Psychological Review
2004, Vol. 111, No. 4, 1036–1060

0033-295X/04/$12.00 DOI: 10.1037/0033-295X.111.4.1036

An Integrated Theory of the Mind

John R. Anderson and Daniel Bothell
Carnegie Mellon University

Michael D. Byrne
Rice University

Scott Douglass, Christian Lebiere, and Yulin Qin
Carnegie Mellon University

Adaptive control of thought–rational (ACT–R; J. R. Anderson & C. Lebiere, 1998) has evolved into a theory that consists of multiple modules but also explains how these modules are integrated to produce coherent cognition. The perceptual-motor modules, the goal module, and the declarative memory module are presented as examples of specialized systems in ACT–R. These modules are associated with distinct cortical regions. These modules place chunks in buffers where they can be detected by a production system that responds to patterns of information in the buffers. At any point in time, a single production rule is selected to respond to the current pattern. Subsymbolic processes serve to guide the selection of rules to fire as well as the internal operations of some modules. Much of learning involves tuning of these subsymbolic processes. A number of simple and complex empirical examples are described to illustrate how these modules function singly and in concert.

Psychology, like other sciences, has seen an inexorable movement toward specialization. This is seen in the proliferation of specialty journals in the field but also in the proliferation of special-topic articles in this journal, which is supposed to serve as the place where ideas from psychology meet. Specialization is a necessary response to complexity in a field. Along with this move to a specialization in topics studied, there has been a parallel move toward viewing the mind as consisting of a set of specialized components. With varying degrees of consensus and controversy, there have been claims for separate mechanisms for processing visual objects versus locations (Ungerleider & Miskin, 1982), for procedural versus declarative knowledge (Squire, 1987), for language (Fodor, 1983), for arithmetic (Dehaene, Spelke, Pinel, Stanescu, & Tsivkin, 1999), for categorical knowledge (Warrington & Shallice, 1984), and for cheater detection (Cosmides & Tooby, 2000), to name just a few.

Although there are good reasons for at least some of the proposals for specialized cognitive modules, there is something unsatisfactory about the result—an image of the mind as a disconnected set of mental specialties. One can ask “how is it all put back together?” An analogy here can be made to the study of the body. Modern biology and medicine have seen a successful movement toward specialization, responding to the fact that various body systems and parts are specialized for their functions. However, because the whole body is readily visible, the people who study the shoulder have a basic understanding how their specialty relates to the specialty of those who study the hand, and the people who study the lung have a basic understanding of how their specialty relates to the specialty of those who study the heart. Can one say the same of the person who studies categorization and the person who studies online inference in sentence processing or of the person who studies decision making and the person who studies motor control?

Newell (1990) argued for cognitive architectures that would explain how all the components of the mind worked to produce coherent cognition. In his book, he described the Soar system, which was his best hypothesis about the architecture. We have been working on a cognitive architecture called adaptive control of thought–rational (ACT–R; e.g., Anderson & Lebiere, 1998), which is our best hypothesis about such an architecture. It has recently undergone a major development into a version called ACT–R 5.0, and this form offers some new insights into the integration of cognition. The goal of this article is to describe how cognition is integrated in the ACT–R theory. The quote below gives the essence of Newell’s argument for an integrated system:

> A single system (mind) produces all aspects of behavior. It is one mind that minds them all. Even if the mind has parts, modules, components, or whatever, they all mesh together to produce behavior. Any bit of behavior has causal tendrils that extend back through large parts of the total cognitive system before grounding in the environmental situation of some earlier times. If a theory covers only one part or component, it flirts with trouble from the start. It goes without saying that there are dissociations, independencies, impenetrabilities, and modularities. These all help to break the web of each bit of behavior being shaped by an unlimited set of antecedents. So they are important to understand and help to make that theory simple enough to use. But they don’t remove the necessity of a theory that provides

John R. Anderson, Daniel Bothell, Scott Douglass, and Yulin Qin, Psychology Department, Carnegie Mellon University; Michael D. Byrne, Psychology Department, Rice University; Christian Lebiere, Human Computer Interaction Institute, Carnegie Mellon University.

This research was supported by National Aeronautics and Space Administration Grant NCC2-1226 and Office of Naval Research Grant N00014-96-01491.

Correspondence concerning this article should be addressed to John R. Anderson, Department of Psychology, Carnegie Mellon University, 352 Baker Hall, Pittsburgh, PA 15213. E-mail: ja@cmu.edu

> the total picture and explains the role of the parts and why they exist. (pp. 17–18)

Newell (1990) enumerated many of the advantages that a unified theory has to offer; this article develops two advantages related to the ones he gives. The first is concerned with producing a theory that is capable of attacking real-world problems, and the second is concerned with producing a theory that is capable of integrating the mass of data from cognitive neuroscience methods like brain imaging.

The remaining sections of this article consist of two major parts and then a conclusion. The first major part is concerned with describing the ACT–R theory and consists of five sections, one describing the overall theory and then four sections elaborating on the major components of the system: the perceptual-motor modules, the goal module, the declarative module, and the procedural system. As we describe each component, we try to identify how it contributes to the overall integration of cognition. The second major part of the article consists of two sections illustrating the applications of an integrated architecture to understanding our two domains of interest. One section describes an application of the ACT–R theory to understanding acquisition of human skill with a complex real-world system, and the other section describes an application to integrating data that come from a complex brain imaging experiment.

The ACT–R 5.0 Architecture

Figure 1 illustrates the basic architecture of ACT–R 5.0. It consists of a set of modules, each devoted to processing a different kind of information. Figure 1 contains some of the modules in the system: a visual module for identifying objects in the visual field, a manual module for controlling the hands, a declarative module for retrieving information from memory, and a goal module for keeping track of current goals and intentions. Coordination in the behavior of these modules is achieved through a central production system. This central production system is not sensitive to most of the activity of these modules but rather can only respond to a limited amount of information that is deposited in the buffers of these modules. For instance, people are not aware of all the information in the visual field but only the object they are currently attending to. Similarly, people are not aware of all the information in long-term memory but only the fact currently retrieved. Thus, Figure 1 illustrates the buffers of each module passing information back and forth to the central production system. The core production system can recognize patterns in these buffers and make changes to these buffers, as, for instance, when it makes a request to perform an action in the manual buffer. In the terms of Fodor (1983), the information in these modules is largely encapsulated, and the modules communicate only through the information they make available in their buffers. It should be noted that the EPIC (executive-process/interactive control) architecture (Kieras, Meyer, Mueller, & Seymour, 1999) has adopted a similar modular organization for its production system architecture.

The theory is not committed to exactly how many modules there are, but a number have been implemented as part of the core system. The buffers of these modules hold the limited information that the production system can respond to. They have similarities to Baddeley's (1986) working memory "slave" systems. The buffers in Figure 1 are particularly important to this article, and we have noted cortical regions we think they are associated with. The goal buffer keeps track of one's internal state in solving a problem. In Figure 1, it is associated with the dorsolateral prefrontal cortex

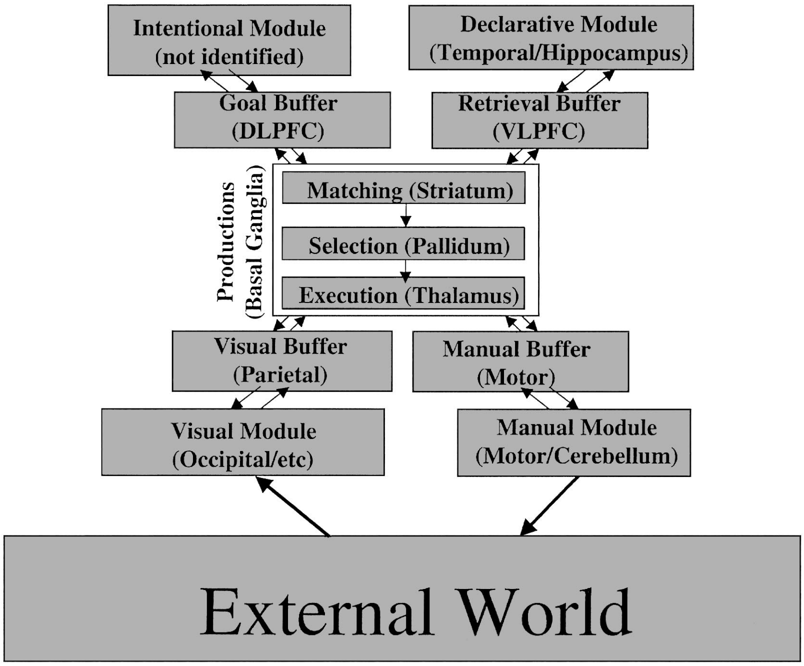


Figure 1. The organization of information in ACT–R 5.0. Information in the buffers associated with modules is responded to and changed by production rules. DLPFC = dorsolateral prefrontal cortex; VLPFC = ventrolateral prefrontal cortex.

(DLPFC), but as we discuss later, its neural associations are undoubtedly more complex. The retrieval buffer, in keeping with the HERA (hemispheric encoding–retrieval asymmetry) theory (Nyberg, Cabeza, & Tulving, 1996) and other recent neuroscience theories of memory (e.g., Buckner, Kelley, & Petersen, 1999; Wagner, Paré-Blagoev, Clark, & Poldrack, 2001), is associated with the ventrolateral prefrontal cortex (VLPFC) and holds information retrieved from long-term declarative memory.[1] This distinction between DLPFC and VLPFC is in keeping with a number of neuroscience results (Braver et al., 2001; Cabeza, Dolcos, Graham, & Nyberg, 2002; Fletcher & Henson, 2001; Petrides, 1994; Thompson-Schill, D'Esposito, Aguirre, & Farah, 1997). The perceptual-motor modules' buffers are based on Byrne and Anderson's (2001) ACT–R/perceptual-motor (ACT–R/PM), which in turn is based on Meyer and Kieras's (1997) EPIC. The manual buffer is responsible for control of the hands and is associated with the adjacent motor and somatosensory cortical areas devoted to controlling and monitoring hand movement. One of the visual buffers, associated with the dorsal "where" path of the visual system, keeps track of locations, while the other, associated with the ventral "what" system, keeps track of visual objects and their identity. The visual and manual systems are particularly important in many tasks to which ACT–R has applied, in which participants scan a computer screen, type, and use a mouse at a keyboard. There also are rudimentary vocal and aural systems. The contents of these buffers can be determined by rather elaborate systems within the modules. For instance, the contents of the visual buffers represent the products of complex processes of the visual perception and attention systems. Similarly, the contents of the retrieval buffer are determined by complex memory processes, as we describe below.

ACT–R 5.0 includes a theory of how these buffers interact to determine cognition. The basal ganglia and associated connections are thought to implement production rules in ACT–R. The cortical areas corresponding to these buffers project to the striatum, part of the basal ganglia, which we hypothesize performs a pattern-recognition function (in line with other proposals; e.g., Amos, 2000; Frank, Loughry, & O'Reilly 2000; Houk & Wise, 1995; Wise, Murray, & Gerfen, 1996). This portion of the basal ganglia projects to a number of small regions known collectively as the pallidum. The projections to the pallidum are substantially inhibitory, and these regions in turn inhibit cells in the thalamus, which projects to select actions in the cortex. Graybiel and Kimura (1995) have suggested that this arrangement creates a "winner-lose-all" system such that active striatal projections strongly inhibit only the pallidum neurons representing the selected action (which then no longer inhibit the thalamus from producing the action). This is a mechanism by which the winning production comes to dominate. According to Middleton and Strick (2000), at least five regions of the frontal cortex receive projections from the thalamus and are controlled by this basal ganglia loop. These regions play a major role in controlling behavior.

Thus, the basal ganglia implement production rules in ACT–R by the striatum serving a pattern-recognition function, the pallidum serving a conflict-resolution function, and the thalamus controlling the execution of production actions. Because production rules represent ACT–R's procedural memory, this also corresponds to proposals that basal ganglia subserve procedural learning (Ashby & Waldron, 2000; Hikosaka et al., 1999; Saint-Cyr, Taylor, & Lang, 1988). An important function of the production rules is to update the buffers in the ACT–R architecture. The organization of the brain into segregated, cortico–striatal–thalamic loops is consistent with this hypothesized functional specialization. Thus, the critical cycle in ACT–R is one in which the buffers hold representations determined by the external world and internal modules, patterns in these buffers are recognized, a production fires, and the buffers are then updated for another cycle. The assumption in ACT–R is that this cycle takes about 50 ms to complete—this estimate of 50 ms as the minimum cycle time for cognition has emerged in a number of cognitive architectures including Soar (Newell, 1990), 3CAPS (capacity-constrained collaborative activation-based production system; Just & Carpenter, 1992), and EPIC (Meyer & Kieras, 1997). Thus, a production rule in ACT–R corresponds to a specification of a cycle from the cortex, to the basal ganglia, and back again. The conditions of the production rule specify a pattern of activity in the buffers that the rule will match, and the action specifies changes to be made to buffers.

The architecture assumes a mixture of parallel and serial processing. Within each module, there is a great deal of parallelism. For instance, the visual system is simultaneously processing the whole visual field, and the declarative system is executing a parallel search through many memories in response to a retrieval request. Also, the processes within different modules can go on in parallel and asynchronously. However, there are also two levels of serial bottlenecks in the system. First, the content of any buffer is limited to a single declarative unit of knowledge, called a *chunk* in ACT–R. Thus, only a single memory can be retrieved at a time or only a single object can be encoded from the visual field. Second, only a single production is selected at each cycle to fire. In this second respect, ACT–R 5.0 is like Pashler's (1998) central bottleneck theory and quite different, at least superficially, from the other prominent production system conceptions (CAPS, EPIC, and Soar).

Subsequent sections of the article describe the critical components of this theory—the perceptual-motor system, the goal system, the declarative memory, and the procedural system. Although each is its own separate system, each contributes to the overall integration of cognition. After describing these components, we discuss two examples of how they work together to achieve integrated cognitive function.

The Perceptual-Motor System

As a matter of division of labor, not as a claim about significance, ACT–R historically was focused on higher level cognition and not perception or action. Perception and action involve systems every bit as complex as higher level cognition. Dealing with higher level cognition had seemed quite enough. However, this division of labor tends to lead to a treatment of cognition that is totally abstracted from the perceptual-motor systems, and there is reason to suppose that the nature of cognition is strongly determined by its perceptual and motor processes, as the proponents of embodied and situated cognition have argued. In particular, the external world can provide much of the connective tissue that integrates cognition. For instance, consider the difficulty one ex-

[1] There is a great deal of evidence that long-term memory, which is part of the retrieval module as distinct from the buffer, is associated with the temporal lobes and hippocampus.

periences trying to do a proof in geometry without a diagram to inspect and mark.

With their EPIC architecture, Meyer and Kieras (1997) developed a successful strategy for relating cognition to perception and action without dealing directly with real sensors or real effectors and without having to embed all the detail of perception and motor control. This is a computational elaboration of the successful model human processor system defined by Card, Moran, and Newell (1983) for human–computer interaction applications. This approach involves modeling, in approximate form, the basic timing behavior of the perceptual and motor systems, the output of the perceptual systems, and the input to the motor system. We have adopted exactly the same strategy and to a substantial degree just reimplemented certain aspects of the EPIC system. Undoubtedly, this strategy of approximation will break down at points, but it has proven quite workable and has had a substantial influence on the overall ACT–R system. We hope that the architecture that has emerged will be compatible with more complete models of the perceptual and motor systems.

The primary difference between ACT–R's perceptual-motor machinery and EPIC's is in the theory of the visual system. The ACT–R visual system separates vision into two modules, each with an associated buffer. A visual-location module and buffer represent the dorsal where system and a visual-object module and buffer represent the ventral what system. ACT–R implements more a theory of visual attention than a theory of perception in that it is concerned with what the system chooses to encode in its buffers but not the details of how different patterns of light falling on the retina yield particular representations.

When a production makes a request of the where system, the production specifies a series of constraints, and the where system returns a chunk representing a location meeting those constraints. Constraints are attribute–value pairs that can restrict the search based on visual properties of the object (such as "color: red") or the spatial location of the object (such as "vertical: top"). This is akin to preattentive visual processing (Treisman & Gelade, 1980) and supports visual pop-out effects. For example, if the display consists of one green object in a field of blue objects, the time to determine the location of the green object is constant regardless of the number of blue objects. If there are multiple objects satisfying a request to the where system, the location of one will be determined at random. To find the target object may require a self-terminating search through the objects satisfying the description.

Through the where system, ACT–R has knowledge of where all the objects are in its environment and what some of their basic features are. However, to identify an object, it must make a request of the what system. A request to the what system entails providing a chunk representing a visual location, which will cause the what system to shift visual attention to that location, process the object located there, and generate a declarative memory chunk representing the object. The system supports two levels of granularity here, a coarse one in which all attention shifts take a fixed time regardless of distance and a more detailed one with an eye-movement model. For the fixed-time approximation, this parameter is set at 185 ms in ACT–R and serves as the basis for predicting search costs in situations in which complete object identification is required.[2] However, ACT–R does not predict that all visual searches should require 185 ms/item. Rather, it is possible to implement in ACT–R versions of feature-guided search that can progress more rapidly. There is considerable similarity between the current implementation of visual attention in ACT–R and Wolfe's (1994) GS (guided search) theory, and indeed we plan to adapt Wolfe's GS into ACT–R.

Salvucci's (2001) EMMA (eye movements and movement of attention) system has been built with ACT–R to provide a more detailed theory of visual encoding. It is based on a number of models of eye-movement control in reading, particularly the E-Z Reader model (Reichle, Pollatsek, Fisher, & Rayner, 1998; Reichle, Rayner, & Pollatsek, 1999). In EMMA, the time between the request for a shift of attention and the generation of the chunk representing the visual object of that location is dependent on the eccentricity between the requested location and the current point of gaze, with nearer objects taking less time than farther objects. The theory assumes that eye movements follow shifts of attention and that the ocular-motor system programs a movement to the object.

The ACT–R model described by Byrne and Anderson (2001) for the Schumacher et al. (1997; also reported in Schumacher et al., 2001) experiment is a useful illustration of how the perceptual-motor modules work together. It involves interleaving multiple perceptual-motor threads and has little cognition to complicate the exposition. The experiment itself is interesting because it is an instance of perfect time sharing. It involved two simple choice reaction time tasks: three-choice (low–middle–high) tone discrimination with a vocal response and three-choice (left–middle–right) visual position discrimination with a manual response. Both of these tasks are simple and can be completed rapidly by experimental participants. Schumacher et al. (1997) had participants train on these two tasks separately, and they reached average response times of 445 ms for the tone discrimination task and 279 ms for the location discrimination task. Participants were then asked to do the two tasks together with simultaneous stimulus presentation, and they were encouraged to overlap processing of the two stimuli. In the dual-task condition, they experienced virtually no dual-task interference—283 ms average response time for the visual-manual task and 456 ms average response time for the auditory-vocal task.

Byrne and Anderson (2001) constructed an ACT–R/PM model of the two tasks and the dual task. A schedule chart for the dual-task model is presented in Figure 2. Consider the visual-motor task first. There is a quick 50-ms detection of the visual position (does not require object identification) and a 50-ms production execution to request the action, followed by the preparation and execution of the motor action. With respect to the auditory-vocal task, there is first the detection of the tone (but this takes longer than detection of visual position), then a production executes requesting the speech, and then there is a longer but analogous process of executing the speech. According to the ACT–R model, there is nearly perfect time sharing between the two tasks because the demands on the central production system are offset in time. Figure 3 presents the predictions of the ACT–R model for the task. There is an ever-so-small dual-task deficit because of variability in the completion times for all the perceptual-motor stages, which occasionally results in a situation

[2] The actual value of this parameter in various instantiations of ACT–R has been the source of some confusion. In the first visual interface for ACT–R, all activity was serialized, and so, this value was 185 ms. However, in ACT–R 5.0, the actual system parameter is 85 ms because the same attention shift now requires two production firings.

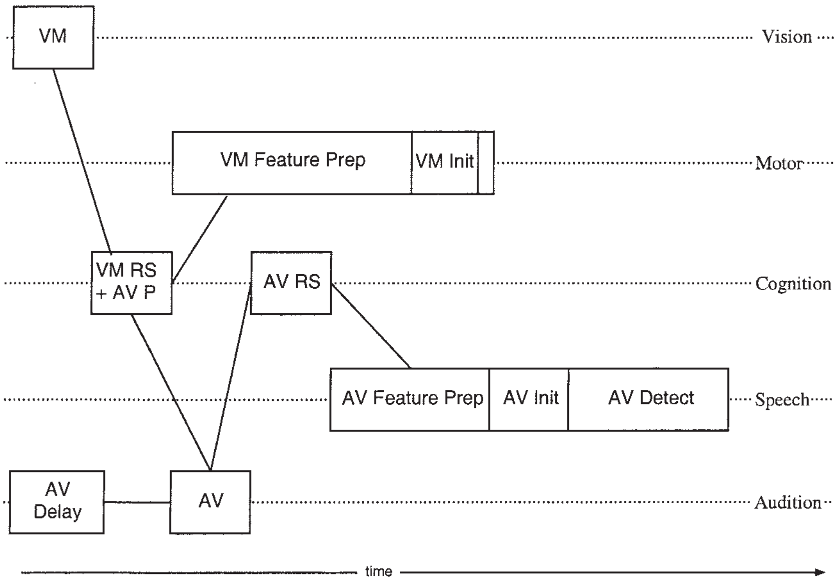


Figure 2. The ACT–R schedule chart for Schumacher et al. (1997). VM = visual–manual task; Prep = preparation; Init = motor initiation; RS = response selection; AV P= auditory–verbal perception; AV = auditory–verbal task. From "Serial Modules in Parallel: The Psychological Refractory Period and Perfect Time-Sharing," by M. D. Byrne and J. R. Anderson, 2001, *Psychological Review, 108,* p. 856. Copyright 2001 by the American Psychological Association.

in which the production for the auditory-vocal task must wait for the completion of the visual-motor production.

This model nicely illustrates the parallel threads of serial processing in each module, which is a hallmark of EPIC and ACT–R. Figure 2 also illustrates that the central production-system processor is also serial, a feature that distinguishes ACT–R from EPIC. However, in this experiment, there was almost never contention between the two tasks for access to the central processor (or for access to any other module).

There has been considerable further analysis of perfect time sharing since the original Schumacher et al. (1997) experiment, including a more elaborate series of studies by Schumacher et al. (2001) and a careful analysis by Hazeltine, Teague, and Ivry (2002) that argue against a central bottleneck and an article by Ruthruff, Pashler, and Hazeltine (2003) that argues for it. ACT–R predicts that the amount of interference will be minimal between two tasks that are well practiced and that do not make use of the same perceptual and motor systems At high levels of practice, each

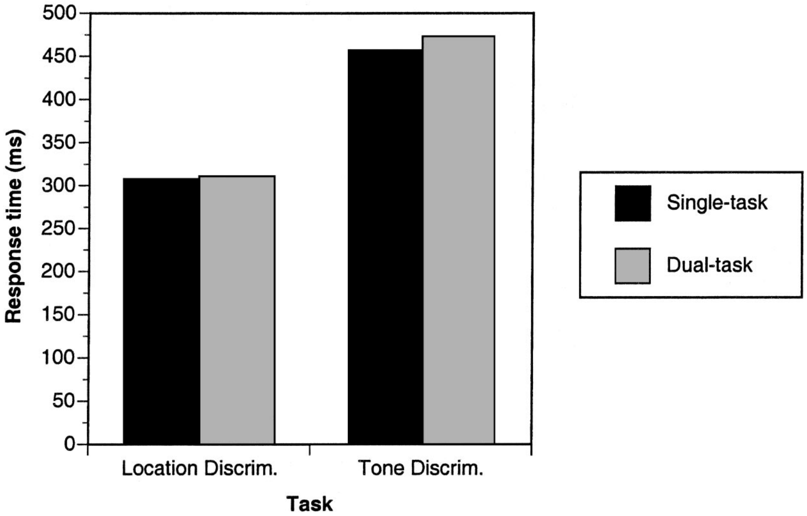


Figure 3. Predictions of the ACT–R model for Schumacher et al. (1997). Discrim. = discrimination. From "Serial Modules in Parallel: The Psychological Refractory Period and Perfect Time-Sharing," by M. D. Byrne and J. R. Anderson, 2001, *Psychological Review, 108,* p. 857. Copyright 2001 by the American Psychological Association.

will be reduced to a single production rule, and the maximal interference that will be displayed between them is 50 ms if the two tasks make simultaneous requests on the production system. Hazeltine et al. did find some interference between simultaneous tasks even after extensive practice. In their careful analysis, they found that 50 ms was within the range of maximal interference for a "worst" alignment of the tasks, although they argued that values in the range of 20–40 ms are more likely.

ACT–R cannot predict no interference because interaction between modules must progress through the serial bottleneck of production execution. However, as discussed at the end of the article, there is evidence for direct module-to-module connections that do not go through the production system. ACT–R may need to be extended to include these. Ruthruff et al. (2003) referred to this with the analogy of "jumper cables" between stimulus and response. When there are such direct stimulus–response connections, overall behavior cannot be integrated, which is the theme of the article, but not all situations require such integration.

The Goal Module

Although human cognition is certainly embodied, its embodiment is not what gives human cognition its advantage over that of other species. Its advantage depends on its ability to achieve abstraction in content and control. Consider a person presented with the numbers 64 and 36. As far as the external stimulation is concerned, this presentation affords the individual a variety of actions—adding the numbers, subtracting them, dialing them on a phone, and so forth. Human ability to respond differently to these items depends on knowledge of what the current goal is and ability to sustain cognition in service of that goal without any change in the external environment. Suppose the goal is to add the numbers. Assuming that one does not already have the sum stored, one will have to go through a series of steps in coming up with the answer, and to do this, one has to keep one's place in performing these steps and keep track of various partial results such as the sum of the tens digits. The goal module has this responsibility of keeping track of what these intentions are so that behavior will serve that goal. It enables people to keep the thread of their thought in the absence of supporting external stimuli.

There are many different aspects of internal context, and it is unlikely that just one brain region maintains them all. Later, we describe research indicating that the posterior parietal cortex plays a major role in maintaining problem state. There is abundant research (Koechlin Corrado, Pietrini, & Grafman, 2000; Smith & Jonides, 1999) indicating that prefrontal regions also play an important role in maintaining the goal state (frequently called working memory), and Figure 1 associates the DLPFC with goal memory. A classic symptom of prefrontal damage is contextually inappropriate behavior such as when a patient responds to the appearance of a comb by combing his or her hair. DLPFC has also been known to track amount of subgoaling in tasks like Tower of London (Newman, Carpenter, Varma, & Just, in press) and Tower of Hanoi (Fincham, Carter, vanVeen, Stenger, & Anderson, 2002).

The Tower of Hanoi task (Simon, 1975) has been a classic paradigm for behavioral studies of goal manipulations. A number of the most effective strategies for solving this problem require that one keep a representation of a set of subgoals. Anderson and Douglass (2001) explicitly trained participants to execute a variant of what Simon (1975) called the sophisticated perceptual strategy in which one learns to set subgoals to place disks; thus, a participant might reason, "To move Disk 4 to Peg C, I have to move Disk 3 to Peg B, and to do this, I have to move Disk 2 to Peg C, and to do this, I have to move Disk 1 to Peg B." In this example, the participant had to create three planning subgoals (move 3 to B, move 2 to C, and move 1 to B). Behavioral studies such as that by Anderson and Douglass (2001) have shown that accuracy and latency is strongly correlated with the number of subgoals that have to be created.

In a follow-up to Anderson and Douglass (2001), Fincham et al. (2002) performed a study to determine what brain regions would respond to the number of goals that have to be created when a move is made. In that version of the task, a move had to be made every 16 s, and the brain was scanned in a 1.5-Tesla functional magnetic resonance imaging (fMRI) magnet every 4 s. Figure 4 shows the response of three regions that reflected an effect of number of goals that were set. Plotted there is the percentage difference between baseline and blood oxygen level dependent (BOLD) response for three regions—DLPFC, bilateral parietal

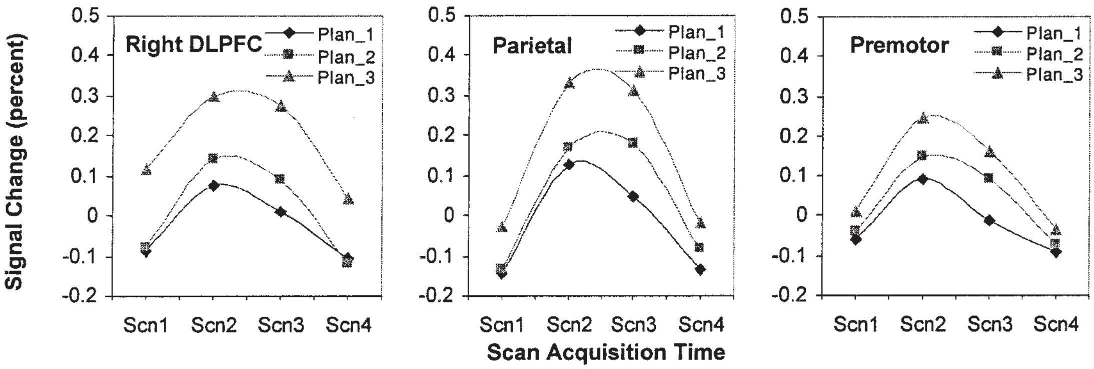


Figure 4. Three regions (left: right DLPFC; middle: parietal; right: premotor) responding to number of planning subgoals. DLPFC = dorsolateral prefrontal cortex; Scn = scan. From "Neural Mechanisms of Planning: A Computational Analysis Using Event-Related fMRI," by J. M. Fincham, C. S. Carter, V. van Veen, V. A. Stenger, and J. R. Anderson, 2002, *Proceedings of the National Academy of Sciences, USA, 99,* p. 3350. Copyright 2002 by the National Academy of Sciences, USA. Reprinted with permission.

regions, and the premotor cortex. We have more to say about such BOLD responses in a later section that reports an fMRI experiment, but for now, the important observation to make is that all three regions are showing a response to number of planning subgoals. This supports the conjecture that goal functions are maintained across multiple brain regions. The DLPFC region probably reflects general cognitive control. As we discuss more later, the parietal region is probably holding a representation of the problem. We have less often obtained premotor activation, but it may be related to the movement patterns that have to be planned in the Tower of Hanoi task. Fincham et al. described an ACT–R model that was used to identify these regions.

Given the cortical distribution of goal functions, one might wonder about the ACT–R hypothesis of a single goal structure. Indeed this is an issue under active consideration in the ACT–R community for many reasons. Many distinct goal modules may manage different aspects of internal state and project this information to the basal ganglia. There is no reason why the different parts of the information attributed to the goal cannot be stored in different locations nor why this information might not be distributed across multiple regions.

The Declarative Memory Module

Whereas the goal module maintains a local coherence in a problem-solving episode, it is the information stored in declarative memory that promotes things like long-term personal and cultural coherence. As a simple example, because most people know arithmetic facts such as 3 + 4 = 7, they can behave consistently in their calculations over time, and social transactions can be reliably agreed upon. However, access to information in declarative memory is hardly instantaneous or unproblematic, and an important component of the ACT–R theory concerns the activation processes that control this access. The declarative memory system and the procedural system to be discussed next constitute the cognitive core of ACT–R. Their behavior is controlled by a set of equations and parameters that will play a critical role in the integration examples to follow. Therefore, we give some space to discussing and illustrating these equations and parameters.

In a common formula in activation theories, the activation of a chunk is a sum of a base-level activation, reflecting its general usefulness in the past, and an associative activation, reflecting its relevance to the current context. The activation of a chunk i (A_i) is defined as

$$A_i = B_i + \sum_j W_j S_{ji}, \quad \text{(activation equation)}$$

where B_i is the base-level activation of the chunk i, the W_js reflect the attentional weighting of the elements that are part of the current goal, and the S_{ji}s are the strengths of association from the elements j to chunk i. Figure 5 displays the chunk encoding for 8 + 4 = 12 and its various quantities (with W_js for 4 and 8, assuming that they are sources). The activation of a chunk controls both its probability of being retrieved and its speed of retrieval.

We now unpack the various components of the activation equation. As for the associative components (the W_j and S_{ji}s), the attention weights W_j are set to $1/n$, where n is the number of sources of activation, and the S_{ji}s are set to $S - \ln(\text{fan}_j)$, where fan_j is the number of facts associated to term j. In many applications, S is estimated to be about 2. As for the base-level activation, it rises and falls with practice and delay according to the equation

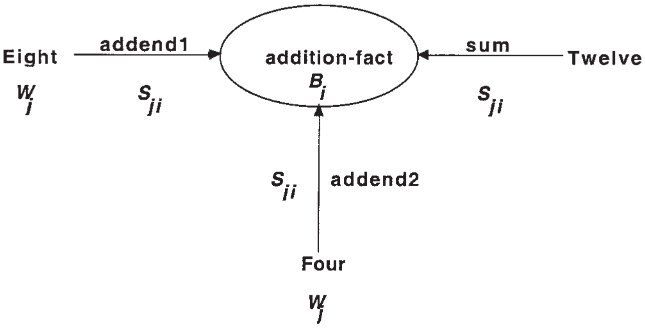


Figure 5. A presentation of a declarative chunk with its subsymbolic quantities. W_j = attentional weights; S_{ji} = strengths of association; B_i = base-level activation.

$$B_i = \ln\left(\sum_{j=1}^{n} t_j^{-d}\right), \quad \text{(base-level learning equation)}$$

where t_j is the time since the jth practice of an item. This equation is based on the rational analysis of Anderson and Schooler (1991), who studied how the pattern of past occurrences of an item predicts the need to retrieve it. They found that the above equation reflects the log odds an item will reoccur as a function of how it has appeared in the past. In developing ACT–R, we assumed that base-level activation would track log odds. Each presentation has an impact on odds that decays away as a power function (producing the power law of forgetting), and different presentations add up (it turns out producing the power law of practice; see Anderson, Fincham, & Douglass, 1999). In the ACT–R community, .5 has emerged as the default value for the parameter d over a large range of applications. This base-level learning equation has been the most successfully and frequently used part of the ACT–R theory.

There are two equations mapping activation onto probability of retrieval and latency. With respect to probability of retrieval, the assumption is chunks will be retrieved only if their activation is over a threshold. Because activation values are noisy, there is only a certain probability that any chunk will be above threshold. The probability that the activation will be greater than a threshold τ is given by the following equation:

$$P_i = \frac{1}{1 + e^{-(A_i - \tau)/s}}, \quad \text{(probability of retrieval equation)}$$

where s controls the noise in the activation levels and is typically set at about .4. If a chunk is successfully retrieved, the latency of retrieval will reflect the activation of a chunk. The time to retrieve the chunk is given as

$$T_i = Fe^{-A_i}. \quad \text{(latency of retrieval equation)}$$

Although we have a narrow range of values for the noise parameter s, the retrieval threshold, τ, and latency factor, F, are parameters that have varied substantially from model to model. However, Anderson, Bothell, Lebiere, and Matessa (1998) have discovered a general relationship between them, which can be stated as

$$F \approx 0.35e^{\tau},$$

which means that the retrieval latency at threshold (when $A_i = \tau$) is approximately 0.35 s. As we show, when we come to the integrated models, it is important to have strong constraints on parameter values of the model so that one is in position to make real predictions about performance.

Historically, the ACT theory of declarative retrieval has focused on tasks that require participants to retrieve facts from declarative memory. The second experiment in Pirolli and Anderson (1985) is a good one to illustrate the contributions of both base-level activations (B_i) and associative strengths (S_{ji}) to the retrieval process. This was a fan experiment (Anderson, 1974) in which participants were to try to recognize sentences such as *A hippie was in the park*. The number of facts (i.e., fan) associated with the person (e.g., *hippie*) could be either 1 or 3, and the fan associated with the location could be either 1 or 3. Participants practiced recognizing the same set of sentences for 10 days. Figure 6 illustrates how to conceive of these facts in terms of their chunk representations and subsymbolic quantities. Each oval in Figure 6 represents a chunk that encodes a fact in the experiment. As a concept like "hippie" is associated with more facts, there are more paths emanating from that concept, and according to ACT–R, the strengths of association S_{ji} will decrease.

Figure 7 illustrates how the activations of these chunks vary as a function of fan and amount of practice. There are separate curves for different fans, which correspond to different associative strengths (S_{ji}). The curves rise with increasing practice because of increasing base-level activation. Figure 8 illustrates the data from this experiment. Participants are slowed in the presence of greater fan but speed up with practice. The practice in this experiment gets participants to the point where high-fan items are recognized more rapidly than low-fan items were originally recognized. Practice also reduces the absolute size of the effect of fan, but it remains substantial even after 10 days of practice.

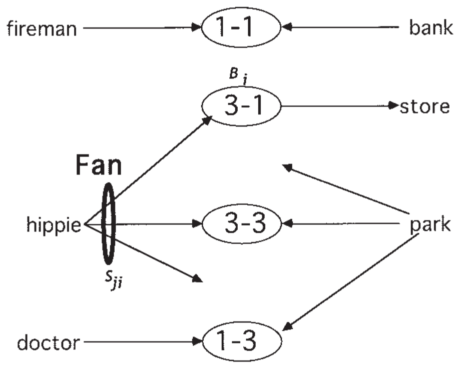


Figure 6. Representation of some of the chunks in Pirolli and Anderson (1985). S_{ji} = strengths of association; B_i = base-level activation.

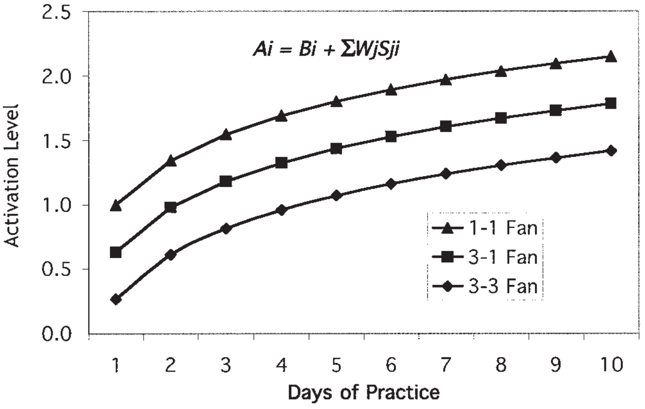


Figure 7. Activation of the chunks in Pirolli and Anderson (1985) as a function of fan and practice. A_i = activation; W_j = attentional weights; S_{ij} = strengths of association; B_i = base-level activation; 1-1 Fan = both person and location have one association; 3-1 Fan = person or location has one association and the other has three associations; 3-3 Fan = both person and location have three associations.

As reviewed above, the strength of association can be calculated by $S - \ln(\text{fan})$. Anderson and Reder (1999) used values of S around 1.5 in fitting the fan data, and this is the value used for fitting the data in Figure 8. The effect of practice is to increase the base-level activation of the facts. One can derive from the base-level learning equation that an item with n presentations will have an approximate base-level activation of $C + .5^*\ln(n)$, where C depends on presentation rate. Because C gets absorbed in the estimation of the latency factor F below, we just set it to 0. Figure 7 shows the activation values that are obtained from combining the base-level activation with the associative activation according to the activation equation, setting the weights, W_j, in this experiment to .333 (as used in Anderson & Reder, 1999, because each of the three content terms—*hippie*, *in*, *park*—in the sentence gets an equal 1/3 source activation). These are parameter-free predictions for the activation values. As can be seen, they increase with practice, with low-fan items having a constant advantage over high-fan items.

According to the ACT–R theory, these activation values can be mapped onto predicted recognition times according to the equation

$$\text{recognition time} = I + Fe^{-A_i},$$

where I is an intercept time reflecting encoding and response time, and F is a latency scale factor. Thus, fitting the model required

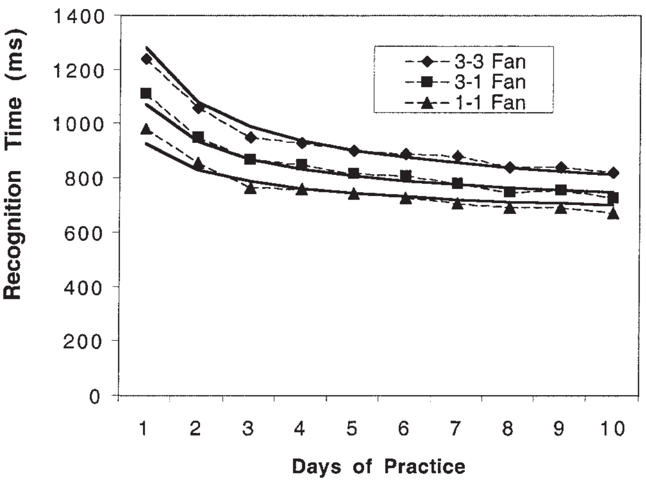


Figure 8. Time to recognize sentences in Pirolli and Anderson (1985) as a function of fan and practice. Solid curves reflect predictions of the ACT–R model. 1-1 Fan = both person and location have one association; 3-1 Fan = person or location has one association and the other has three associations; 3-3 Fan = both person and location have three associations.

estimating two parameters, and these were $I = 597$ ms and $F = 890$ ms, which are quite similar to the parameters estimated in Anderson and Reder (1999). The value of I is also quite reasonable as the time to encode the words and emit a response (keypress). The overall quality of fit is good with a correlation of .986. Moreover, this correlation does not depend on the parameter estimates I and F but only on e^{-A_i}, which means that it measures a prediction of ACT–R that does not depend on the estimation of the parameters I and F. The effect of I and F is only to scale this critical quantity onto the range of the latencies.

Although this example illustrates the ACT–R theory of declarative memory, it is by no means the only example. This part of the theory has been perhaps the most successful, enjoying applications to list memory (Anderson et al., 1998), implicit memory (Lebiere & Wallach, 2001), category learning (Anderson & Betz, 2001), sentence processing (Anderson, Budiu, & Reder, 2001), and individual differences (Lovett, Daily, & Reder, 2000), among other domains. The theory of declarative memory gives a natural account of the explicit–implicit distinction. Explicit memories refer to specific declarative chunks that can be retrieved and inspected. Implicit memory effects reflect the subsymbolic activation processes that govern the availability of these memories. This is substantially the same theory of memory as that of Reder and Gordon's (1997) SAC (source of activation confusion) theory.

Procedural Memory

As described so far, ACT–R consists of a set of modules that progress independently of one another. This would be a totally fragmented concept of cognition except for the fact that they make information about their computations available in buffers. The production system can detect the patterns that appear in these buffers and decide what to do next to achieve coherent behavior. The acronym ACT stands for adaptive control of thought, and this section describes how the production system achieves this control and how it is adaptive. The key idea is that at any point in time multiple production rules might apply, but because of the seriality in production rule execution, only one can be selected, and this is the one with the highest utility. Production rule utilities are noisy, continuously varying quantities just like declarative activations and play a similar role in production selection as activations play in chunk selection. The other significant set of parameters in ACT–R involve these utility calculations. The utility of a production i is defined as

$$U_i = P_iG - C_i, \quad \text{(production utility equation)}$$

where P_i is an estimate of the probability that if production i is chosen the current goal will be achieved, G is the value of that current goal, and C_i is an estimate of the cost (typically measured in time) to achieve that goal. As we discuss, both P_i and C_i are learned from experience with that production rule.

The utilities associated with a production are noisy, and on a cycle-to-cycle basis, there is a random variation around the expected value given above. The highest valued production is always selected, but on some trials, one might randomly be more highly valued than another. If there are n productions that currently match, the probability of selecting the ith production is related to the utilities U_i of the n production rules by the formula

$$P_i = \frac{e^{U_i/t}}{\sum_j^n e^{U_j/t}}, \quad \text{(production choice equation)}$$

where the summation is over all applicable productions and t controls the noise in the utilities. Thus, at any point in time there is a distribution of probabilities across alternative productions reflecting their relative utilities. The value of t is about .5 in our simulations, and this is emerging as a reasonable setting for this parameter.

Learning mechanisms adjust the costs C_i and probabilities P_i that underlie the utilities U_i according to a Bayesian framework. Because the example that we describe concerns learning of the probabilities, we expand on that, but the learning of costs is similar. The estimated value of P is simply the ratio of successes to the sum of successes and failures:

$$P = \frac{Successes}{Successes + Failures}, \quad \text{(probability of success equation)}$$

However, there is a complication here that makes this like a Bayesian estimate. This complication concerns how the counts for successes and failures start out. It might seem natural to start them out at 0. However, this means that P is initially not defined, and after the first experience the estimate of P will be extreme at either the value 1 or 0, depending on whether the first experience was a success or failure. Rather, P is initially defined as having a prior value θ, and this is achieved by setting successes to $\theta V + m$ and failures to $(1 - \theta)V + n$, where m is the number of experienced successes, n is the number of experienced failures, and V is the strength of the prior θ. As experience $(m + n)$ accumulates, P will shift from θ to $m/(m + n)$ at a speed controlled by the value of V.

The simplest example we can offer of the utility calculations at work is with respect to probability learning. We describe here an application to one condition of Friedman et al. (1964) in which participants had to guess which of two buttons would light up when one of the buttons had a 90% probability and the other 10%. Figure 9 shows the results from the experiment in terms of mean probability that participants would guess the more probable light

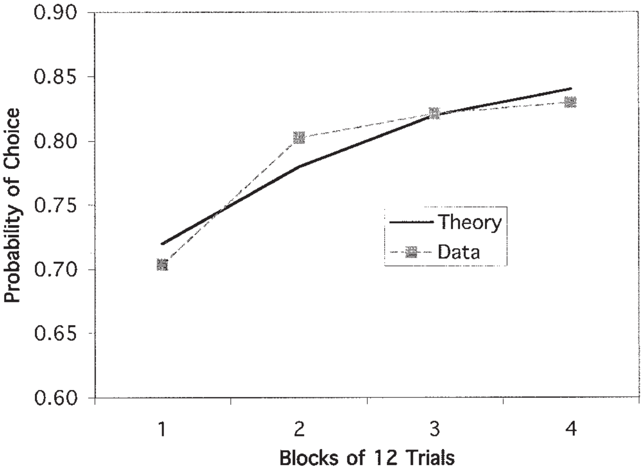


Figure 9. Predictions of the ACT–R utility learning mechanism for the experiment of Friedman et al. (1964).

for four successive blocks of 12 trials and the predictions of the ACT–R model. In this model, there were two production rules that competed, one for each light. The two rules started out with equal expected cost C and equal expected probability of success P. With time, however, the probability of the more successful production increased, and the probability of the less successful one decreased. It can be shown that a consequence of the previous equations is that the probability P_1 that Button 1 will be chosen is

$$P_1 = \frac{1}{1 + e^{(P_2 - P_1)G/t}},$$

where G is the value of the goal, t is the utility noise, and P_1 is the estimated probability of success for Button 1 and P_2 is the estimated probability of success for Button 2. According to the formulas given earlier, the estimated probability P_1 will be

$$P_1 = \frac{\theta V + m_1}{V + m_1 + n_1},$$

where θ is the prior, V is its strength, m_1 is the number of experienced successes, and n_1 is the number of experienced failures. An analogous formula applies for P_2. We set $\theta = .5$ and $V = 2$, which are the uninformed priors (Berger, 1985), leaving only G/t to be estimated, and this was estimated to be 2.25.

Lovett (1998) gave a much more thorough account of a wide range of choice learning tasks, but this simple example does illustrate how the utility learning mechanisms in ACT–R produce the kinds of probabilistic behavior observed in people and how that changes with experience. One of the issues that Lovett discussed is probability matching, which is the phenomenon that in many situations people's probability of a choice approximately matches the probability that this choice will be successful. The model in Figure 9 was still learning but asymptotically would have reached a .86 probability of choosing the alternative that was successful with a probability of .90.

In the previous examples, the productions that would do the task were prespecified. These essentially amount to degrees of freedom in constructing a model, although in the case of something like probability matching it is pretty obvious what the production rules have to be. We could eliminate these degrees of freedom if we could specify the process by which these production rules were learned. Taatgen and Anderson (2002) have developed a production learning mechanism for ACT–R called production compilation, which shows considerable promise. It bears some similarity to the chunking mechanism in Soar (Newell, 1990) and is basically a combination of composition and proceduralization as described in Anderson (1983) for ACT*. Production compilation will try to take each successive pair of productions and build a single production that has the effect of both. There are certain situations in which this is not possible, and these involve the perceptual-motor modules. When the first production makes a request for a perceptual encoding or motor action and the second production depends on completion of this request, then it is not possible to compile the two productions together. For instance, it is not possible to collapse a production that requests a word be encoded and another production that acts on the encoding of this word (or else ACT–R would wind up hallucinating the identity of the word). Thus, the perceptual and motor actions define the boundaries of what can be composed. An interesting case concerns when the first production rule requests a retrieval and the second harvests it. The resulting production rule is specialized to include the retrieved information.

Production compilation can be illustrated with respect to a simple paired-associate task. Suppose the following pair of production rules fire in succession to produce recall of a paired associate:

IF reading the word for a paired-associate test
  and a word is being attended,
THEN retrieve the associate of the word.

IF recalling for a paired-associate test
  and an associate has been retrieved with response *N*,
THEN type *N*.

These production rules might apply, for instance, when the stimulus *vanilla* is presented: A participant recalls the paired associate *vanilla–7* and produces *7* as an answer. Production compilation collapses these two productions into one. To deal with the fact the second production rule requires the retrieval requested by the first, the product of the retrieval is built into the new production. Thus, ACT–R learns the following production rule:

IF reading the word for a paired-associate test
  and *vanilla* is being attended,
THEN type *7*.

This example shows how production rules can be acquired that embed knowledge from declarative memory.

After a production New is composed from productions Old1 and Old2, whenever New can apply, Old1 can also apply. The choice between New, Old1, and whatever other productions might apply will be determined by their utilities. However, the new production New has no prior experience, and so, its initial probabilities and costs will be determined by the Bayesian priors. We describe how the prior θ is set for P, noting a similar process applies for C. When New is first created, θ is set to be 0. Thus, there is no chance that the production will be selected. However, whenever it is recreated, its θ value is incremented according to the Rescorla–Wagner (Rescorla & Wagner, 1972) or delta rule: $\Delta\ \theta = a(P - \theta)$, where P is the probability of Old1. Eventually, if the production rule New is repeatedly created, its priori θ will converge on P for the parent Old1. The same will happen for its cost, and it will be eventually tried over its parent. If it is actually superior (the typical situation is that the new production has the same P but lower C), it will come to dominate its parent. Although our experience with this production rule learning mechanism is relatively limited, it seems that a working value of the learning rate a is .05.

Putting It All Together: The Effects of Instruction and Practice in a Dynamic Task

Having described the components of the ACT–R theory, we now turn to discussing how they work together to contribute to modeling complex real-world tasks (this section) and integrating brain imaging data (next section). Pew and Mavor (1998) reviewed some of the practical needs for cognitive architectures. One of these is in training and educational applications (Anderson, 2002), and the application we describe here has this as its ultimate motivation. Other domains include development of synthetic agents (Freed, 2000) and human–computer interaction (Byrne, 2003). Such applications do not respect the traditional divisions in cognitive psychology and so require integrated architectures. For instance, high school mathematics involves reading and language processing (for processing of instruction, mathematical expressions, and word problems), spatial processing (for processing of graphs and diagrams), memory (for formula and theorems), problem solving, reasoning, and skill acquisition.

However, such applications pose an additional requirement beyond just integrating different aspects of cognition. These models have to predict human behavior in situations in which they have not been tuned. As such, these are demanding tests of the underlying theory. Most tests of models in psychology, including the just-presented Figures 8 and 9, involve "postdiction." Data are collected, a set of parameters is estimated, and the model is judged according to how well it fits the data. However, many applications want predictions in advance of getting the data. For instance, it is very expensive to run educational experiments and sort out the multitude of instructional treatments that might be proposed. One wants to predict what will be the effective instructional intervention and use that. As another example, when the military uses synthetic agents in training exercises (e.g., Jones et al., 1999), it simply cannot create the real war situation in advance to provide data to fit. This need for true prediction is the reason for the concern in the preceding sections with fixing parameter values.

However, besides the numerical values, there is another kind of parameter that is not typically accounted for in modeling. This is the structure of the model itself. In traditional information-processing psychology, this takes the form of different flowchart options. In neural network models, this takes the form of different topologies and representations. In the ACT–R models, this takes the form of assumptions about the chunks and productions in a model. What we would like is to have a system that takes the instruction for a task and configures itself. The application that we describe here comes close to accomplishing just this. Its limitation is that it does not process full natural language but rather accepts only a restricted instructional format. These instructions are converted into a declarative representation. We have developed a set of production rules that will interpret any such instruction set. The production compilation mechanism will eventually convert these instructions into a set of productions for directly performing the task without declarative retrieval of the instructions. Thus, this system can configure itself to do any task. This approach also accounts for one of the mysteries of experimental psychology, which is how a set of experimental instructions causes a participant to behave according to the experimenter's wishes. According to this analysis, during the warm-up trials, which are typically thrown away in an experiment, the participant is converting from a declarative representation and a slow interpretation of the task to a smooth, rapid procedural execution of the task.

The Dynamic Task: The Anti-Air Warfare Coordinator (AAWC)

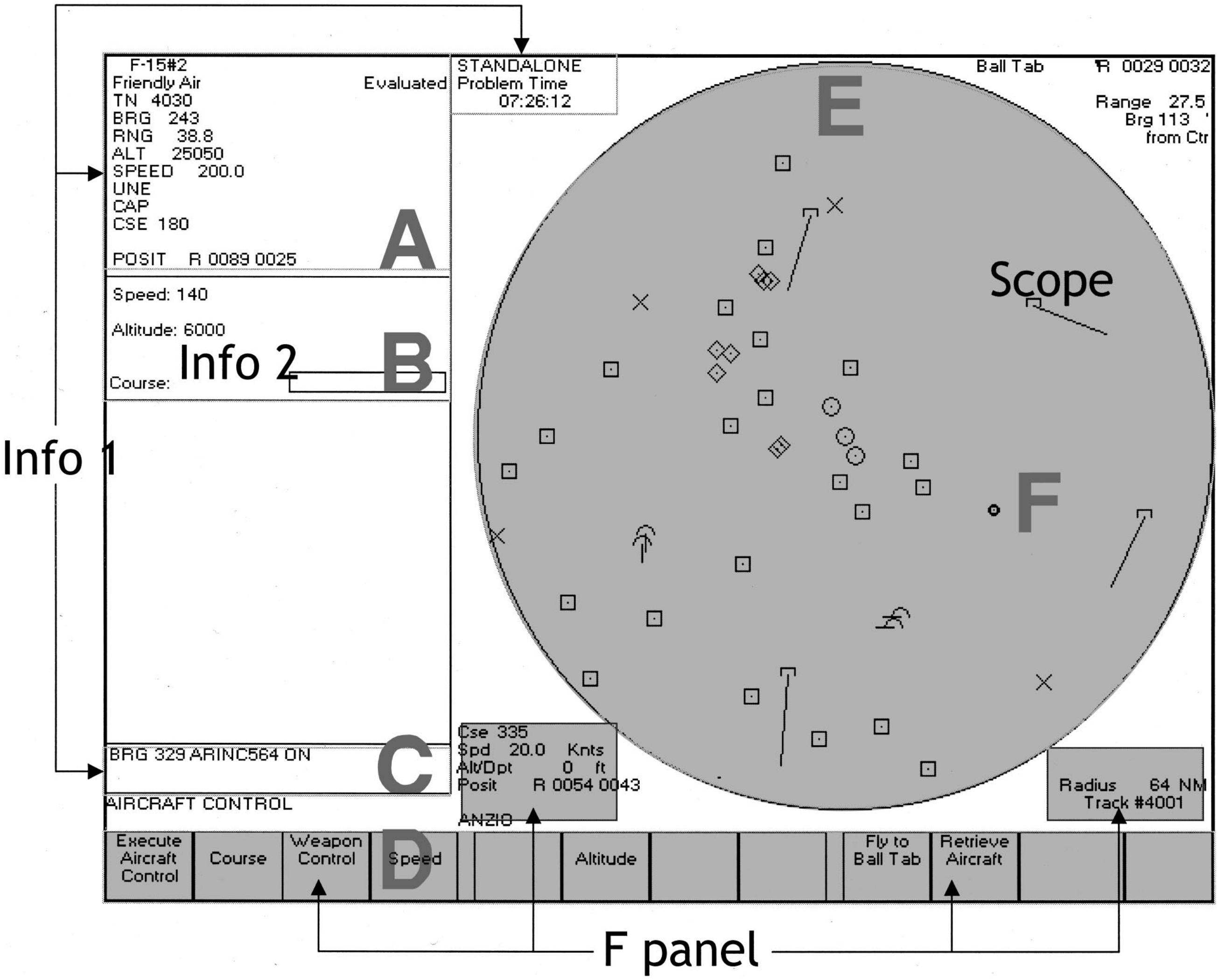
Dynamic tasks like air traffic control are ideal domains for testing integration of modules. They involve strongly goal-directed processing that must play itself out in the presence of demanding perceptual displays. There is often extreme time pressure, which puts severe constraints on all aspects of the architecture including the details of motor execution. There is a rich body of declarative knowledge that needs to be brought to bear in the performance of the task. A great deal of practice is needed. The project we are involved in has as its ultimate goal to provide real-time instruction and coaching in such tasks. We have done a series of experiments in which all the instruction is given in advance, and we observe how the participants improve with practice on the task.

The task we have been working with is the Georgia Tech Aegis Simulation Program (GT-ASP; Hodge et al., 1995). GT-ASP is a tactical decision-making computer game that simulates tasks facing an AAWC on board U.S. Navy cruisers and destroyers. A participant assumes the role of an AAWC, which includes monitoring a radar screen for unknown aircraft, requesting and collecting information regarding the unknown aircraft, and updating the identity of the aircraft. GT-ASP is like the system that is currently used in the Navy but reduces much of the complexity. Sohn, Douglass, Chen, and Anderson (in press) described the general behavioral characteristics associated with learning this task. Here, we report on a somewhat simpler version of the original system that we have attempted to model in great detail.

The radar screen of the GT-ASP task (see Figure 10) consists of three major areas. First, the radarscope shows various air tracks. Vectors emanating from the aircraft indicate speed and course. The AAWC moves the mouse within the scope and "hooks" a target airplane by clicking the mouse button. This hooking is necessary whenever the AAWC tries to update identity of unknown aircraft. Second, there is a group of information boxes on the left of the screen where the participant can get information on tracks. Third, the menu panel shows the current bindings of the function keys (F1–F12 on the computer keyboard) that are used to issue commands. As in the shipboard system, the meaning of these keys changes depending on where one is in the task. The dynamically changing binding of the function keys is a critical feature of the task. One of the important aspects of learning is coming to know what function key serves the desired function without having to search the menu panel.

Although the primary responsibility of the AAWC is to assure that rules of engagement are followed to protect home ship, the majority of the time spent in service of that goal involves identifying the intent (friendly or hostile) of tracks on the screen and their airframe type (e.g., helicopter, strike, or commercial). It is this identification task that the experiment focuses on. Figure 11 illustrates the decomposition of an identification task into five functional subtasks, each with its defining actions. There is the selection phase in which the AAWC searches the screen for an appropriate track to identify and concludes with a mouse hook of the target aircraft. Then, in the search subtask, the AAWC gathers information about the target unknown aircraft. In our version of the task, there are two sources of information for a classification. The hooked plane may display in the character read-out area (upper left in Figure 10) the speed and altitude indicating that it is a commercial airliner, and if so, it can be immediately classified as such. Alternatively, the AAWC may request the electronic warfare signal (EWS) of the plane, which if available in the track reporting area (lower left) will identify the frame. Once the AAWC has found the necessary information in the search subtask, the AAWC selects the air-track managing mode from the top-level menu by pressing the F6 key.

The remaining three subtasks are relatively more motor and less cognitive. We sometimes collapse these into a single execute subtask. After pushing the F6 key, the AAWC executes two keystrokes (F2 and F9) to choose the updating mode from other modes available under the track-managing mode, the initiate subtask. These two keystrokes do not vary depending on the type of


F-15#2
Friendly Air Evaluated
TN 4030
BRG 243
RNG 38.8
ALT 25050
SPEED 200.0
UNE
CAP
CSE 180

POSIT R 0089 0025

Speed: 140

Altitude: 6000

Course:

BRG 329 ARINC564 ON

AIRCRAFT CONTROL

STANDALONE
Problem Time
07:26:12

Ball Tab R 0029 0032

Range 27.5
Brg 113
from Ctr

Cse 335
Spd 20.0 Knts
Alt/Dpt 0 ft
Posit R 0054 0043
ANZIO

Radius 64 NM
Track #4001

Figure 10. The screen layout in the Georgia Tech Aegis Simulation Program synthetic task. Shaded areas are the on-task regions where the ACT–R model looks. The character readout box (Region A) provides available information of the currently hooked air track. The character type-in box (Region B) shows the numbers entered in to change flight profiles of a Combat Air Patrol. The message box (Region C) shows the radar signal or the visual identification of an air track. The menu panel (Region D) shows the currently available function keys and their labels. The scope (Region E) shows all the air tracks and surface tracks. The balltab (Region F) is the region surrounding the currently hooked air track.

identification. The classify subtask requires four keystrokes, which differ depending on the correct identity of the aircraft. The AAWC first indicates the information to update by pressing the F4 key for the primary intent or the F7 key for the air type. On pressing either F4 or F7, the menu panel provides several choices from each category. For example, there are four kinds of primary intent and 10 kinds of air type. Therefore, the classify subtask imposes somewhat different demands from those of the initiate subtask

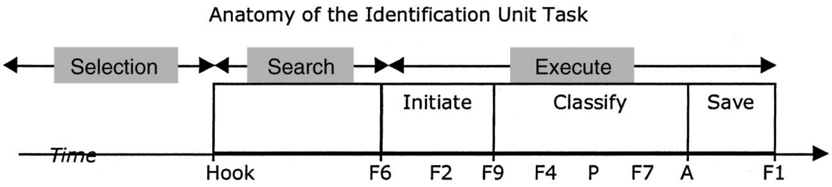


Figure 11. The anatomy of an identification unit task. P stands for the keystroke corresponding to the primary identification, and A stands for the keystroke corresponding to the air identification. There is the option of the F10 key followed by F1 during the search interval to get electronic warfare signal information.

because it involves choosing keys from multiple choices. The save subtask requires only one keystroke, the F1 key, to finalize and confirm the updating.

The Experiment

Sixteen participants were recruited for a 2-day experiment. On the first day, they were given the instructions that are summarized by the 13 rules in the Appendix. With each rule in the Appendix is the translation of that instruction into the stilted text that the model could process (this translation was not given to the participants). The instructions take goals unique to the task (identify tracks, identify a track, confirm planes are flying at commercial altitudes, request an EWS identity, record a primary and air ID, etc.) and decompose these into goals that we assume the participant already knows how to achieve (e.g., clicking, reading, hitting keys) and that the model already can do.

On the first day, participants memorized the instructions and were tested on ten 6-min scenarios. On the second day, they were tested on 10 more. Each scenario involved 40 tracks randomly placed on the screen. Twenty of these tracks satisfied a commercial profile and so could be categorized on this basis, and 22 gave EWS signals and could be classified on this basis. These two sets intersected such that 12 could be classified on either basis and 8 tracks could not be classified on either basis. Of the 32 tracks that could be identified, participants improved from an average of 12.1 identifications on the first scenario to an average of 30.4 on the last scenario.

To provide further data about the behavior of the participants, we tracked their eyes during the performance of the task. We used an ETL-500 video-based, head-mounted eye-tracking system with magnetic-based head tracker from ISCAN (Cambridge, MA). The software for collecting and analyzing eye data consisted of the EPAL (Douglass, 1998) software suite that was internally developed in our lab to facilitate the development of eye-tracking experiments and their analyses.

The Model

The model we developed to perform this task was given the instructions in the Appendix. The following is a review of the parameter settings of the model:

1. The cycle time was set at 50 ms, which is an emerging generalization in production system architectures.

2. Motor times were adopted from the EPIC parameters.

3. The visual encoding time (time to encode an object) was set at the default value of 85 ms.

4. The decay parameter, *d*, for decay of declarative information was set at the default of .5.

5. The activation noise parameter, *s*, was set to a typical value of .4.

6. As we noted previously, we find it necessary to estimate on a task-by-task basis the retrieval threshold, τ, and in this experiment, we estimated it to be 1.0. This controlled rate of location learning, as we describe below.

7. Given the threshold parameter, we set the latency threshold parameter to be $F = 1.0$, which satisfies the constraint given earlier ($F = e^{-\tau} = .368$, which is approximately 0.35 s).

8. The *t* parameter controlling expected gain noise was set to be a typical value of .5.

9. The strength of the priors, *V*, was set to be 10, which is the tentative default value.

10. The learning rate *a* for production rules was set to be .05, which is the tentative default value.

11. We estimate 200 ms for the model from reading a term like *track manager* to obtaining a representation of its meaning.

Thus, the only parameters that were estimated to fit the data were the retrieval threshold (6 above) and the comprehension time (11 above). We should also mention here an idealization in the model for this task. We assumed that participants had perfect access to the initial instructions resulting in no retrieval failures or latencies for the instructions. The instructions were well practiced and errors were few, but not to this extent.

The Speed Up in Task Performance

Figure 12 compares the model and participants in terms of the speed up in their performance of each of the five functional subgoals in the Figure 11 task decomposition. Although not perfect, the correspondence is quite good. To give some idea of why these trends are happening, we give a detailed analysis of the learning changes happening in Figure 12E, which concerns the interval between the pressing of the key indicating the air ID and the pressing of the F1 key, which is the last keystroke. This interkey latency starts out at 1.738 s and ends at 0.300 s in the model.

Figure 13 compares the timing of the first save F1 keypress (top) and of a late F1 save keypress (bottom). In both cases, corresponding to Instruction 11 in the Appendix, the model is pressing a sequence of keys. Our interest is in processing between the key just before F1 (an F9 key indicating a nonmilitary airframe in both cases) and the F1 save command. At the beginning (see the top panel of Figure 13) corresponding to Instruction 12, the command is implemented as a search for the item in the bottom menu followed by a press on the corresponding function key. Corresponding to Instruction 13, the search of the bottom menu is implemented as a search that starts at the bottom left and progresses to the right. (Note in the case of the save command, because its function key [F1] is the first menu item, this search will be brief.) By the end (see the bottom panel of Figure 13), the commands directly map onto the key actions without the intermediate goals. The times at which each goal is set and keys pressed are given in Figure 13.

In the top panel of Figure 13, the F9 key is hit 250 ms after selection of the hit-key goal, and the F1 key is hit 400 ms after selecting of its goal. These times reflect 50 ms for the production to fire, 50 or 200 ms to set the movement features, 50 ms to initiate the key, and 100 ms to complete the press action (all the motor parameters are taken from EPIC). The F1 key takes 150 ms more than the F9 key because the key previous to F1, which is F9, was hit with the other hand, and additional features have to be pre-

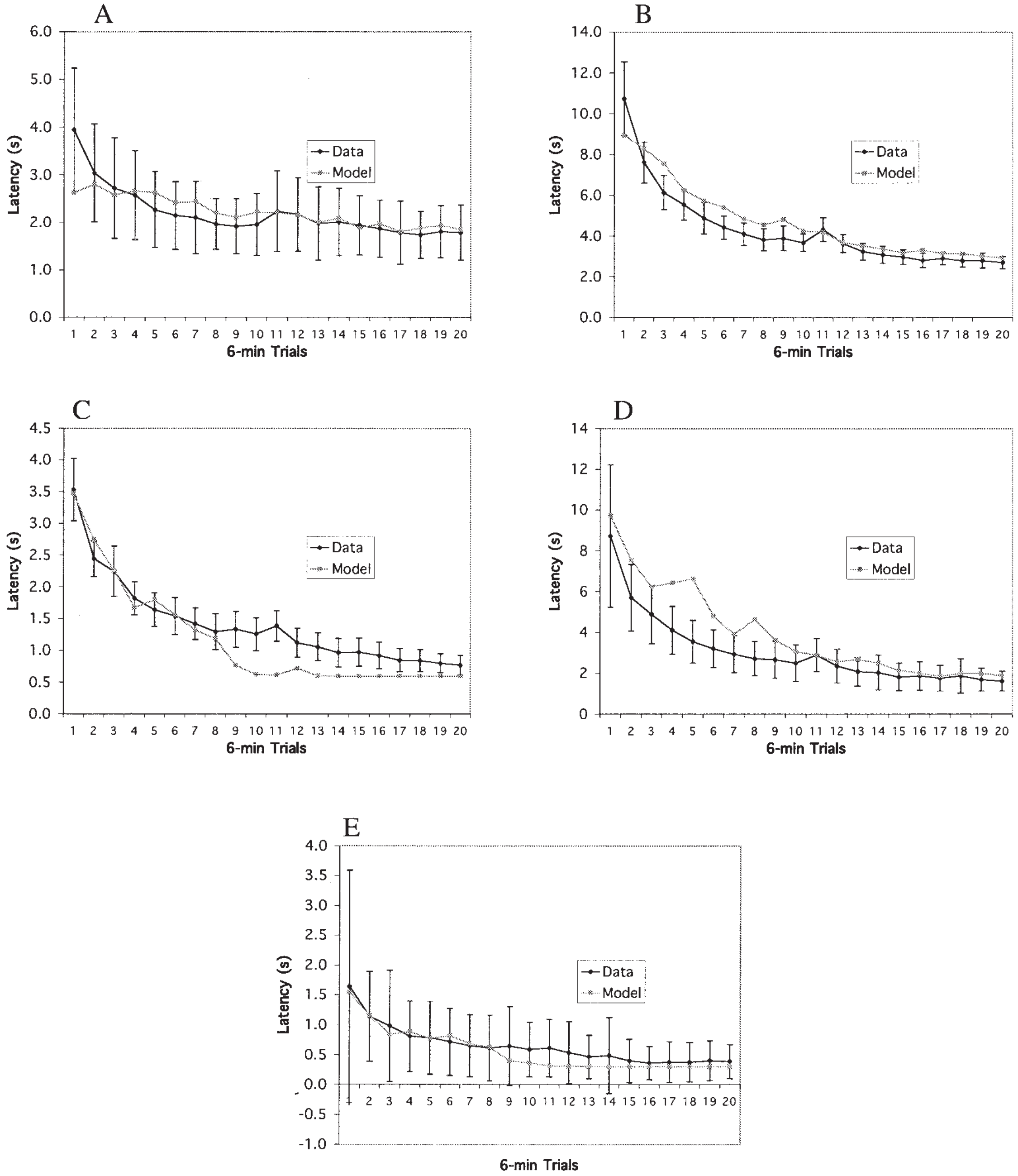


Figure 12. Time for various components of the task (see Figure 11). A: Selection. B: Search. C: Initiation. D: Classification. E. Save. Error bars represent one standard deviation of the means.

pared. The 200 or 350 ms after the production rule has fired can progress in parallel with the selection of other productions. This is why, for instance, the F9 key is pressed after the select-save subgoal is set. To understand the overall 1.738 ms between keystrokes, then, we need to understand the 1.588 ms between the setting of the hit-F9 subgoal and the hit-F1 subgoal. We can break this up into the five transitions among the goals involved:

1. "Hit the F9 Key" to "Select Save" (150 ms). This reflects the time for three productions to fire—one that calls for the hitting of the F9 key, a production that returns to the parent goal, and a production that retrieves the next step from the instructions.

2. "Select Save" to "Find Save on the Menu" (150 ms). This reflects the time for three productions to fire—one that decides to use the instructions, one that retrieves Instruction 12 for selecting a key, and one that retrieves the first step of this instruction.

3. "Find Save on the Menu" to "Look to the Lower Left" (518 ms). The first production to fire takes 50 ms and

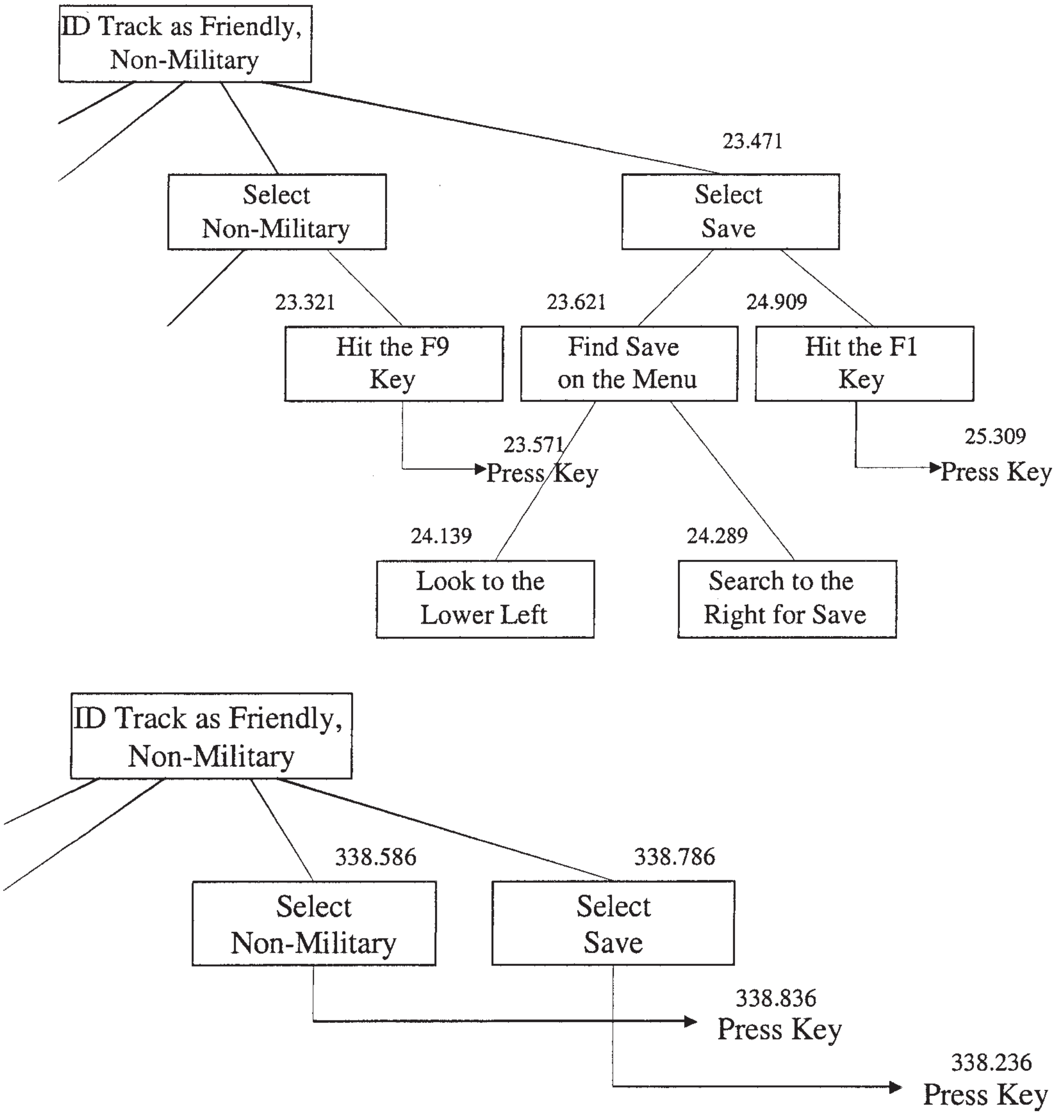


Figure 13. Goal structures and times controlling the model's interkey latency at the beginning of the experiment (top) and the end (bottom). Note the goals and actions in this figure are labeled with the actual times that they occurred in the 360-s scenario.

tries to retrieve the menu location of 'Save." The retrieval process times out 368 ms later with a failure. After this, two productions fire, one to retrieve Instruction 13 for finding locations and one to retrieve the first step of this instruction.

4. "Look to the Lower Left" to "Search to the Right for Save" (150 ms). This reflects the time for three productions to fire—one to retrieve where the lower left of the screen is, one to find the location, and one to request that attention be switched to the menu button at that location.

5. "Search to Right for Save" to "Hit the F1 Key" (620 ms). This sequence begins with 85 ms to encode the leftmost menu button and then a production fires (50 ms) that switches attention to the text object on that menu button (another 85 ms). It then takes an additional 200 ms to interpret the text (a parameter of the model). Then four productions fire—one that recognizes the target button has been found, two more to retrieve the two levels of the goal structure, and one to retrieve the next instruction.

The system slowly learns the menu locations and the locations of other information. Each time a menu location is discovered, memory of its location is strengthened, and with time, the menu locations can be retrieved without search. The retrieval threshold determines how fast these locations are learned. Also being learned are the locations of the altitude and speed used to identify commercial airlines.

When the system can retrieve the location of the *save* key, it will stop expanding the "Find Save on the Menu" into the two subgoals of "Look to the Lower Left" and "Search to the Right for Save."

Then, production compilation turns all the steps under "Select Save" into a single request for the F1 key. At this point we have the situation in the bottom panel of Figure 13 in which there is a direct transition between the two select goals with a lag of 200 ms. Only two productions fire between these goals—one that requests the keypress and one that changes goals—but the first must wait 100 ms until the motor programming from the prior step is complete. Although the goals are only 200 ms apart, the actual keystrokes are 300 ms apart. This reflects the minimum time between keystrokes—150 ms for the finger to return, 50 ms for the next press to be initiated, and 100 ms for the finger to strike the key. Thus, limitations at this point have become purely motor.

Figure 14 shows the overall speed up of the simulation compared with the participants' average, and the correspondence is reasonably close. The figure also indicates how much of the improvement is due to production rule learning and how much is due to location learning. This was achieved in simulations that disabled one or the other of these two mechanisms We also provide the data from a simulation that has both turned off. Figure 14 indicates that both production compilation and location learning are major contributors to the overall learning.

Eye Movements

We also performed an analysis of how well the eye movements of the participants corresponded to switches of attention in the model. A significant proportion (51%) of participants' eye movements were to regions of the screen that contained no information or were off screen altogether. The model never shifts attention to such locations, although there are many times when the model is not occupied encoding information from the screen and would be free to look elsewhere. We decided to use a measure of the relative proportion of task-relevant fixations and compare it with the model. Figure 10 illustrates the three regions of relevance: radar scope, the info regions that contained information relevant to identifying planes, and the function panel at the bottom of the screen.

Figure 15 shows the proportion of time spent viewing these regions as a function of trial for the selection phase before the hook (A), the search phase between the hook and the classification (B), and the execution of the classification (C). With respect to the selection phase, the participants and the model spend the majority of their time looking at the screen and the least time looking at the menu for the function keys. The model shows some change across time in the proportion of time it spends on various regions, coming by the end of the experiment to better match the participants. Figure 15B illustrates the proportion of time in the various regions in the information-gathering time. The model and participants spend most of their time looking at the information sources. The correspondence between the model and the participants is really quite striking. Figure 15C illustrates the proportion of time in the various regions during execution. Now the majority of time is spent fixating on the function menu giving the key identities, and this proportion tends to go down as the location of these keys becomes memorized. The correspondence between theory and model is again quite good. The major qualitative discrepancy is that the model shows a much more substantial increase in the proportion of time spent fixating the scope.

Summary

Given the few parameters estimated, the correspondence between model and average data in Figures 12 and 15 is quite compelling. This relative success depends on the integration of all the components of the ACT–R cognitive architecture:

1. Much of the timing rested on the perceptual-motor parameters. Again we note that the manual parameters were inherited from EPIC.

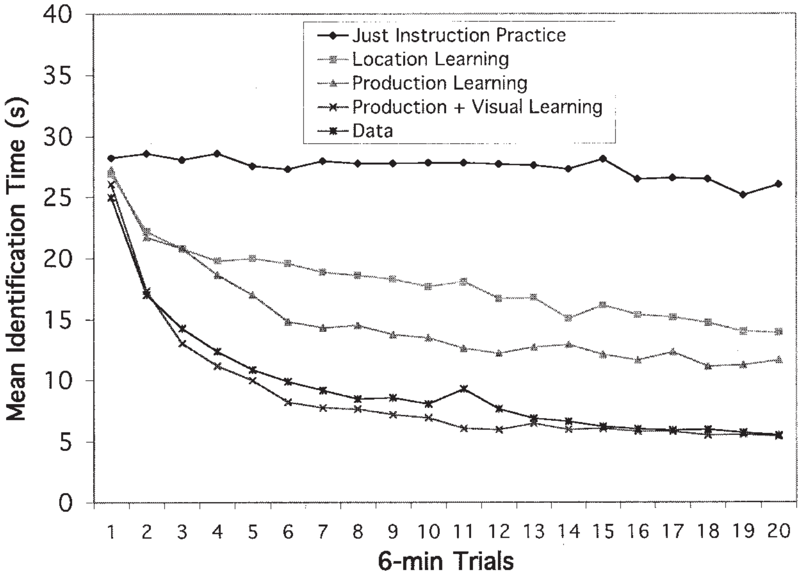


Figure 14. Overall time of participants to perform a unit task and the times taken by the model with various types of learning enabled.

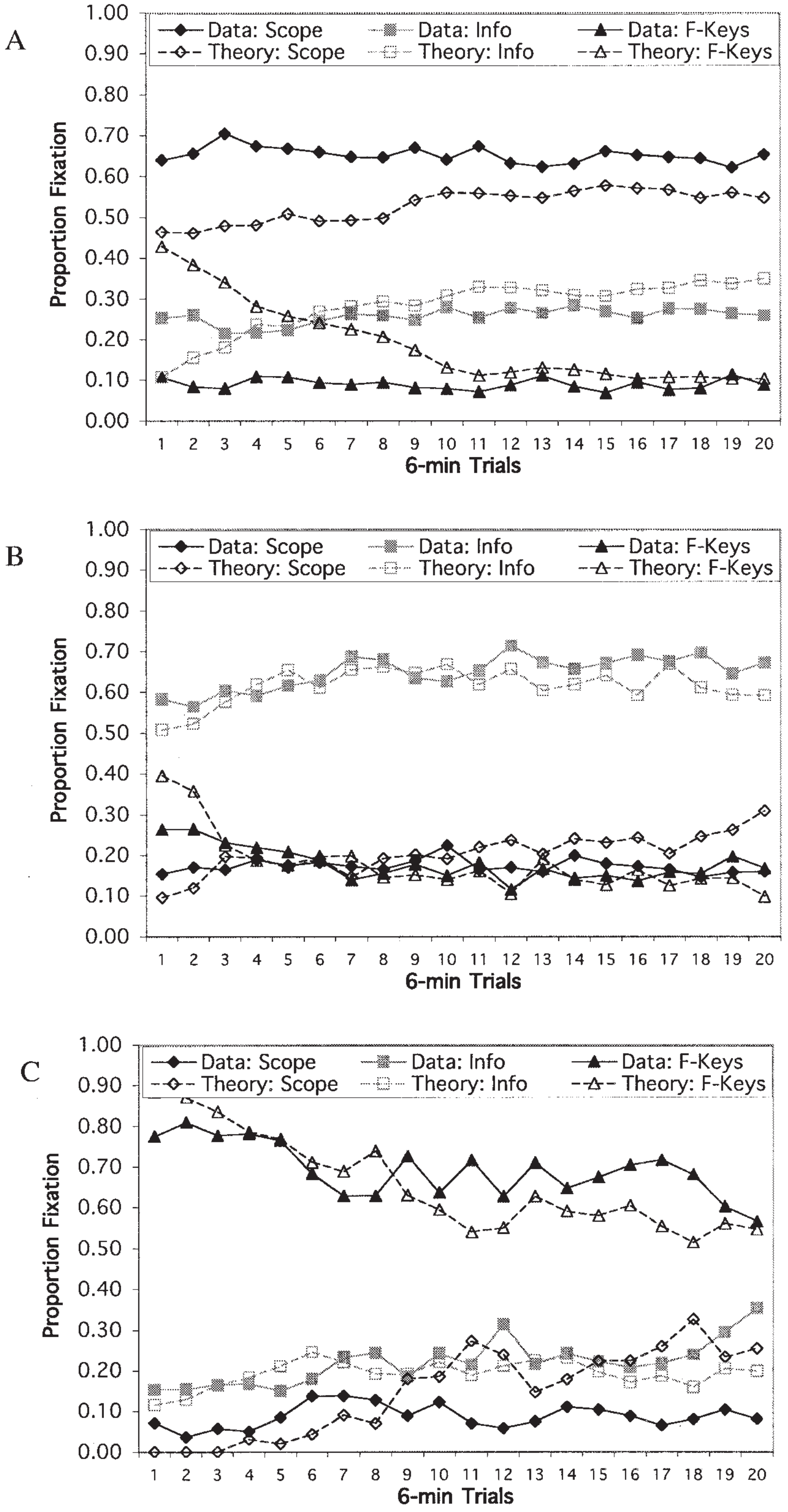


Figure 15. Comparison of proportion of time spent by the participants and the model fixating the three regions of interest. A: Selection time. B: Search time. C: Initiation + execution + save time. F-Keys = function keys.

2. The overall control structure depends on the goal module and transition among goals as revealed in Figure 13.

3. One of the significant aspects of the learning was location learning. This depended on the declarative memory module that eventually committed to memory the locations.

4. The other significant aspect of the learning depended on the procedural memory component and its production rule learning.

Although cognitive processes were the major time cost at the beginning, the end performance of the model largely depended on these perceptual-motor parameters. Basically, the current ACT–R model instantiates the shift from cognitive to perceptual motor factors that Ackermann (1990) claimed about ACT some time ago.

Putting It All Together: Tracking Multiple Buffers in an fMRI Study

The previous example discussed how the perceptual-motor, goal, declarative, and procedural modules interacted in the learning of a complex skill. This second example tracks their activity in a brain imaging study. One of the goals of this research is to find neural anchors for the concepts in the ACT–R model. The reason for attempting this is not to prove ACT–R correct but to acquire new sources of data to guide the development of the theory. Although the previous example of relatively successful behavior predictions is compelling at one level, many readers are no doubt ill at ease at the indirection between the behavior measures as in Figure 12 and the ascriptions of goal structures as in Figure 13. We are uncomfortable too and would like to gather more proximal measures of the activity of various architectural modules. Therefore, we have been collecting fMRI brain activity as participants do various tasks that have been modeled in ACT–R.

The methods we are using have contributions to make beyond guiding ACT–R development. Although perhaps a modeling effort like in the previous section is suffering from too much theory and too little data, the opposite is true of cognitive neuroscience. A typical brain imaging study, for instance, will find activation in a large number of regions with little basis for being able to judge which of those activations may be significant and which may be spurious or how they relate. We describe some methods that can be used to place such brain imaging data rigorously in the interpretative framework of an information-processing theory like ACT–R.

The Experiment

There have been a number of studies of adults and adolescents solving algebraic equations (Anderson, Qin, Sohn, Stenger, & Carter, 2003; Qin et al., 2004; Sohn et al., 2004). However, here we describe an artificial algebra task (Qin et al., 2003) that involves the same ACT–R modules and brain regions but that allows us to track learning of a symbol-manipulation skill from its beginning. Participants in this experiment were performing an artificial algebra task (based on Blessing & Anderson, 1996) in which they had to solve "equations." To give an illustration, suppose the equation to be solved were

②P③4 ↔ ②5,

where solving means isolating the P before the ↔. In this case, the first step is to move the ③4 over to the right, inverting the ③ operator to a ② so that the equation now looks like

②P ↔ ②5②4.

Then, the ② in front of the P is eliminated by converting ②s on the right side into ③s, so that the "solved" equation looks like

P ↔ ③5③4.

Participants were asked to perform these transformations in their heads and then key out the final answer—this involved keying 1 to indicate that they have solved the problem and then keying 3, 5, 3, and 4 on this example. The problems required zero, one, or two (as in this example) transformations to solve. The experiment looked at how participants speed up over 5 days of practice. Figure 16 shows time to hit the first key in various conditions as a function of days. The figure shows a large effect of number of transformations but also a substantial speed up over days.

Participants were imaged on Days 1 and 5. Qin et al. (2003) reported the activity for the three cortical regions illustrated in Figure 17. That figure shows a prefrontal region that tracks the operations in a retrieval buffer, a motor region that tracks operations in a manual buffer, and a parietal region that tracks operations in an imaginal buffer (part of the goal buffer) that holds the problem representation. All three regions are in the left cortex. Each region was defined as 100 voxels of 5 voxels wide × 5 voxels long × 4 voxels deep, approximately 16 × 16 × 13 mm^3. The centers of these regions, given in the figure caption, were based on previous work (Anderson et al., 2003; Qin et al., 2004; Sohn et al., 2004).

Participants had 18 s for each trial. Figure 18 shows how the BOLD signal varies over the 18-s period beginning 3 s before the onset of the stimulus and continuing for 15 s afterward, which was long after the slowest response. Activity was measured every 1.5 s. The first two scans provide an estimate of baseline before the

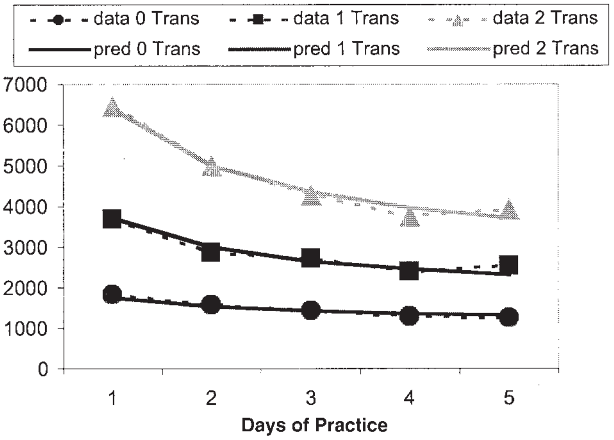


Figure 16. Performance in the symbol manipulation task: Effects of number of transformations and days of practice. Trans = transformations; pred = predictions. From "Predicting the Practice Effects on the Blood Oxygenation Level-Dependent (BOLD) Function of fMRI in a Symbolic Manipulation Task," by Y. Qin et al., 2003, *Proceedings of the National Academy of Sciences, USA, 100,* p. 4952. Copyright 2003 by the National Academy of Sciences, USA. Adapted with permission.

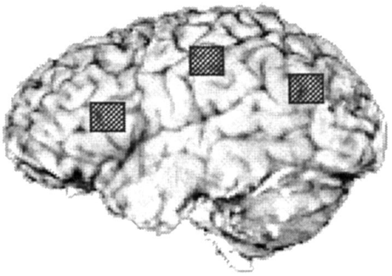


Figure 17. An illustration of the three left regions of interest for modeling. The Talairach coordinates of the left motor area are (−37, −25, 47), the Talairach coordinates of the left posterior parietal lobe are (−23, −64, 34), and the Talairach coordinates of the left prefrontal region are (−40, 21, 21). From "Predicting the Practice Effects on the Blood Oxygenation Level-Dependent (BOLD) Function of fMRI in a Symbolic Manipulation Task," by Y. Qin et al., 2003, *Proceedings of the National Academy of Sciences, USA, 100,* p. 4954. Copyright 2003 by the National Academy of Sciences, USA. Adapted with permission.

stimulus comes on. These figures also display the ACT–R predictions that we explain below. The BOLD functions displayed are typical in that there is some inertia in the rise of the signal after the critical event and then decay. The BOLD response is delayed so that it reaches a maximum about 5 s after the brain activity.

The top portion of Figure 18 shows the activity around the central sulcus in the region that controls the right hand. The effect of complexity is to delay the BOLD function (because the first finger press is delayed in the more complex condition), but there is no effect on the basic shape of the BOLD response because the same response sequence is being generated in all cases. The effect of practice is also just to move the BOLD response forward in this motor region. The middle portion of Figure 18 shows activity around the intraparietal sulcus. It shows an effect of complexity and is not much affected by practice. It shows a considerable rise even in the simplest no-operation condition. This is because it is still necessary to encode the equation in this condition. The amount of information to be encoded or transformed also does not change with practice, and so one would expect little change. The functions do rise a little sooner on Day 5 reflecting the more rapid processing. The bottom part of Figure 18 shows the activity around the inferior frontal sulcus, which we take as reflecting the activity of the retrieval buffer. Although it also shows a strong effect of number of transformations, it contrasts with the parietal region in two ways. First, it shows no rise in the zero-transformation condition because there are no retrievals in this condition. Second, the magnitude of the response decreases after 5 days, reflecting the fact that the declarative structures have been greatly strengthened and the retrievals are much quicker.

The ACT–R Model

Qin et al. (2003) described an ACT–R model that solves these problems. Figure 19 shows the activity of the ACT–R buffers solving an equation that involves a single transformation. It includes an imaginal buffer that holds the mental image of the string of symbols as they are encoded and transformed. Researchers (e.g., Anderson et al., 2003; Sohn, Goode, Stenger, Carter, & Anderson, 2003) have been able to model a number of data sets assuming a 200-ms time for each of the imaginal transformations, and this is the value assumed in Figure 19. The encoding begins with the identification of the ↔ sign and then the encoding of the symbols to the right of the sign. Then begins the process of encoding the elements to the left of the sign and their elimination to isolate the P. In the example in Figure 19, six operations (Steps 1–6) are required to encode the string, and an additional two operations (Steps 9 and 10) are required to encode the transformation. Each of these requires activity in the imaginal module. There are 5 such operations in the case of zero transformations and 10 in the case of two transformations. With respect to retrievals in Figure 19, two pieces of information have to be retrieved for each transformation (Steps 7 and 8) that must be performed. One piece was the operation to perform ("flip" in Figure 19), and the other was the identity of the terms to apply this operation to (argument position in Figure 19). There were 5 retrieval operations in the case of two transformations and none in the case of zero transformations. In all cases, there are the final 5 motor operations (Steps 11–15 in Figure 19), but their timing will vary with how long the overall process takes. The timing of these motor operations is determined by the EPIC-inherited parameters. Qin et al. (2003) estimated that each retrieval operation took 650 ms on Day 1 and because of base-level learning had sped up to 334 ms on Day 5. Base-level learning is the sole factor producing the speed up in Figure 16.

The imaginal buffer in Figure 19 is serving a goal function of maintaining and transforming internal states. We have found this module to be active in all tasks in which participants must imagine changes in a spatial problem representation. Earlier we showed its involvement in the Tower of Hanoi task. This part of the goal representation seems to be maintained in the parietal cortex. Unlike the Tower of Hanoi, equation solving of this kind does not involve means–ends reasoning but progresses by simple transformation of the symbol string to be ever closer to the target state. We think it is for this reason that we have not observed the dorsolateral activation in equation solving or other symbolic transformation tasks.

The behavior in Figure 18 of the cortical regions is qualitatively in accord with the behavior of the buffers of the ACT–R model in Figure 19. We now describe how the quantitative predictions in Figure 18 were obtained. These prediction methods can be used by other information-processing models. A number of researchers (e.g., Boyton, Engel, Glover, & Heeger, 1996; Cohen, 1997; Dale & Buckner, 1997) have proposed that the BOLD response (*B*) to an event varies according to the following function of time, *t*, since the event:

$$B(t) = t^a e^{-t},$$

where estimates of the exponent, *a*, have varied between 2 and 10. This is a gamma function that will reach maximum at $t = a$ time units after the event. Anderson et al. (2003) proposed that while a buffer is active it is constantly producing a change that will result in a BOLD response according the above function. The observed fMRI response is integrated over the time that the buffer is active. Therefore, the observed BOLD response will vary with time as

$$CB(t) = M\int_0^t i(x)B\left(\frac{t-x}{s}\right)dx,$$

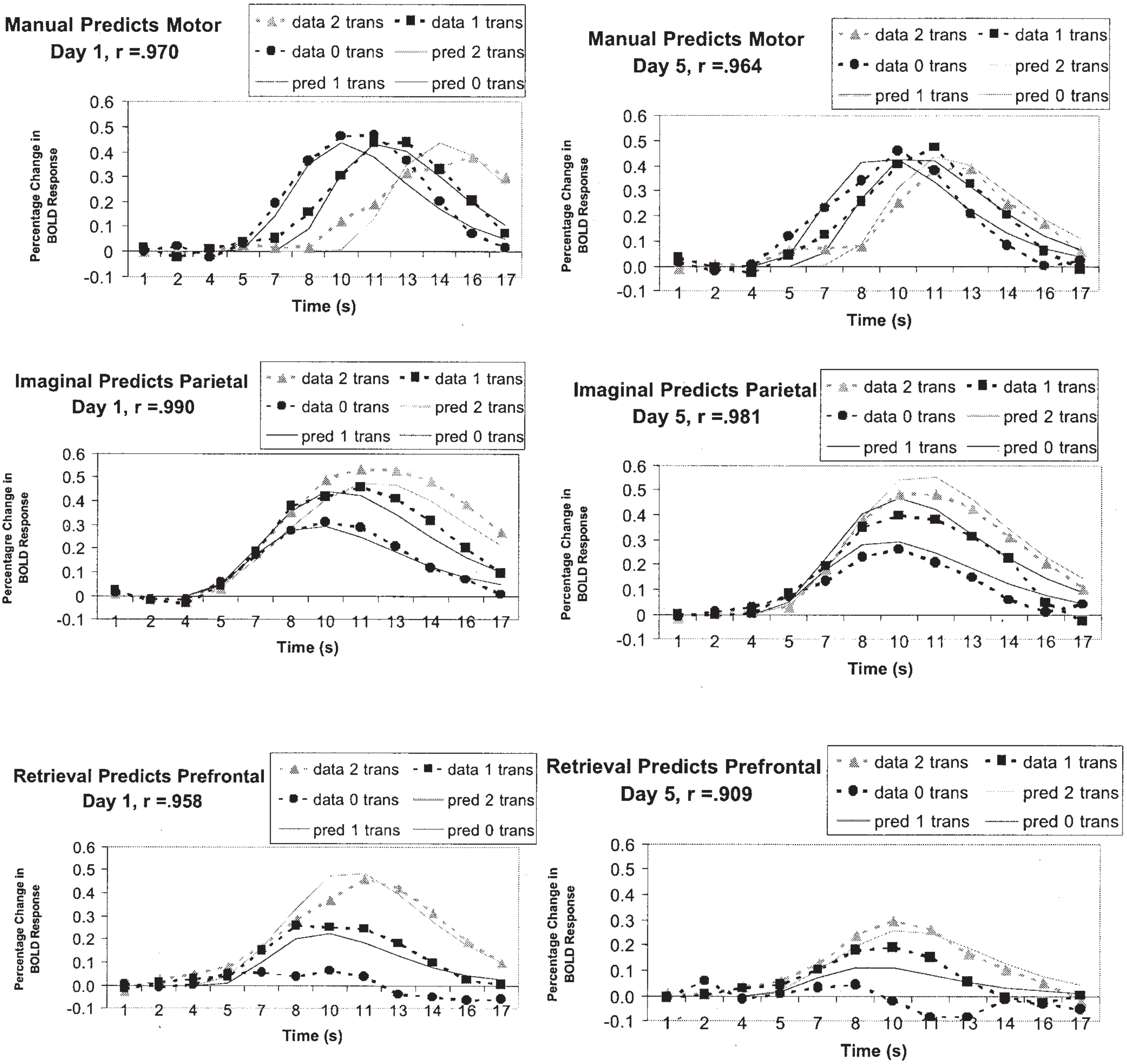


Figure 18. Top row: Ability of manual buffer to predict the activity in the motor particle on Day 1 (left) and Day 5 (right). Middle row: Ability of imaginal buffer to predict the activity in the parietal particle on Day 1 (left) and Day 5 (right). Bottom row: Ability of the retrieval buffer to predict the activity in the prefrontal particle on Day 1 (left) and Day 5 (right). trans = transformations; pred = predictions; BOLD = blood oxygen level dependent. From "Predicting the Practice Effects on the Blood Oxygenation Level-Dependent (BOLD) Function of fMRI in a Symbolic Manipulation Task," by Y. Qin et al., 2003, *Proceedings of the National Academy of Sciences, USA, 100,* p. 4954. Copyright 2003 by the National Academy of Sciences, USA. Adapted with permission.

where *M* is the magnitude scale for response, *s* is the latency scale, and $i(x)$ is 1 if the buffer is occupied at time *x* and 0 otherwise.

This provides a basis for taking patterns of buffer activity such as that in Figure 19 and making predictions for the BOLD functions as in Figure 18. This requires estimating parameters *a, s,* and *M* that reflect the characteristics of the BOLD function in the particular region. Table 1 reproduces those parameter estimates from Qin et al. (2003). Although predicting the exact shape of the BOLD function depends on these parameters, this approach makes some significant parameter-free predictions. These are the relative points at which the functions will peak in different conditions and the relative areas under the curves for different conditions. The differences in time of the peak reflect differences in the onset of the activity, and predictions about peak times are confirmed for the motor region at the top of Figure 18, where the response is being delayed as a function of condition. The differences in area reflect

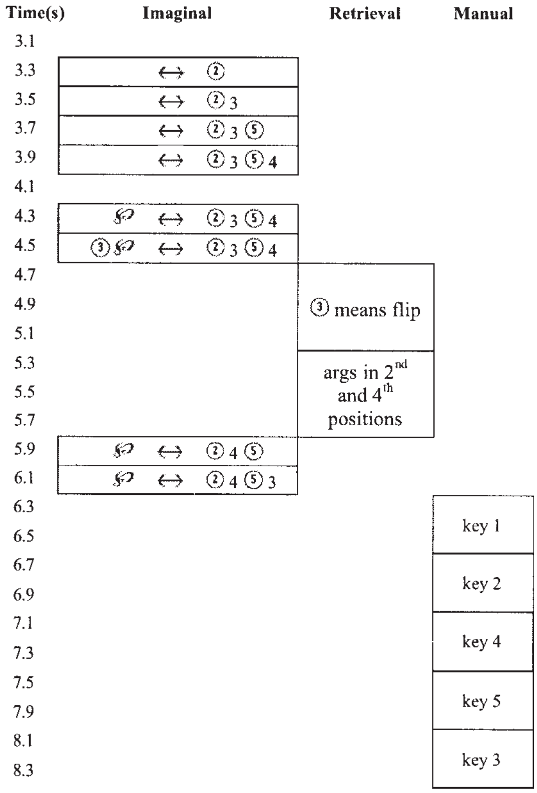


Figure 19. Activity of ACT–R buffers in solving an equation. From "Predicting the Practice Effects on the Blood Oxygenation Level-Dependent (BOLD) Function of fMRI in a Symbolic Manipulation Task," by Y. Qin et al., 2003, *Proceedings of the National Academy of Sciences, USA, 100,* p. 4955. Copyright 2003 by the National Academy of Sciences, USA. Adapted with permission.

the differences in total time the buffer is activated, and predictions about relative area are confirmed in the middle (parietal) and bottom (prefrontal) parts of Figure 18. Thus, once one has committed to an information-processing model like Figure 19 (perhaps by fitting the behavioral data like those in Figure 16) and committed to the corresponding brain regions, one has committed to strong a priori predictions about the resulting BOLD functions. This potential is of great promise in providing guidance for theory development.

The Basal Ganglia

Figure 1 presented the proposal that the basal ganglia were critically involved in the implementation of production rules. Figure 20 shows unpublished data from the Qin et al. (2003) study, showing activity in a region of the basal ganglia corresponding to the head of the caudate. This is a bilateral region, which other studies (e.g., Poldrack, Prabakharan, Seger, & Gabrieli, 1999) have indicated may be related to procedural learning. In Figure 20, there is a response on Day 1 and no response on Day 5. On neither day are the conditions clearly discriminated. An analysis of variance confirms the existence of a day effect, $F(1, 7) = 28.33$, $p < .005$, but not a condition effect, $F(2, 14) = .78$. There is a marginally significant interaction between the two factors, $F(2, 14) = 3.47$,

Table 1
Parameters and the Quality of the BOLD Function Prediction

Parameter	Imaginal	Retrieval	Manual
Scale (s)	1.634	1.473	1.353
Exponent (a)	4.3794	4.167	4.602
Magnitude: $M\,\Gamma(a + 1)$[a]	2.297	1.175	1.834
χ^2[b]	86.85	73.18	74.02

Note. BOLD = blood oxygen level dependent. From "Predicting the Practice Effects on the Blood Oxygenation Level-Dependent (BOLD) Function of fMRI in a Symbolic Manipulation Task," by Y. Qin et al., 2003, *Proceedings of the National Academy of Sciences, USA, 100,* p. 4955. Copyright 2003 by the National Academy of Sciences, USA. Adapted with permission.
[a] This is a more meaningful measure because the height of the function is determined by the exponent as well as *M*. [b] In calculating these chi-squares, we divided the summed deviations by the variance of the means calculated from the Condition × Participant interaction. The chi-square measure has 69 degrees of freedom (72 observation − 3 parameters). None of these reflect significant deviations.

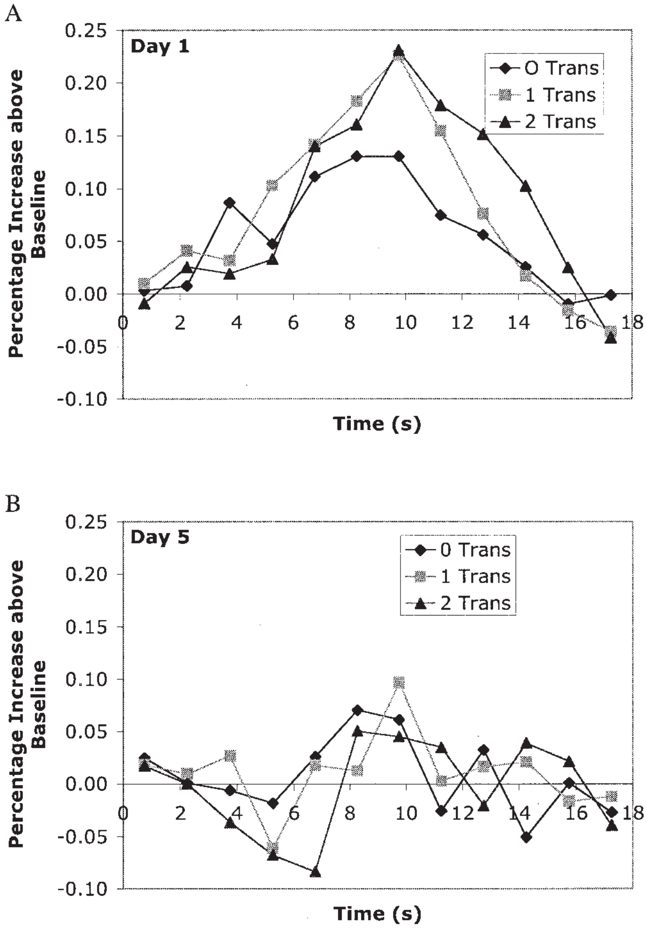


Figure 20. Activity in the basal ganglia for Day 1 (A) and Day 5 (B). This reflects the average of two 4- × 4- × 4-voxel regions centered at (−15, 10, 6) and (15, 10, 6) in the left and right hemispheres, respectively. The area is across the head of the caudate nucleus and putamen (and some globus pallidus). The center of each region of interest is in the internal capsule (white matter, between the head of the caudate nucleus and putamen). Trans = transformations.

$p < .10$, reflecting some tendency for the zero-transformation condition to show a weaker response on Day 1 than the other two conditions.

On the ACT–R analysis, if basal ganglia represent production firing, it is not clear why there should be any rise from baseline in any condition. In all conditions, production rules are firing at more or less a constant rate all the time. Presumably, the participant is thinking about something in the intervals between the trials and so activating the basal ganglia. Trial activity should just maintain that the basal ganglia activity should be at same level as the pretrial level. Apparently, however, as others have found, the basal ganglia are especially active when procedural learning is happening. Thus, it shows a rise from baseline on Day 1 when the procedure is novel but not on Day 5 when it is practiced. Although this is a neural marker possibly related to things in ACT–R such as production rule compilation, we do not yet have a theory of why there is greater activity at these points. The theory that allowed us to predict the BOLD functions in Figure 18 was that the magnitude of the BOLD function reflected the time a module was engaged. Perhaps to account for the basal ganglia response, we need to extend the theory to include some notion of magnitude of engagement. Conversely, the greater activity of the basal ganglia on Day 1 might reflect engagement of production learning processes that are only operating early in learning.

The following summarizes the results found in this study of skill learning:

1. The motor area tracks onset of keying. Otherwise, the form of the BOLD function is not sensitive to cognitive complexity or practice.

2. The parietal area tracks transformations in the imagined equation. The form of the BOLD function is sensitive to cognitive complexity but not practice.

3. The prefrontal area tracks retrieval of algebraic facts. The form of the BOLD function is sensitive to cognitive complexity and decreases with practice.

4. The caudate tracks learning of new procedural skill. The BOLD function is not sensitive to cognitive complexity and disappears with practice.

The ability to predict the first three results shows the promise of an integrated architecture in terms of making sense of complex brain imaging data. The fourth result shows the potential of brain imaging data to stimulate development of a theory of the cognitive architecture in that it challenges us to relate this marker of procedural learning to production rule learning.

General Discussion

Although our concern has naturally been with the ACT–R architecture, we are advancing it as an illustration of the potential of integrated architectures rather than as the final answer. With respect to applications of architectures to understanding the details of human performance in tasks like the AAWC task, there are also impressive contributions of both the EPIC architecture (Kieras & Meyer, 1997) and the Soar architecture (Jones et al., 1999) to similar tasks. Although these architectures have typically been applied to their own tasks, one report (Gluck & Pew, 2001, in press) does compare a number of architectures applied to the same task. With respect to relating to brain imaging data, the 4CAPS (cortical capacity-constrained collaborative activation-based production system) architecture has also been used for this purpose (Just, Carpenter, & Varma, 1999). It is perhaps significant that these are all production system architectures reflecting Newell's (1973) original intention for production systems that they serve an integration function. It is interesting to note in this regard that such integration is not high on the agenda of at least some connectionistic architectures (McClelland, Plaut, Gotts, & Maia, 2003).

Just because such architectures aspire to account for the integration of cognition, they should not be interpreted as claiming to have the complete truth about the nature of the mind. ACT–R is very much a work in progress, and its final fate may be just to point the way to a better conception of mind. No theory in the foreseeable future can hope to account for all of cognition. What distinguishes theories like ACT–R is their focus on the integration of cognition. This concern creates certain demands that are often ignored by other theoretical approaches. As in the AAWC simulation, one cannot model complex tasks without some strong constraints on the possible parameters of the model. It is not feasible to do the kind of parameter estimation that is common in other theoretical approaches. Moreover, the goal of such projects is to predict data not postdict data. Model fitting has been criticized (Roberts & Pashler, 2000) because of the belief that the parameter estimation process would allow any pattern of data to be fit. Although this is not true, such criticisms would be best dealt with by simply eliminating parameter estimation.

Another consequence of the integrated approach is to make us more open to additional data such as fMRI. Basically, we need all the help we can get in terms of guiding the design of the system. A theory that aspires to account for the integration of cognition cannot declare such data beyond its relevance. However, as in the case of the data from the basal ganglia, it is not always immediately obvious how to relate such data in detail to the architecture.

The ACT–R architecture in Figure 1 is incomplete. Of course, it is missing certain modules. More fundamentally, the proposal that all neural processing passes through the basal ganglia is simply false. There are direct connections between cortical brain areas. This would seem to imply that all information processing does not occur through the mediation of production rules as proposed by ACT–R. We already discussed, with respect to the issue of perfect time sharing, that such direct stimulus-to-response connections would provide a way for processing to avoid a serial bottleneck.

Although the analysis in ACT–R is certainly neither complete nor totally correct, we close with review of the answer it gives to how cognition is integrated. As the emerging consensus in cognitive science, it begins with the observation that the mind consists of many independent modules doing their own things in parallel. However, some of these modules serve important place-keeping functions—the perceptual modules keep our place in the world, the goal module our place in a problem, and the declarative module our place in our own life history. Information about where we are in these various spaces is made available in the buffers of the modules. A central production system can detect patterns in these buffers and take coordinated action. The subsymbolic learning and performance mechanisms in ACT–R work to make these actions appropriate. In particular, the subsymbolic declarative mechanisms work to bring the right memories to mind, and the subsymbolic procedural mechanisms work to bring the right rules to bear. The successful models reviewed in this article suggest that this characterization of the integration of cognition contains some fundamental truths.

References

Ackerman, P. L. (1990). A correlational analysis of skill specificity: Learning, abilities, and individual differences *Journal of Experimental Psychology: Learning, Memory, and Cognition, 16,* 883–901.

Amos, A. (2000). A computational model of information processing in the frontal cortex and basal ganglia. *Journal of Cognitive Neuroscience, 12,* 505–519.

Anderson, J. R. (1974). Retrieval of propositional information from long-term memory. *Cognitive Psychology, 5,* 451–474.

Anderson, J. R. (1983). *The architecture of cognition*. Cambridge, MA: Harvard University Press.

Anderson, J. R. (2002). Spanning seven orders of magnitude: A challenge for cognitive modeling. *Cognitive Science, 26,* 85–112.

Anderson, J. R., & Betz, J. (2001). A hybrid model of categorization. *Psychonomic Bulletin & Review, 8,* 629–647.

Anderson, J. R., Bothell, D., Lebiere, C., & Matessa, M. (1998). An integrated theory of list memory. *Journal of Memory and Language, 38,* 341–380.

Anderson, J. R., Budiu, R., & Reder, L. M. (2001). A theory of sentence memory as part of a general theory of memory. *Journal of Memory and Language, 45,* 337–367.

Anderson, J. R., & Douglass, S. (2001). Tower of Hanoi: Evidence for the cost of goal retrieval. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 27,* 1331–1346.

Anderson, J. R., Fincham, J. M., & Douglass, S. (1999). Practice and retention: A unifying analysis. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 25,* 1120–1136.

Anderson, J. R., & Lebiere, C. (1998). *The atomic components of thought.* Mahwah, NJ: Erlbaum.

Anderson, J. R., Qin, Y., Sohn, M.-H., Stenger, V. A., & Carter, C. S. (2003). An information-processing model of the BOLD response in symbol manipulation tasks. *Psychonomic Bulletin & Review, 10,* 241–261.

Anderson, J. R., & Reder, L. M. (1999). The fan effect: New results and new theories. *Journal of Experimental Psychology: General, 128,* 186–197.

Anderson, J. R., & Schooler, L. J. (1991). Reflections of the environment in memory. *Psychological Science, 2,* 396–408.

Ashby, F. G., & Waldron, E. M. (2000). The neuropsychological bases of category learning. *Current Directions in Psychological Science, 9,* 10–14.

Baddeley, A. D. (1986). *Working memory.* Oxford, England: Oxford University Press.

Berger, J. O. (1985). *Statistical decision theory and Bayesian analyses.* New York: Springer-Verlag.

Blessing, S. B., & Anderson, J. R. (1996). How people learn to skip steps. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 22,* 576–598.

Boyton, G. M., Engel, S. A., Glover, G. H., & Heeger, D. J. (1996). Linear systems analysis of functional magnetic resonance imaging in human V1. *Journal of Neuroscience, 16,* 4207–4221.

Braver, T. S., Barch, D. M., Kelley, W. M., Buckner, R. L., Cohen, N. J., Miezin, F. M., et al. (2001). Direct comparison of prefrontal cortex regions engaged by working and long-term memory tasks. *NeuroImage, 14,* 48–59.

Buckner, R. L., Kelley, W. M., & Petersen, S. E. (1999). Frontal cortex contributes to human memory formation. *Nature Neuroscience, 2,* 311–314.

Byrne, M. D. (2003). Cognitive architecture. In J. Jacko & A. Sears (Eds.), *The human-computer interaction handbook. Fundamentals, evolving technologies and emerging applications* (pp. 97–117). Mahwah, NJ: Erlbaum.

Byrne, M. D., & Anderson, J. R. (2001). Serial modules in parallel: The psychological refractory period and perfect time-sharing. *Psychological Review, 108,* 847–869.

Cabeza, R., Dolcos, F., Graham, R., & Nyberg, L. (2002). Similarities and differences in the neural correlates of episodic memory retrieval and working memory. *NeuroImage, 16,* 317–330.

Card, S., Moran, T., & Newell, A. (1983). *The psychology of human-computer interaction.* Hillsdale, NJ: Erlbaum.

Cohen, M. S. (1997). Parametric analysis of fMRI data using linear systems methods. *NeuroImage, 6,* 93–103.

Cosmides, L., & Tooby, J. (2000). The cognitive neuroscience of social reasoning. In M. S. Gazzaniga (Ed.), *The new cognitive neurosciences* (2nd ed., pp. 1259–1272). Cambridge, MA: MIT Press.

Dale, A. M., & Buckner, R. L. (1997). Selective averaging of rapidly presented individual trials using fMRI. *Human Brain Mapping, 5,* 329–340.

Dehaene, S., Spelke, E., Pinel, P., Stanescu, R., & Tsivkin, S. (1999, May 7). Sources of mathematical thinking: Behavior and brain-imaging evidence. *Science, 284,* 970–974.

Douglass, S. A. (1998). EPAL: Data collection and analysis software for eye-tracking experiments [Computer software]. Pittsburgh, PA: Carnegie Mellon University.

Fincham, J. M., Carter, C. S., van Veen, V., Stenger, V. A., & Anderson, J. R. (2002). Neural mechanisms of planning: A computational analysis using event-related fMRI. *Proceedings of the National Academy of Sciences, USA, 99,* 3346–3351.

Fletcher, P. C., & Henson, R. N. A. (2001). Frontal lobes and human memory: Insights from functional neuroimaging. *Brain, 124,* 849–881.

Fodor, J. A. (1983). *The modularity of the mind.* Cambridge, MA: MIT Press/Bradford Books.

Frank, M. J., Loughry, B., & O'Reilly, R. C. (2000). *Interactions between frontal cortex and basal ganglia in working memory: A computational model* (Tech. Rep. No. 00-01). Boulder: University of Colorado, Institute of Cognitive Science.

Freed, M. (Ed.). (2000). *Simulating human agents: Papers from the 2000 AAAI Fall Symposium*. Menlo Park, CA: AAAI Press.

Friedman, M. P., Burke, C. J., Cole, M., Keller, L., Millward, R. B., & Estes, W. K. (1964). Two-choice behavior under extended training with shifting probabilities of reinforcement. In R. C. Atkinson (Ed.), *Studies in mathematical psychology* (pp. 250–316). Stanford, CA: Stanford University Press.

Gluck, K. A., & Pew, R. W. (2001). Overview of the agent-based modeling and behavior representation (AMBR) model comparison project. In *Proceedings of the Tenth Conference on Computer Generated Forces* (pp. 3–6). Orlando, FL: SISO.

Gluck, K. A., & Pew, R. W. (in press). *Modeling human behavior with integrated cognitive architectures: Comparison, evaluation, and validation*. Mahwah, NJ: Erlbaum.

Graybiel, A. M., & Kimura, M. (1995). Adaptive neural networks in the basal ganglia. In J. C. Houk, J. L. Davis, & D. G. Beiser (Eds.), *Models of information processing in the basal ganglia* (pp. 103–116). Cambridge, MA: MIT Press.

Hazeltine, E., Teague, D., & Ivry, R. B. (2002). Simultaneous dual-task performance reveals parallel response selection after practice. *Journal of Experimental Psychology: Human Perception and Performance, 28,* 527–545.

Hikosaka, O., Nakahara, H., Rand, M. K., Sakai, K., Lu, Z., Nakamura, K., et al. (1999). Parallel neural networks for learning sequential procedures. *Trends in Neuroscience, 22,* 464–471.

Hodge, K. A., Rothrock, L., Kirlik, A. C., Walker, N., Fisk, A. D., Phipps, D. A., & Gay, P. E. (1995). *Trainings for tactical decision making under stress: Towards automatization of component skills* (Tech. Rep. No.

HAPL-9501). Atlanta, GA: Georgia Institute of Technology, School of Psychology, Human Attention and Performance Laboratory.

Houk, J. C., & Wise, S. P. (1995). Distributed modular architectures linking basal ganglia, cerebellum, and cerebral cortex: Their role in planning and controlling action. *Cerebral Cortex, 2,* 95–110.

Jones, R. M., Laird, J. E., Nielsen, P. E., Coulter, K. J., Kenny, P., & Koss, F. V. (1999). Automated intelligent pilots for combat flight simulation. *AI Magazine, 20*(1), 27–41.

Just, M. A., & Carpenter, P. N. (1992). A capacity theory of comprehension: Individual differences in working memory. *Psychological Review, 99,* 122–149.

Just, M. A., Carpenter, P. A., & Varma, S. (1999). Computational modeling of high-level cognition and brain function. *Human Brain Mapping, 8,* 128–136

Kieras, D., & Meyer, D. E. (1997). An overview of the EPIC architecture for cognition and performance with application to human-computer interaction. *Human-Computer Interaction, 12,* 391–438.

Kieras, D. E., Meyer, D. E., Mueller, S., & Seymour, T. (1999). Insights into working memory from the perspective of the EPIC architecture for modeling skilled perceptual-motor performance. In P. Shah & A. Miyake (Eds.), *Models of working memory: Mechanisms of active maintenance and executive control* (pp. 183–223). Cambridge, England: Cambridge University Press.

Koechlin, E., Corrado, G., Pietrini, P., & Grafman, J. (2000). Dissociating the role of the medial and lateral anterior prefrontal cortex in human planning. *Proceedings of the National Academy of Sciences, USA, 97,* 7651–7656.

Lebiere, C., & Wallach, D. (2001). Sequence learning in the ACT–R cognitive architecture: Empirical analysis of a hybrid model. In R. Sun & C. L. Gilles (Eds.), *Sequence learning: Paradigms, algorithms, and applications* (pp. 188–212). Berlin, Germany: Springer.

Lovett, M. C. (1998). Choice. In J. R. Anderson & C. Lebiere (Eds.), *The atomic components of thought* (pp. 255–296). Mahwah, NJ: Erlbaum.

Lovett, M. C., Daily, L. Z., & Reder, L. M. (2000). A source activation theory of working memory: Cross-task prediction of performance in ACT-R. *Cognitive Systems Research, 1,* 99–118.

McClelland, J. L., Plaut, D. C., Gotts, S. J., & Maia, T. V. (2003). Developing a domain-general framework for cognition: What is the best approach? *Behavioral and Brain Sciences, 26,* 611–614.

Meyer, D. E., & Kieras, D. E. (1997). A computational theory of executive cognitive processes and multiple-task performance. Part 1. Basic mechanisms *Psychological Review, 104,* 2–65.

Middleton, F. A., & Strick, P. L. (2000). Basal ganglia and cerebellar loops: Motor and cognitive circuits. *Brain Research Reviews, 31,* 236–250.

Newell, A. (1973). Production systems: Models of control structures. In W. G. Chase (Ed.), *Visual information processing* (pp. 463–526). New York: Academic Press.

Newell, A. (1990). *Unified theories of cognition*. Cambridge, MA: Harvard University Press.

Newman, S. D., Carpenter, P. A., Varma, S., & Just, M. A. (in press). Frontal and parietal participation in problem-solving in the Tower of London: fMRI and computational modeling of planning and high-level perception. *Neuropsychologia.*

Nyberg, L., Cabeza, R., & Tulving, E. (1996). PET studies of encoding and retrieval: The HERA model. *Psychonomic Bulletin & Review, 3,* 135–148.

Pashler, H. E. (1998). *The psychology of attention.* Cambridge, MA: MIT Press.

Petrides, M. (1994). Frontal lobes and working memory: Evidence from investigations of the effects of cortical excisions in nonhuman primates. In F. Boller & J. Grafman (Eds.), *Handbook of neuropsychology* (Vol. 9, pp. 59–82). Amsterdam: Elsevier.

Pew, R. W., & Mavor, A. S. (1998). *Modeling human and organizational behavior: Application to military simulations.* Washington, DC: National Academy Press.

Pirolli, P. L., & Anderson, J. R. (1985). The role of practice in fact retrieval. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 11,* 136–153.

Poldrack, R. A., Prabakharan, V., Seger, C., & Gabrieli, J. D. E. (1999). Striatal activation during cognitive skill learning. *Neuropsychology, 13,* 564–574.

Qin, Y., Anderson, J. R., Silk, E., Stenger, V. A., & Carter, C. S. (2004). The change of the brain activation patterns along with the children's practice in algebra equation solving. *Proceedings of the National Academy of Sciences, USA, 101,* 5686–5691.

Qin, Y., Sohn, M.-H., Anderson, J. R., Stenger, V. A., Fissell, K., Goode, A., & Carter, C. S. (2003). Predicting the practice effects on the blood oxygenation level-dependent (BOLD) function of fMRI in a symbolic manipulation task. *Proceedings of the National Academy of Sciences, USA, 100,* 4951–4956.

Reder, L. M., & Gordon, J. S. (1997). Subliminal perception: Nothing special, cognitively speaking. In J. Cohen & J. Schooler (Eds.), *Cognitive and neuropsychological approaches to the study of consciousness* (pp. 125–134). Mahwah, NJ: Erlbaum.

Reichle, E. D., Pollatsek, A., Fisher, D. L., & Rayner, K. (1998). Toward a model of eye movement control in reading. *Psychological Review, 105,* 125–157.

Reichle, E. D., Rayner, K., & Pollatsek, A. (1999). Eye movement control in reading: Accounting for initial fixation locations and refixations within the E-Z Reader model. *Vision Research, 39,* 4403–4411.

Rescorla, R. A., & Wagner, A. R. (1972). A theory of Pavlovian conditioning: Variations on the effectiveness of reinforcement and nonreinforcement. In A. H. Black & W. F. Prokasy (Eds.), *Classical conditioning: II. Current research and theory* (pp. 64–99). New York: Appleton-Century-Crofts.

Roberts, S., & Pashler, H. (2000). How persuasive is a good fit? A comment on theory testing. *Psychological Review, 107,* 358–367.

Ruthruff, E., Pashler, H., & Hazeltine, E. (2003). Dual-task interference with equal task emphasis: Graded capacity sharing or central postponement? *Perception & Psychophysics, 65,* 801–816.

Saint-Cyr, J. A., Taylor, A. E., & Lang, A. E. (1988). Procedural learning and neostriatal dysfunction in man. *Brain, 111,* 941–959.

Salvucci, D. D. (2001). An integrated model of eye movements and visual encoding. *Cognitive Systems Research, 1,* 201–220.

Schumacher, E. H., Seymour, T. L., Glass, J. M., Fencsik, D. E., Lauber, E. J., Kieras, D. E., & Meyer, D. E. (2001). Virtually perfect time sharing in dual-task performance: Uncorking the central cognitive bottleneck. *Psychological Science, 12,* 101–108.

Schumacher, E. H., Seymour, T. L., Glass, J. M., Lauber, E. J., Kieras, D. E., & Meyer, D. E. (1997, November). *Virtually perfect time sharing in dual-task performance.* Paper presented at the 38th annual meeting of the Psychonomic Society, Philadelphia, PA.

Simon, H. A. (1975). The functional equivalence of problem solving skills. *Cognitive Psychology, 7,* 268–288.

Smith, E. E., & Jonides, J. (1999, March 12). Storage and executive processes in the frontal lobes. *Science, 283,* 1657–1661.

Sohn, M.-H., Douglass, S. A., Chen, M.-C., & Anderson, J. R. (in press). Characteristics of fluent skills in a complex, dynamic problem-solving task. *Human Factors.*

Sohn, M.-H., Goode, A., Koedigner, K. R., Stenger, V. A, Fissell, K., Carter, C. S., & Anderson, J. R. (2004). *Behavioral equivalence, but not neural equivalence: Neural evidence of alternative strategies in mathematical problem solving.* Manuscript submitted for publication.

Sohn, M.-H., Goode, A., Stenger, V. A, Carter, C. S., & Anderson, J. R. (2003). Competition and representation during memory retrieval: Roles of the prefrontal cortex and the posterior parietal cortex. *Proceedings of National Academy of Sciences, USA, 100,* 7412–7417.

Squire, L. R. (1987). *Memory and brain.* New York: Oxford University Press.

Taatgen, N. A., & Anderson, J. R. (2002). Why do children learn to say "broke"? A model of learning the past tense without feedback. *Cognition, 86,* 123–155.

Thompson-Schill, S. L., D'Esposito, M., Aguirre, G. K., & Farah, M. J. (1997). Role of left inferior prefrontal cortex in retrieval of semantic knowledge: A re-evaluation. *Proceedings of the National Academy of Science, USA, 94,* 14792–14797.

Treisman, A. M., & Gelade, G. (1980). A feature-integration theory of attention. *Cognitive Psychology, 12,* 97–136.

Ungerleider, L. G., & Miskin, M. (1982). Two cortical visual systems. In D. J. Engle, M. A. Goodale, & R. J. Mansfield (Eds.), *Analysis of visual behavior* (pp. 549–586). Cambridge, MA: MIT Press.

Wagner, A. D., Paré-Blagoev, E. J., Clark, J., & Poldrack, R. A. (2001). Recovering meaning: Left prefrontal cortex guides controlled semantic retrieval. *Neuron, 31,* 329–338.

Warrington, E. K., & Shallice, T. (1984). Category specific semantic impairments. *Brain, 197,* 829–854.

Wise, S. P., Murray, E. A., & Gerfen, C. R. (1996). The frontal cortex-basal ganglia system in primates. *Critical Reviews in Neurobiology, 10,* 317–356.

Wolfe, J. M. (1994). Guided search 2.0: A revised model of visual search. *Psychonomic Bulletin & Review, 1,* 202–238.

Appendix

Rules for the GT-ASP Experiment

1. The task is to identify unidentified tracks. Unidentified tracks are half squares with vectors emanating from them. One should hook (click on) such tracks and then go through the sequence of identifying them. (To identify-tracks first look-for a track that is "half-square" then hook the track then idsequence the track and then repeat)
2. One way to identify a track is to confirm that it is flying at a commercial altitude and speed and then record it as friendly primary ID and nonmilitary air ID. (To idsequence a track first altitude-test then speed-test and then record it as "friend" "non-military")
3. The other way to identify a track is to request its EWS identity and then classify the track according to that identity. (To idsequence a track first ews the track for a ews-signal and then classify the track according to the ews-signal)
4. To confirm that a plane is flying at the commercial altitude, look in the upper left, search down for "alt," read the value to the right, and confirm that it is more than 25,000 and less than 35,000. (To altitude-test first seek "upper-left" and then search-down for "alt" at a location then read-next from the location a value then check-less 25000 than the value and then check-less the value than 35000)
5. To confirm that a plane is flying at the commercial speed, look in the upper left, search down for "speed," read the value to the right, and confirm that it is more than 350 and less than 550. (To speed-test first seek "upper-left" and then search-down for "speed" at a location then read-next from the location a value then check-less 350 than the value and then check-less the value than 550)
6. To request the EWS identity of a track, select the *ews* key, then select *query sensor status* key, and encode the value that you are told. (To ews a track for a ews-signal first select "ews" then select "query sensor status" and then encoe-ews the ews-signal)
7. To classify a track whose EWS identity is ARINC, record it as "friendly" primary ID and "nonmilitary" air ID. (To classify a track according to a ews-signal first match the ews-signal to "arinc564" and then record it as "friend" "non-military")
8. To classify a track whose EWS identity is APQ, record it as hostile primary ID and strike air ID. (To classify a track according to a ews-signal first match the ews-signal to "apq" and then record it as "hostile" "strike")
9. To classify a track whose EWS identity is APG, record it as friendly primary ID and strike air ID. (To classify a track according to a ews-signal first match the ews-signal to "apg" and then record it as "friend" "strike")
10. To classify a track whose EWS identity is negative, treat it as unclassifiable. (To classify a track according to a ews-signal first match the ews-signal to "negative" and then mark-node the track)
11. To record a primary ID and a secondary ID, select the following sequence of keys: *track manager, update hooked track, class/amp, primary-id, the primary id, air-id, the air-id,* and *save.* Then, you have succeeded. (To record a primary-id and a air-id first select "track manager" then select "update hooked track" then select "class/amp" then select "primary id" then select the primary-id then select "air id amp" then select the air-id then select "save changes" and then success)
12. To select a key, find where it is in the menu and hit the corresponding function key. (To select a option first find-menu the option at a location and then hit-key corresponding to the location)
13. To find where an item is in the menu, look to the lower left and search to the right for the term. (To find-menu a option at a location first seek "lower-left" and then search-right for the option at a location)

Note. The exact instructions that were given to the ACT–R model are shown in parentheses. GT-ASP = Georgia Tech Aegis Simulation Program; EWS = electronic warfare signal.

Received October 7, 2002
Revision received September 29, 2003
Accepted October 6, 2003