Production System Models of Complex Cognition

John R. Anderson
Carnegie Mellon University
Department of Psychology
Pittsburgh, PA 15213
ja@cmu.edu

Bonnie E. John
Carnegie Mellon University
Computer Science & Psychology Depts. & Human-Computer Interaction Institute
Pittsburgh, PA 15213
bej@cs.cmu.edu

Marcel Adam Just and Patricia A. Carpenter
Carnegie Mellon University
Department of Psychology
Pittsburgh, PA 15213
just@psy.cmu.edu
carpenter@psy.cmu.edu

David E. Kieras and David E. Meyer
University of Michigan
Electrical Engineering and Computer Science Dept.
Ann Arbor, MI 48109
kieras@eecs.umich.edu

Abstract

There have been a number of production system models which have recently made substantial advances in modeling higher-level cognition. These type of model offers only comprehensive approaches to the modeling of higher level cognition. This symposium will involve presentations by four exemplars of this approach to cognitive modeling (ACT, CAPS, EPIC, and SOAR). The presentations will try to illustrate the range of applications to which such models are appropriate, what the similarities and differences are among the various architectures, and what some of the interesting research questions are within each architecture.

The ACT-R Theory

ACT-R (Anderson, 1993) is a model of human cognition which assumes that a production system operates on a declarative memory. It is a successor to previous ACT production-system models (Anderson, 1976, 1983) and continues the emphasis on activation-based processes as the mechanisms for relating the production system to the declarative memory. Different traces in declarative memory have different levels of activation which determine their rates and probabilities of being processed by the production rules. ACT-R is distinguished from the prior ACT theories in that the details of its design have been strongly guided by the rational analysis of Anderson (1990). Essentially, ACT-R is a production system tuned to perform optimally given the statistical structure of the environment.

According to the ACT theories, knowledge is divided into declarative knowledge and procedural knowledge. In ACT-R, declarative knowledge is represented in terms of **chunks** which are schema-like structures, consisting of an *isa* slot specifying their category and some number of additional slots encoding their contents. Below is a graphical display of a chunk encoding the addition fact that 3+4=7.

fact 3 + 4

isa	addition-fact
addend1	three
addend2	four
sum	seven

According to ACT, procedural knowledge, such as mathematical problem-solving skill is represented by productions. For instance, suppose a child was at the point illustrated below in the solution of a multi-column addition problem:

$$\begin{array}{r} 5\,3\,1 \\ \underline{+\,2\,4\,8} \\ 9 \end{array}$$

Focused on the tens column, the following production rule might apply from the simulation of multi-column addition (Anderson, 1993):

PROCESS-COLUMN
IF the goal is to write out an answer in column c1
and d1 and d2 are digits in that column
and d3 is the sum of d1 and d2
THEN set a subgoal to write out d3 in c1.

The first clause in this production matches the current goal to process the tens column; the second clause matches the digits in the tens column; and the third clause matches a fact or chunk from long-term memory. According to the ACT-R theory, an important component of the time for this production to apply will be the time to retrieve the long-term memories required to match the production rule. So, in this case where 3 and 4 are in the current column, the time to match the last clause will be determined by the level of activation of the chunk encoding 3 + 4 = 7. The next subsection will explain how activation determines match time.

The presentation will describe an ACT-R model of memory span to illustrate the activation computations, limitations on capacity, and role of activation in partial matching.

The Soar Unified Theory of Cognition

Soar is a symbolic cognitive architecture that implements goal-oriented behavior as search through a problem space. Complete discussions of Soar describe a hierarchy from an abstract knowledge level down to a hardware-, or wetware-dependent technology level (Polk & Rosenbloom, 1994), but the two middle levels are of primary interest when comparing Soar to other cognitive architectures. The problem-space level describes deliberate, goal-oriented behavior; the architecture level implements the problem-space level and is concerned with the mechanisms of memory-retrieval and learning.

At the problem-space level, Soar can be described as a set of interacting problem spaces, where each problem space contains a set of *operators* that are applied to *states* to produce new states. A task, or goal, in a problem-space is modeled by the specification of an initial state and one or more desired states. When sufficient knowledge is available in the problem space for a single operator to be selected and applied to the current state, then behavior of a Soar model is strongly directed and smooth, as is skilled human behavior. When knowledge is missing, either search in additional problem spaces may be necessary to locate the knowledge, or decisions must be made without the knowledge, leaving open the probability of errors, and thus, error-recovery activities. This produces more complex branching and back-tracking performance, as displayed in human problem-solving behavior.

The architecture level is itself a hierarchy. At the lowest level, Soar consists of perceptual and motor modules that provide the means of perceiving and acting upon an external world. At the next level, *associations*, in the form of symbolic productions, match the contents of working-memory (comprised of the inputs of perception, the output parameters for the motor modules, and purely internal structure) to retrieve additional information from long-term memory. In contrast to most classical production systems, Soar's associations match and fire in parallel, are limited in their action repertoire to the generation of preferences for the activation of working-memory structure, and automatically retract these preferences when their conditions no longer match. Associations repeatedly fire and retract until no further associations are eligible to do so, then Soar's decision-level process weighs all the active preferences and chooses a new problem-space, operator, or state. Whenever the activations are not sufficient to allow a unique choice, the architecture responds by setting a subtask to resolve this impasse, and the entire process recurs. If the recursive processing adds new preferences that are active in the original task, Soar's architectural learning mechanism (called *chunking*) creates a new association between those working-memory structures in the original task that led, through a chain of associations in the subtask, to the new preferences in the original task. Thus, chunking effectively transfers knowledge from the subtask space to the original task space. Chunking straightforwardly produces speed-up, but can also inductively acquire new knowledge (Rosenbloom, Laird & Newell, 1991).

Using an estimate of 50 msec per decision cycle, Soar has been used to model human performance in many different real-time tasks, from visual search, to natural-language comprehension, to planning and problem-solving tasks. The presentation will emphasize the contribution of architectural constraint on integrating individually-developed models in the service of high-level tasks that require these component capabilities.

3CAPS Simulation Systems for Modeling a Limited-Capacity Working Memory

3CAPS is a Capacity-Constrained, Concurrent, Activation-based Production System which instantiates a capacity theory of working memory (Just & Carpenter, 1992). The theory proposes that a major constraint in immediate processing in language comprehension, problem solving, and spatial reasoning is the amount of activation available to simultaneously perform cognitive computations and actively maintain intermediate and final products.

3CAPS instantiates the capacity theory in a symbolic processing environment (a production system) that incorporates several activation-based, connectionist mechanisms. First, the representations are graded, in that each representational element has an associated activation level that changes when a production either increments or decrements it, or when there is a global deallocation of activation. An element can enable a production to fire only if its activation level is above some threshold. Second, the processing is graded in that the productions do their work gradually, over several cycles of the production system, by incrementally propagating activation from source elements to output elements, until the target elements reach threshold or some other process intervenes. Third, all satisfied productions can fire in parallel.

The model's assumptions include:

1. There is a limited amount of activation to support both information maintenance and computations. The demand for activation differs among tasks,

producing cognitive performance differences in response times and error rates. Individuals differ in the amount of the activation resources that they possess, accounting for some of the systematic individual differences in cognitive performance.

2. In the event of activation demand exceeding the supply, activation is partially deallocated both from the processing function (producing a slowing of processing because the productions will require more cycles to propagate a given amount of activation), and from the maintenance of previously computed partial products (producing forgetting).

3. Capacity utilization, some measure of the proportion of the resource pool that is being consumed in a given time interval, is determined conjointly by the size of an individual's resource pool and by the demand of a given task.

Capacity utilization can be measured in the models as the proportion of the resource pool that is in play. In experimental studies, capacity utilization may correspond to workload or effort, and may be manifested in physiologically based measures of performance, such as brain activation, pupillary response, and ERPs. Moreover, these capacity utilization effects can be evaluated as they are engendered by task effects and by individual differences.

The 3CAPS system can be used for modeling diverse tasks such as sentence comprehension (Just & Carpenter, 1992), story comprehension (Goldman & Varma, 1995), and human-computer interaction (Huguenard, Lerch, Junker, Patz, & Kass, 1993) and in aphasic comprehension (Miyake, Carpenter, & Just, 1994). The presentation will describe recent 3CAPS models that illustrate the effects of individual differences and capacity constraints.

The EPIC Architecture Computational Models of Human Performance

EPIC (Executive Process-Interactive Control) is a computational framework for constructing models of human information-processing and performance which couples perceptual-motor mechanisms with a production-system representation of procedural skill. EPIC has a production-rule cognitive processor surrounded by perceptual and motor peripheral processors whose properties are based on the current research literature. We are pursuing two lines of work with EPIC: One is detailed analyses of multimodal, high-performance human-computer interaction situations, the other is understanding executive processes in human multiple-task performance.

Our key principles can be summarized as follows:

1. Our computational models are built in terms of a detailed general architecture that covers human perceptual, cognitive, and motor mechanisms, and which is required to be accurate and applicable across task domains.

2. A central role is given to cognitive strategies for task execution, which we represent using production systems.

3. Executive processes for coordinating multiple tasks are treated simply as additional strategies, and likewise are represented with sets of production rules.

4. EPIC does not assume an inherent central-processing bottleneck. We attempt to explain performance limitations in both single- and multiple-task situations in terms of the strategic effects of the task instructions, limited working memory capacity, and perceptual-motor constraints.

Thus unlike the heavy emphasis on purely cognitive processes in many modeling efforts, we have undertaken a detailed and explicit consideration of how perceptual and motor mechanisms interact with cognitive mechanisms to determine human abilities and limitations. Our presentation provides a brief description of the EPIC architecture and a summary of how we have applied EPIC to human-computer interaction problems and complex dual-task performance.

References

Anderson, J. R. (1976). *Language, memory, and thought*. Hillsdale, NJ: Erlbaum.

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

Anderson, J. R. (1993). *Rules of the mind*. Hillsdale, NJ: Erlbaum.

Goldman, S. R., & Varma, S. (1995). CAPing the construction-integration model of discourse comprehension. In C. A. Weaver, S. Mannes, & C. R. Fletcher (Eds.), *Discourse comprehension: Essays in Honor of Walter Kintsch*. (pp. 337-358). Hillsdale, NJ: Erlbaum.

Huguenard, B. R., Lerch, F. J., Junker, B. J., Patz, R., & Kass, R. (1993). Modeling working memory failure in phone-based interaction. Working Paper, Graduate School of Industrial Administration, Carnegie Mellon University, Pittsburgh, PA.

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

Miyake, A., Carpenter, P. A., & Just, M. A. (1994). A capacity approach to syntactic comprehension disorders: Making normal adults perform like aphasic patients. *Cognitive Neuropsychology, 11*, 671-717.

Polk, T. A. & Rosenbloom, P. S. 1994. Task-independent constraints on a unified theory of cognition. In F. Boller & J. Grafman (Eds.), *Handbook of Neuropsychology*, Volume 9. Amsterdam, Netherlands: Elsevier.

Rosenbloom, P. S., Newell, A., & Laird, J. E. (1991). Towards the knowledge level in Soar: The role of the architecture in the use of knowledge. In K. VanLehn (Ed.), *Architectures for Intelligence*. Hillsdale, NJ: Erlbaum.