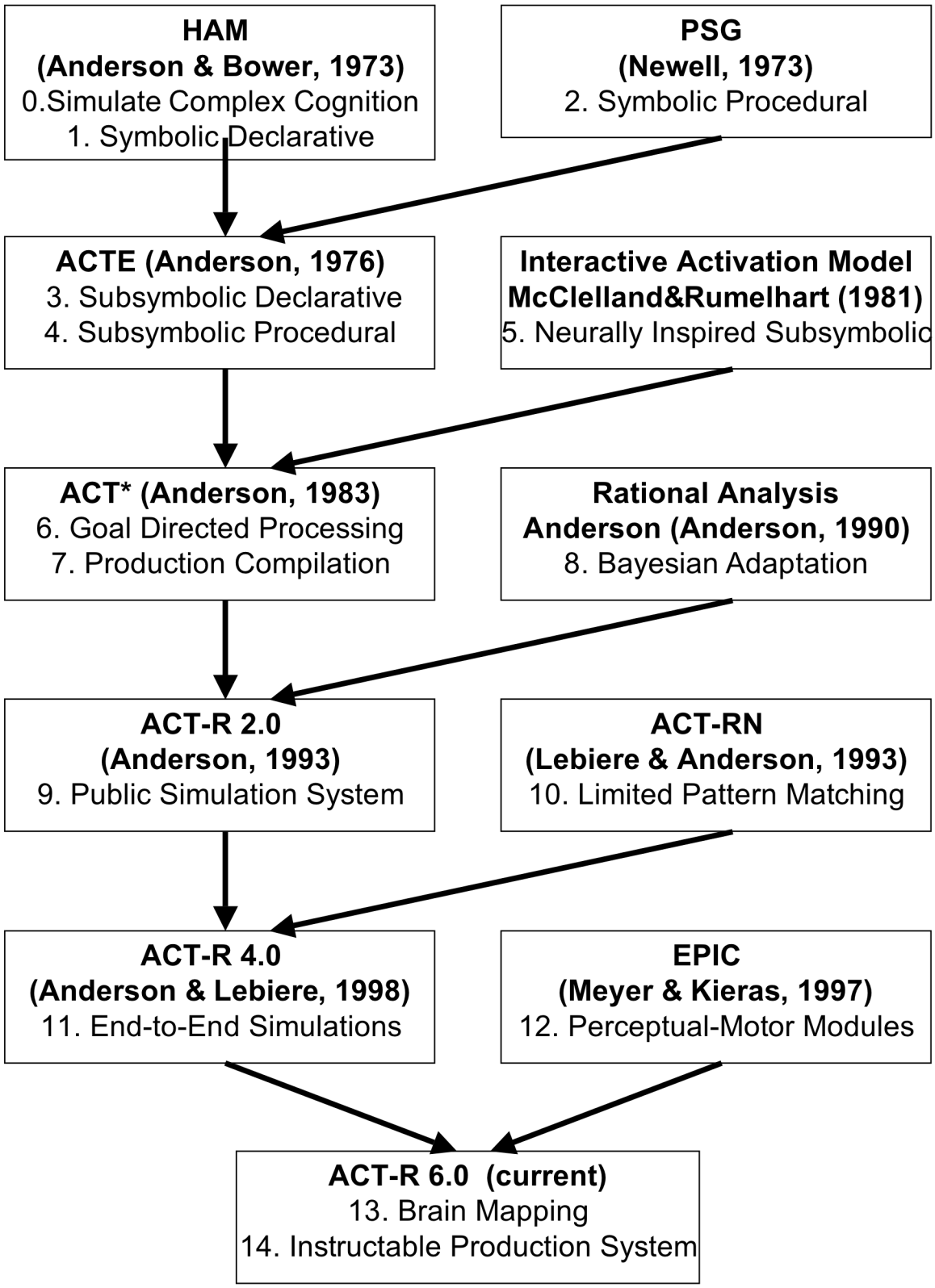


Figure 1.11


HAM
(Anderson & Bower, 1973)
0.Simulate Complex Cognition
1. Symbolic Declarative
PSG
(Newell, 1973)
2. Symbolic Procedural
ACTE (Anderson, 1976)
3. Subsymbolic Declarative
4. Subsymbolic Procedural
Interactive Activation Model
McClelland&Rumelhart (1981)
5. Neurally Inspired Subsymbolic
ACT* (Anderson, 1983)
6. Goal Directed Processing
7. Production Compilation
Rational Analysis
Anderson (Anderson, 1990)
8. Bayesian Adaptation
ACT-R 2.0
(Anderson, 1993)
9. Public Simulation System
ACT-RN
(Lebiere & Anderson, 1993)
10. Limited Pattern Matching
ACT-R 4.0
(Anderson & Lebiere, 1998)
11. End-to-End Simulations
EPIC
(Meyer & Kieras, 1997)
12. Perceptual-Motor Modules
ACT-R 6.0 (current)
13. Brain Mapping
14. Instructable Production System