Simulation of Expert Memory Using EPAM IV

Howard B. Richman, James J. Staszewski, and Herbert A. Simon
Carnegie Mellon University

EPAM is a theory of the processes of human perception and memory, first programmed for a computer by Feigenbaum in 1959, which has shown an excellent fit to experimental data from a wide variety of psychological tasks. Over the years, it has been progressively extended to new domains without essential change in its central mechanisms. This article examines EPAM IV, a version extended to account for expert memory, especially the work in recent years by Chase and Simon, Chase and Ericsson and Staszewski. EPAM IV has also been adapted to deal with numerous other short-term and long-term memory tasks which will be reported elsewhere. The main modifications of EPAM that are relevant to the serial recall task examined in this article are a schema in long-term memory (called a retrieval structure) created by the expert's learning, and the addition of an associative search process in long-term memory. These new components operate in close interaction with the other EPAM structures to match the observed behavior. EPAM IV reproduces all of the phenomena explained previously by EPAM III, and in addition, gives an accurate detailed account of the performance (studied by Staszewski) of an expert recalling long sequences of digits. The theory substantially revises, improves and extends Chase and Simon's earlier "chunking" explanation of expert memory.

In cognitive psychology in recent years much empirical research has focused on the abilities of experts to retain large amounts of information in memory after brief exposure to stimuli — much briefer exposure than is required for rote verbal learning in the standard paired-associate paradigms (Bellezza, Six, & Phillips, 1992; Chase & Ericsson, 1982; Ericsson, Chase & Faloon, 1980; Ericsson & Oliver, 1984; Ericsson & Staszewski, 1989; Payne & Wenger, 1992; Staszewski, 1988a; Thompson, Cowan & Frieman, 1993).

Extrapolation from the results of verbal learning experiments (see Baddeley, 1981) would predict that short-term memory (STM) has insufficient capacity to retain the information presented and that the information could not be stored in long-term memory (LTM) at the rapid rates at which it was presented (Bogelski, 1962; Simon, 1976). The problem is to explain how the information can be acquired and retained within the narrow time limits allowed.

We show how these and similar phenomena can be explained by the Elementary Perceiver and Memorizer, Model IV (EPAM IV) program, a version of EPAM that incorporates in the LTM of the previous model a learned retrieval structure that stores one part of the accumulating experience of the expert subject. Such a mechanism was proposed by Chase and Ericsson (1982) and by Staszewski (1988a), but has not previously been incorporated in a comprehensive model of perception and memory.

A well-known example of a retrieval structure is the so-called Method of Loci, used since Greek and Roman times to facilitate memorization, successive items are associated with successive loci in the building; and recall is achieved by visiting these loci and noting what items are stored there.

Experts in a domain commonly possess retrieval structures that they have learned, deliberately or incidentally, while acquiring expertise. Experimental data reveal considerable detail about the form that such schemas assume in several task domains, including memory for lengthy number sequences, short number sequences in mental calculation, restaurant waiters' memories for customers' orders and chess (Chase & Ericsson, 1981, 1982; Ericsson & Polson, 1988a; Staszewski, 1988a, 1988b). For the chess expert, chessboards have slots with which typical patterns of pieces and other information can be associated (Gobet, 1993).

In recent years additional support for the expert memory theory has been obtained in studies that vary in a number of details, such as the stimuli (numbers, words), and the presentation rates (1 s to 5 s or more; Bellezza et al., 1992; Payne & Wenger, 1992; Thompson et al., 1993). All of these studies report that the mnemonists made use of previously acquired semantic memory, retrieval structures, or both, and report retention of sequences which, while not quite as long as those reported here, exceed by many times the normal short-term digit span of 7 or so.

From the data that we examine here, obtained over 3 years from a subject, DD, it appears that he used retrieval structures conjointly with information already stored in semantic memory. The data of DD's performance, and the earlier performance of another subject SF (Ericsson, et al., 1980) are the first to be modeled in detail. A major goal of the present study is to build and test, within the EPAM model, a viable theory of retrieval structures.

A model of a single subject carrying out a single, rather unusual, task would not be of general interest except for the insight it gives us into the mechanisms that account for the performance, mechanisms that can be generalized to many other subjects and many other tasks. Our goal, then, is to model a
After he was cued by the digit chunk at a particular position in his retrieval structures; (b) his gradual improvement in five aspects of DD's skilled-memory performance: (a) his creation of mnemonic aids. At the same time he learned, and stored in long-term memory. He was able to do this with a high level of accuracy, not in order of presentation, but according to the semantic interpretations he had associated with them in long-term memory. DD had associated with them in long-term memory. He was able to do this with a high level of accuracy, not in order of presentation, but according to the semantic interpretations he had associated with them in long-term memory. Over the course of about 865 practice sessions he acquired the ability to recall, in the order of presentation, up to 104 digits. At the end of each practice session, DD was also asked to recall all of the digits that had been presented during that session. He was able to do this with a high level of accuracy, not in order of presentation, but according to the semantic interpretations he had associated with them in long-term memory. After about 790 practice sessions, when DD's span was about 90 digits, he performed a third memory task, memory scanning: After he was cued by the digit chunk at a particular position in the list, he recalled the just-following or just-preceding chunk. This article reports EPAM IV's simulation of the following five aspects of DD's skilled-memory performance: (a) his creation of his retrieval structures; (b) his gradual improvement in performance on the serial recall task over the course of the practice sessions; (c) the detail of his performance on the serial recall task; (d) his performance on the free-recall task; and (e) his performance on the memory scanning task.

Introduction and Overview

In the first section of the article, we will briefly describe the tasks performed by the subject and sketch out the EPAM theory and its implementation as a computer program. Next, we describe more fully the newest version of EPAM, EPAM IV, and the modifications that permit it to simulate expert performance in recalling sequences of digits presented at one per second. The remaining sections will describe the operation of EPAM IV in this task and compare its behavior with that of a human subject, DD. We are not acquainted with any other theories that undertake to account, in detail, for DD's performance or the performance of other skilled mnemonists, or to show how such skills can be reconciled with "normal" memory capacity limits and rates of acquisition in a general model of memory.

In discussing the changes introduced into EPAM, we need to distinguish between alterations in architecture, which are genuine changes in the theory, and changes through learning (without modification of EPAM's architecture) that represent performance-enhancing additions to the expert's knowledge. "Additions to knowledge" include both new information and new strategies for performing particular tasks, both stored in LTM. These additions to memory alter the initial conditions from which the experiment starts and at each subsequent stage. They represent the subject's learning prior to and during the experiment. What is being tested in this study is whether EPAM, after initial learning has occurred, will continue to learn at the same rate and will achieve the same level of performance as the expert subject did in a number of memory tasks.

Expert Memory Tasks

Over a period of more than 3 years, a subject, DD, working in almost daily sessions, attained a remarkable ability to recall long sequences of digits that were presented at the rate of about one digit per second. In order to do this, he created and learned a sequence of larger and larger retrieval structures, which he used as mnemonic aids. At the same time he learned, and stored in semantic memory, triples and quadruples of digits interpreted as running times or as ages, and used these interpretations to help recall the digits reliably. He also learned to notice, and use as cues, symmetries and other patterns among the digits. Over the course of about 865 practice sessions he acquired the ability to recall, in the order of presentation, up to 104 digits.

At the end of each practice session, DD was also asked to recall all of the digits that had been presented during that session. He was able to do this with a high level of accuracy, not in order of presentation, but according to the semantic interpretations he had associated with them in long-term memory.

After about 790 practice sessions, when DD's span was about 90 digits, he performed a third memory task, memory scanning: After he was cued by the digit chunk at a particular position in the list, he recalled the just-following or just-preceding chunk.

This article reports EPAM IV's simulation of the following five aspects of DD's skilled-memory performance: (a) his creation of his retrieval structures; (b) his gradual improvement in performance on the serial recall task over the course of the practice sessions; (c) the detail of his performance on the serial recall task; (d) his performance on the free-recall task; and (e) his performance on the memory scanning task.

Brief Introduction to EPAM

Initially constructed in 1959 by Edward A. Feigenbaum, EPAM has undergone several revisions. The present version, EPAM IV, grows out of EPAM III, first reported in Simon and Feigenbaum (1964) and described more fully in Feigenbaum and Simon (1984). A variant of EPAM III, called EPAM IIIA (Richman & Simon, 1989), successfully simulated context effects in letter perception in addition to replicating all of the earlier EPAM simulations.

EPAM describes and explains a wide range of human perceptual and memory processes. In addition to the tasks reported in this article and the experiment on context effects just mentioned, EPAM has been used successfully to account for the observed effects, in paired-associate or serial anticipation verbal learning paradigms, of speed of presentation of stimuli, interlist and intralist similarity, familiarization, one-trial versus multi-trial learning, and so forth. A discussion of the principal successful predictions of EPAM, as of a decade ago, will be found in Feigenbaum and Simon (1984), pp. 150-152.

EPAM IV can now account for expert mnemonic performance, performance in several concept-attainment (categorization) paradigms and performance in a substantial number of standard short-term memory paradigms. This article is limited to expert mnemonic performance.

Core EPAM Structures

The core of the EPAM system consists of a small STM and a long-term semantic memory accessed from a discrimination net. These three structures are operated upon by programs of information processes, also stored in long-term memory. Both LTM and the discrimination net expand and are modified by learning processes that operate concurrently with task performance processes like recognition and recall. The EPAM programs are controlled by learned strategies for specific tasks which are invoked by task instructions.

Performance Processes

Stimuli for EPAM are represented by lists of features or attribute-values (Estes, 1994), or by objects that have subobjects and lists of features associated with the objects or subobjects at each level. Similarity between stimuli depends on the number of features, and which ones, stimuli have in common.

EPAM sorts stimuli through its discrimination net to recognize them and gain access to information about them that is stored in long-term semantic memory. The successive nodes through which a stimulus is sorted test the values of the stimulus attributes that select the subsequent node to which it will be passed. Ultimately, the stimulus reaches a leaf node. A leaf stores a partial image of the stimulus (a chunk) together with associations to structures in semantic memory that contain additional information about it. Thus, a chunk is a perceptual unit that has become familiar and recognizable through previous experience with it. The chunking concept dates back to Miller's...
The discrimination net in EPAM performs essentially the same function as the hidden layers in connectionist schemes or the computations that measure similarity in other memory models. Our concern here is whether the EPAM mechanisms are sufficient to account for the observed phenomena and how accurate an account they can give. Comparison with alternatives will have to wait until other theories have been enlarged to deal with the expert memory phenomena considered here.

Semantic Memory

Semantic memory consists of images at leaf nodes of the discrimination net and associative structures of nodes and links, of arbitrary size and configuration, that contain the information used to make responses. The various nodes in the net can be linked with each other through learning of new associations. EPAM's semantic memory, then, closely resembles some other semantic memories that have been proposed in the literature, for example, Quillian's (1968) semantic net and Anderson's (1983) ACT*.

The images and structures in semantic memory (A1) can represent objects or classes of objects of the outside world as sets of attribute-value pairs along with optional lists of images representing subobjects. Thus, the image of car may have the attribute color with value black and subobjects like head, legs, body, tail, and so on.

A node may represent the class of cats. Other leaf nodes represent specific objects, for example, my marmalade cat, Mehitabel. The term object also comprehends larger structures that represent complex situations (scenes or scripts).

Descriptions (full or partial) of objects, then, are stored as images at leaf nodes of the discrimination net or elsewhere in semantic memory. In particular, they may be stored in a retrieval structure which is a specialized and learned tree of nodes and links. Most descriptions contain only incomplete information about their objects. Attributes of objects are represented as slots whose values can be chunks, system primitives or variables. Slots can be instantiated by filling in particular attribute values to represent more specific objects. A slotted schema is a memory structure having at least one variable among its attribute values.

Learning Processes

EPAM is capable of two kinds of learning: (a) It can learn to recognize new stimuli and to discriminate among stimuli previously judged to be the same by adding new tests and branches to its discrimination net (discrimination); and (b) it can store new information about stimuli by elaborating the images at leaf nodes and the associative structures in LTM (familiarization).

Several kinds of indirect evidence indicate that adding a branch to the discrimination net requires 5 s or more; but adding a feature to an image or supplying the value of a variable feature in an associative structure requires only a fraction of a second (Simon, 1976). When a stimulus is recognized, this information becomes accessible, either directly (if it is part of the image), or by associative search from the leaf node through semantic memory.

STM

Both while responding to stimuli and while learning, EPAM holds chunks of symbols in a STM whose capacity in EPAM IV, is limited by the need to rehearse chunks before they fade from memory (Baddeley, 1981; Zhang & Simon, 1985).

Outline of Changes in EPAM IV

EPAM IV elaborates and improves the earlier model of STM by including specialized auditory and visual components, with separate iconic memories for each. In addition EPAM IV's STM holds a small cache of symbols for current inputs and outputs to its processes. These new features bring EPAM's STM model closer to numerous empirical phenomena that have been reported in the literature; but have only secondary effects upon the findings reported in this article.

EPAM IV also makes provision for learned retrieval structures in LTM that augment the other contents of semantic memory, thereby enabling the system to perform difficult mnemonic tasks. The unexpected absence of retroactive inhibition in Charness's chess memory experiments (Charness, 1976) is explained by assuming that adding a branch to the discrimination net will take more than 5 s but adding a value to an existing attribute of a structure in semantic memory will take only a fraction of a second. Subsequently, Chase, Ericsson and Staszewski (Chase & Ericsson, 1981; Staszewski, 1988a) have demonstrated the presence and nature of retrieval structures; but EPAM IV represents the first detailed specification and computer implementation of such a scheme.

Finally, a process has been added to EPAM IV for activating links and nodes of LTM when they are accessed, and a process has been added for searching activated memory along associative paths. These new processes incorporate in EPAM mechanisms that have received strong empirical support from the work of, among others, Quillian (1968) and Anderson (1983).

In summary, EPAM IV contains the following major components (see Figure 1):

A. LTM, consisting of semantic memory, a growing network of list structures — (a) retrieval structures and (b) other semantic memory, organized associatively; a discrimination net that learns, and gives access to semantic memory by recognizing percepts; and other processes (productions) — (a) learned strategies (task specific) and (b) association and activation processes in LTM.

B. STM, consisting of auditory — (a) iconic ("echo box") and (b) STM, with capacity determined by articulatory loop; visual — (a) iconic ("Sperling memory") and (b) imaging memory ("mind's eye"); and cache (WM); small immediate memory for process inputs and outputs.

The basic architecture of EPAM conforms closely to the memory model that has been prevalent in the experimental literature of the past 20 years, following, say, the publication of the Atkinson and Shiffrin (1968) article.

Only the LTM structures affect, significantly, EPAM IV's performance on the tasks reported in this article. The expert memory performances would not be materially altered if the more primitive STM of earlier versions of EPAM had been retained.
Strategies for Evolution of EPAM

EPAM contributes toward the long-run strategy of building a unified theory of cognition (Newell, 1990). Chapter 6 of Newell (1990), especially pages 341-343, shows how EPAM can be accommodated in Soar. However Soar, at present, cannot simulate the perceptual and memory tasks that are central to EPAM.

In general, the modifications made to EPAM do not significantly alter its basic mechanisms, but represent the learning of strategies that allow it to perform new tasks and that typically involve storing new knowledge in LTM. To perform these same tasks, human subjects also have to adopt new strategies and acquire appropriate knowledge. The acquisition of new strategies and knowledge does not alter the EPAM theory but (just as is the case for humans), modifies its performance in new task environments.

In making these extensions, the perceptual and memory architecture of EPAM remains mostly unaltered. The same parameter values for architectural features are retained from one experiment to another, and the extended EPAM is always tested on the old to verify that it performs consistently with the earlier versions over the whole range of tasks previously simulated.

EPAM IV

To insure that EPAM IV can still account for the phenomena simulated previously by EPAM III, it has been tested as a subject in serial-anticipation and paired-associate experiments, producing good replications, both qualitative and quantitative, of the simulation data reported for those earlier versions of EPAM (Feigenbaum & Simon, 1984; Gregg & Simon, 1967; Richman & Simon, 1989; Simon and Feigenbaum, 1964). The present study reports further testing of EPAM IV with data on new tasks of expert memory, specifically, human subject DD’s memory for long strings of digits presented to him at a rapid rate (1 s per digit; Staszewski, 1988a).

By using the same theory, including the same parameters, to account for all of these phenomena, we bring a range of converging operations to bear on the theory, thereby reducing the ratio of degrees of freedom to the number of independent observations to which the theory is fitted.

The only completely satisfactory description of a model of complex cognitive processes is the program itself, accompanied by explanatory text. The most recent version of EPAM IV, written in executable form in Common Lisp and accompanied by explanations of its main processes, is available on computer disk.1

The substantive changes in EPAM IV from EPAM III are as follows: (a) simplification of the learning mechanism, (b) provision for multiple responses to a stimulus, (c) elaboration of the short-term memory mechanism (Figure 1, B), (d) provision for learning in LTM in background at the same time that STM processes (such as rehearsal) occur in awareness, (e) addition of a depth-first search mechanism which can follow pathways through the discrimination net that have previously been activated and (f) addition of a retrieval structure (Figure 1, A1a).

All of these changes, except the last, are changes in the EPAM architecture. In each case they are motivated either by phenomena observed in the experiments with DD or phenomena that have been reported in the literature, or both. Each change is described and motivated separately.

EPAM IV in Skilled Memory Performance

EPAM IV’s performance is based primarily on knowledge it has acquired and stored in LTM, consisting of (a) the retrieval structures (Figure 1, A1a); (b) semantic categories (e.g., running times, ages) for sublists of three or four digits each; and (c) numerical pattern codes (i.e., symmetries like 13-31, 27-27) used to recognize and encode higher-order patterns explicitly (Staszewski, 1990, 1993). Items (b) and (c), both in A1 (Figure 1), offer no novelty. The former are stored as images at the nodes of the EPAM discrimination net, just as other LTM contents are; the latter are implemented by a separate set of recognition processes. There is direct evidence in the data from DD, and also from other subjects, that these three kinds of memory structures and the information in them are in fact created and used in performing the task (Chase & Ericsson, 1982; Staszewski, 1990, 1993).

When the experiments with DD began, he already had stored in memory a substantial amount of information about running times and was capable of recognizing simple patterns in number sequences. Over the course of the 3 years of the experiment, these LTM structures expanded through gradual learning, and DD also stored in memory the retrieval structure, which he elaborated by adding new branches as the lengths of the lists he was recalling grew. The EPAM runs simulated both the performance of the recall task and the long-term learning in semantic memory.

In simulating DD’s performance of the digit retrieval task, at the outset of each trial EPAM uses components already in LTM to construct an empty retrieval structure for the list. EPAM stores information about the digits as they are presented, both in the retrieval structure and with associative links to the chunks that are already available in semantic memory and recognizable as running times, ages, or patterns. Next, EPAM rehearses, using semantic memory to supply information that is missing from the retrieval structure. Finally, EPAM, guided by the retrieval structure and using information stored in that structure as well as information in semantic memory accessible by association and from the discrimination net, recalls the successive groups of digits in the list.

EPAM’s ability to recall lists of slowly increasing length is due to the gradual expansion of the retrieval structure, the gradual accumulation in semantic memory of digit chunks associated with running times and ages, and the gradual expansion of the set of symmetry patterns in digit sequences that it recognizes. The expert performance cannot be produced by any one of these mechanisms operating separately, as even a small probability of forgetting a single item would make error-free retrieval of long lists impossible.

1 EPAM IV can be obtained on 3-1/2-in. disk from Howard Richman, R.D. 2, Box 117, Kittanning, PA 16201. Send $5 to cover shipping and specify whether you want DOS or Macintosh format. The authors will be glad to provide assistance to enable researchers to examine the performance of EPAM IV in detail, to work toward the modification and extension of the theory, or to use the program as an aid in teaching cognitive psychology.
Retrieval Structure

The retrieval structure is treelike, hence it can be viewed as a generalization of the EPAM discrimination net, with slots at each terminal (leaf) node to hold a string of three or four digits together with some information about special features the string may possess. During performance of the memory retrieval task, the retrieval structure is traversed node by node in linear order (i.e., by sequential depth-first search of the tree), thereby retrieving the items in their original order. The EPAM retrieval structure was modeled directly on the one that DD intentionally and explicitly learned and used.

DD is also able to recite the digits under free recall instructions. He orders them according to the categories by which they are stored in the semantic net rather than in the sequence defined by the retrieval structure. The free recall provides direct evidence that the information about the lists, or a large part of it, is stored redundantly in LTM according to two or more distinct classificatory schemes.

Semantic Memory

The semantic memory, indexed by EPAM's standard discrimination net, also holds at its nodes information about possible digit strings, together with their semantic interpretations. Each of the digit strings that is to be recalled is stored (although not always completely or accurately) at two loci — at a node of the retrieval structure and at a node of the semantic memory. These nodes are connected by links (associations) that may run either or both ways. The dual storage provides essential redundancy. There is detailed evidence in DD's data of the way he associates the information in semantic memory with corresponding information in the retrieval structure. The memory structures assumed in the simulation are based directly on this evidence.

The semantic memory is acquired gradually, using EPAM's standard learning processes. The memory is permanent, so that once a chunk has been added to semantic memory, it will remain there and be accessible if the same chunk is seen again in later trials. Associative links between nodes in semantic memory and slots in the retrieval structure are formed during presentation of each list, and their permanence is not assumed or required to account for DD's performance.

Pattern Codes

Numerical pattern codes are also stored in long-term semantic memory, but in the form of discrete tests that can be performed on digits and groups of digits, not as leaf nodes of the EPAM net. These codes contain information about perceptual patterns of digits, describable as symmetries within and between groups. When such patterns are detected, this information is stored, along with information about the group, in the retrieval structure.

DD's data provide direct evidence for his use of pattern codes to augment further the information about each digit group that he stores in the retrieval structure and semantic net. For example, D may encode the fact that the digits of the group 1331 are symmetric or "back to back," or that the first digit of a group is the same as the last digit of the previous group.

Experimental evidence shows that DD recalls the digits more rapidly when lists provide many opportunities for storing pattern information than when the lists provide few such opportunities (Staszewski, 1990). This evidence, together with DD's direct statements about his active search for, and coding of, such patterns, shows how they provide additional redundancy, serving to counteract interference as a result of similarities among the digit groups.

Summary of Encoding and Recall

Before being presented with a list of digits of specified length, EPAM instantiates a retrieval structure containing slots equal in number to the expected number of digits. While the list is being presented, EPAM memorizes the items by inserting successive digits into the corresponding slots of the retrieval structure. At the same time, associations are formed with corresponding chunks already stored in the semantic memory, and with any pattern codes that are detected. After presentation of the list, time is allowed for rehearsal. During this time, EPAM notes missing information in the retrieval structure or semantic memory and fills it in if it is available from one of the other sources.

Subsequently, as EPAM tries to recall successive chunks, it accesses the appropriate (next) leaf node of the retrieval structure; then, by recognition, the corresponding node, if any, in the semantic memory, and any pattern codes that it detects. Combining the information from these three sources almost always provides EPAM with enough total information to recover the digits in the chunk and to report them. It then proceeds to the next chunk until it has recalled the entire list (or failed).

The principal and essential novelty in EPAM IV is the presence of the retrieval structure with its rapidly fillable slots and the association of these slots with learned chunks in semantic memory. The redundancy of information provided by these memory structures plays an essential role in the reliability of recall. The existence of retrieval structures is strongly supported by empirical evidence in all of the tasks domains where such performance has been studied.

Learning Mechanism

In EPAM IV only one node is learned at a time and every node learned is provided with a (partial) image.

Discrimination Net

EPAM IV employs the same discrimination net architecture as did previous versions of EPAM. The discrimination net (Figure 1, A2) is a tree structure. The top node is the ancestor of all of the nodes below it, down to the leaf nodes at the bottom of the net which link the net to semantic memory (A1). An object from the outside world is recognized by being sorted through the net. The principal mode of learning involves growing new nodes in the net.

In the studies reported here, EPAM simulates the discrimination net of DD. Figures 2A and 2B, illustrate learning in the portion of EPAM's discrimination net that discriminates quarter-mile running times from each other.

Figure 2A shows an EPAM IV discrimination net after the quarter-mile times 40.0 s and 50.0 s have been learned completely. The root node holds the various subnets together.
Other subnets sort 1-mile running times, ages and other types of stimuli.

If the quarter-mile running time 50.0 is presented, it will be sorted to the node with the image (5 0 0). The sorting would begin at the root node, which includes a test for type. Then the branch labeled quarter-mile running time is followed to the node that includes a test for first digit. The branch labeled 5 is followed to the node having the image (5 0 0). As no tests are associated with this node (a leaf node), the sorting routine outputs the node and terminates. If the quarter-mile running time 53 s were now presented, it would be sorted to the same node.

Figure 2B shows the same net after a new node, for a run of 53 s, has been learned. A test for second digit has been added to the node, which has the image (5 0 0). The running time would be sorted to the node having the image (5 3).

Multiple Responses to a Stimulus

The simulations using EPAM III never required multiple associations (alternative responses) to a stimulus chunk. However, multiple associations in SAL (Hintzman, 1968), a model similar to EPAM, allowed SAL to perform successfully in simulations of paired-associate experiments in which more than one response to a single stimulus was required. EPAM IV adopts this improvement. In EPAM IV, the image at a node in the discrimination net can include the attribute "instances." The value of this attribute can be a single instance or a set of instances that have been associated with this node.

**STM Mechanisms**

In EPAM II and EPAM III, STM is represented by a small set (7 ± 2) of slots in which information can be held. In EPAM IV, STM is represented in more detail and more realistically by sensory stores (Figure 1, B1a and B2a) and imagery stores (B1b and B2b) in auditory and visual modality and a small push-down stack, or cache (B3) which can hold a few pointers to chunks. These changes relate EPAM more closely with our current empirical knowledge about the structure of STM and related sensory stores (Baddeley, 1981; Zhang & Simon, 1985).

Specifically, the visual and auditory sensory stores correspond to the echoic (Darwin, Turvey & Crowder, 1972) and iconic (Sperling, 1960) memories, and the auditory and visual imagery stores correspond to Baddeley's articulatory loop and visuospatial sketch pad (Baddeley, 1981; Zhang & Simon, 1985).

The push-down stack is used to hold the inputs and outputs of various processes. For example, to rehearse (i.e., refresh) a group of objects (such as words) in the articulatory loop, EPAM IV first recognizes the objects (using the discrimination net), puts pointers to these objects at the top of the push-down stack, and then creates objects in the auditory modality that are placed in the articulatory loop.

**Long-Term Learning in Background**

The fuller specification of STM mechanisms and rehearsal processes in EPAM IV caused us to make a small change to EPAM's serial processing assumption. We adopted the assumption of SHORT (Gilmartin et al., 1976) that LTM learning processes operate in parallel with rehearsal; that is, that these processes occur in background (without awareness) at the same time that processes involving STM (such as rehearsal) occur in the foreground (with awareness).

With EPAM's current parameters each step in LTM learning takes at least 1.75 s. Building new chunks and adding new nodes to the discrimination net is especially costly in terms of time (about 8 s). EPAM has successfully simulated the incremental learning of a response syllable as requiring several learning steps, one at a time (Gregg & Simon, 1967). EPAM IV has successfully matched the total learning time, which is independent of presentation rate, required to memorize stimulus-response pairs (Bugelski, 1962).

**Depth-First Search Mechanism**

The process of recognition — of sorting a set of properties of an object through a discrimination net — is very rapid (EPAM hypothesizes that it takes about 10 ms per test node) and is not accessible to conscious awareness. In recognition, a person is aware of the result (the leaf node in the net that is reached), but not of the path leading to it. People can also access contents of LTM by the more deliberate process of association, which involves moving step by step from one node in memory to a connected (associated) node, using the information at each node to determine what step to take next. EPAM IV is capable of both recognition and association.

**Associative Searches in LTM**

EPAM IV also uses the discrimination net to perform depth-first searches by the process of association, which we hypothesize to take about 250 ms to traverse each node, as compared with 10 ms per test node in recognition. The 250-ms time is consistent with the time, estimated from DD's performance, to move one link upward or downward in the retrieval structure. The association mechanism is used to search for recently activated information in the discrimination net when only incomplete information about the object is available.

Depth-first associative search requires some specific information and employs several special processes:

1. The node where search begins.
2. An exit test, used to determine if the search should be terminated because a node with the proper characteristics has been found.
3. A continue-test routine, used to determine whether search will continue down a particular path.
4. Partial information about the object being sought, which is used to guide the search. This partial information includes the knowledge that the object in question is an example of a slotted schema. The search process considers the members associated with a slot of the schema to be alternative possible locations of the missing information.

Some examples will make clear what is meant by typed slots of a schema. A consonant-vowel-consonant schema has three slots. The first is for an object of type consonant, the second for an object of type vowel, and the third for an object of type consonant. If the word (C_N) is a member of the consonant-vowel-consonant schema, then the vowel slot causes EPAM to
try the vowels a, e, i, o, u, and y in the middle position during its depth-first search for a chunk in the net. Similarly, a three-digit number schema has three typed slots, each representing a digit.

The output of the depth-first routine is either a node in the net that satisfies the exit test or the Lisp symbol “nil,” which is used to indicate that the net has been searched without finding the sought-for information.

**Activation**

Consistently with several other current models of semantic memory (e.g., Soar, ACT*), EPAM postulates that nodes in memory can be activated. Like Act*, EPAM IV employs two kinds of activation: for branches (the arrows of Figure 2) and for nodes. A branch or a node is either completely activated or not activated. Once a branch or a node has been activated, it remains activated throughout that particular experimental “day.” In the interval between sessions it loses its activation. No claim is made that activation is all-or-none, or that it does not fade gradually; but that degree of refinement of the mechanism is not required for the tasks which EPAM has done thus far.

Branches and nodes are activated automatically by EPAM processes. A branch is activated whenever EPAM sorts through it. A branch is also activated whenever it is checked in an EPAM search process to determine its state of activation. As a result, the act of searching the net gradually expands the regions of activation. The depth-first search mechanism restricts EPAM’s search to activated portions of the net. A node is activated whenever new information is added to its image.

**Retrieval Structures**

A major finding of research on memory feats using mnemonics is that, in the course of acquiring the new skill, the expert builds up in LTM a new schema, a retrieval structure, containing slots in which new information can be stored far more rapidly than it could be added to semantic memory. Only specific types of information can be stored in these slots.

The research on rote verbal learning shows that it requires about 8 s to add a new chunk to LTM, and the learning parameters in the EPAM system match quantitatively the average rate of acquisition in rote learning experiments. In contrast, in expert memory performances, items appear to be stored in long-term memory at the rate of one item per second, or (taking account of chunks the expert has already learned) one chunk every 3 or 4 s. Moreover, this material is generally retained longer than is the material acquired in the standard rote learning paradigms using subjects who lack expert memory skills. (For the bases of these parameter estimates, see Newell, 1990, pp. 129-149, 271-273; Newell & Simon, 1972, chapter 14; Simon, 1974; Simon, 1976.)

The experiments with EPAM IV were designed to test whether adding retrieval structures to EPAM would enable EPAM to match the performance of an expert mnemonist.

**DD’s Retrieval Structures**

Chase & Ericsson (1982) and Staszewski (1988a, 1990) have mapped out the characteristics of the retrieval structures that DD uses. Their exact organization varies from trial to trial, being adjusted precisely to the length of the list presented on a trial; however, considerable stability is evident in their organization.

For example, when DD is tested with lists of the same length on different occasions, he reports using the same retrieval structure each time. Moreover, even structures that vary considerably in length share most of the same architectural properties.

Both earlier analyses of DD’s performance (Chase & Ericsson, 1982; Staszewski, 1988a) and a previous information-processing model of his thought processes over the course of a digit-span session (Staszewski, 1993) suggest that DD’s retrieval structure performs at least three distinct functions in a single digit-span trial. First, during list presentation, the retrieval structure guides his parsing of the list into three- and four-digit sublists, each of which is stored as a meaningful chunk. Second, as each such sub-list of digits is stored at a node of the retrieval structure, the node provides the “address” of the sublist. Third, at the time of retrieval, DD’s depth-first traversal of the structure reactivates the addresses in the order dictated by the organization of the structure, and each address cues the retrieval of the content of its slot. In short, DD’s retrieval structures constitute a highly effective implementation of the encoding specificity principle (Tulving & Thomson, 1973).

**EPAM IV’s Retrieval Structures**

In speed of storage, retrieval structures resemble STM structures like Baddeley’s (1981) articulatory loop; but they are LTM structures traversable by (slow) association processes. EPAM IV’s model of retrieval structures has the following properties:

1. Items can be placed in slots on the retrieval structure about as rapidly as they can be articulated: about 300 ms for a one-syllable word (Zhang and Simon, 1985).
2. Each node in the structure can have several typed slots, holding information about the digits stored at that node.
3. Each slot can hold only one of a small set of values of a particular attribute.
4. The system does not process information in the entire retrieval structure at once — just one node. To obtain information that is stored at other nodes, it traverses the retrieval structure at the depth-first (associative) search rate of 250 ms per node.
5. New information that has been stored in a retrieval structure slot can be forgotten. Although it is likely that this forgetting occurs gradually over time, the current simulation assumes one-time loss of a fixed percentage of the information as soon as the system moves its focus of attention away from that node. According to our current estimate, about one fourth of the new information that has been attached to a retrieval structure node is lost when attention shifts from it.

**Simulations**

EPAM IV was used to simulate four major aspects of DD’s skilled-memory performance: (a) his creation of his retrieval structures; (b) his gradual improvement in performance on the serial recall task over the course of about 865 practice sessions; (c) his performance on the serial recall task; and (d) his performance on a free-recall task.

For each task, we determined the strategy that DD gave evidence of using and then programmed that strategy as part of
the EPAM model. The strategies were based on observations of DD while he was doing the same task, and on protocols taken from him during or after performing the tasks.

Hence the strategies are not free parameters, assigned at will to fit the learning data; the data on which they were based are separate from the task performance that we measured. The strategies do not add degrees of freedom to the theory.

Most of the numerical parameters are permanent features of EPAM's structure, estimated from experimental data during early simulations with EPAM (e.g., Gregg and Simon, 1967; Simon and Feigenbaum, 1964) and held constant over the numerous different task environments in which EPAM has been tested; they are not free parameters, and do not introduce additional degrees of freedom into the theory.

Creating Retrieval Structures

DD's retrieval structures for 25-, 50-, 75-, and 100-digit lists have been mapped by Staszewski (1988a), using the explicit comments DD made, while performing the tasks, about the structures he planned to use for specific lists (Figure 3). These postulated structures agree closely with the retrieval intervals between successive digits.

EPAM creates new retrieval structures by chunking together their components. The new chunks can be chunked, in turn, producing recursive branching trees of arbitrary depth, as in Figure 3. Because new structures can be readily assembled from a few basic chunks, because all the structures that DD (and EPAM) uses are modeled on the same basic pattern, and because lists change in length slowly over any sequence of trials, a structure appropriate to the length of the next list announced by the experimenter can be activated rather rapidly in long-term memory.

There are three primitive components, with slots for three, four and five digits, respectively. EPAM IV's chunks can be represented by the following images:

\[
\begin{align*}
&d_3 = (\text{digit digit digit}) \\
&d_4 = (\text{digit digit digit digit}) \\
&d_5 = (\text{digit digit digit digit digit})
\end{align*}
\]

EPAM IV constructs retrieval structures of other lengths following an algorithm that assembles them, recursively, from these basic chunks. The basic pattern is very simple: EPAM begins with groups of 3 or 4 or occasionally 5 digits, then assembles these into groups of 2 to 4, then assembles pairs of these into groups at the next level, and so on.

Applying these productions, EPAM's structures for 25 digits, 50 digits, 75 digits, and 100 digits, which correspond to DD's structures as illustrated in Figure 3, are represented by the following EPAM images:

\[
\begin{align*}
d_{25} &= (d_{16} d_9) \\
d_{50} &= (d_{16} d_21 d_13) \\
d_{75} &= (d_{16} d_21 d_21 d_17) \\
d_{100} &= (d_{16} d_21 d_21 d_21 d_21)
\end{align*}
\]

EPAM's branching factors are four or less, even at the root node, for structures of less than 82 digits. For structures of from 83 to 102 digits the branching factor is five at the root node. For structures larger than 102 digits and less than 124 digits the branching factor is six at the root node. Theoretical models of chunking arrive at optimal chunk sizes of three or four (Dirlam, 1972) or find that the optimum may vary, depending upon assumptions, up to seven (MacGregor, 1987). There is much empirical evidence in the psychological literature that people usually group things by threes and fours (e.g., Broadbent, 1975; McLean & Gregg, 1967; Wicklegren, 1964; Woodworth, 1938, pp. 28-30).

Later, when we compare DD's performance with EPAM's, we will suggest a modification of the top-level retrieval structure that will give a better fit to DD's learning data and will provide a possible explanation for the plateau he experienced after reaching a list length of about 80 items.

Once a retrieval structure of a particular length has been constructed by EPAM, it uses its usual chunk-learning mechanism to memorize that structure so that it can find it in LTM without having to reconstruct it each time it encounters a list of that length.

Learning Curves

DD's gradual improvement in performance and the corresponding improvement of EPAM are shown on the same scale in Figures 4A and 4B. Over the course of 850 serial recall sessions, each on a separate day, DD slowly increased his memory digit-span from the normal range of 7 to 9 digits to a peak of 104 digits. Each time that he was able to recall all of the presented digits in order, the next list presented to him was one digit longer, but when he failed on a list the next list was one digit shorter. The same rule was followed in the EPAM simulations.

Each session involved presentation and recall of from 3 to 25 lists of random digits. The total number of lists presented in a session was gradually decreased by the experimenter as the lists grew longer so as to limit sessions to about 1 hour. In our simulations the same number of lists were presented to EPAM as were presented to DD in the corresponding session.

Prestimulation Learning

Before any simulations began, an initial EPAM net of 109 nodes was created to represent information known to be available to DD before the experiment began or very shortly thereafter. Twenty-one of these nodes were used for translating digits into
their spoken forms. Forty-three of the nodes represented EPAM's initial retrieval structure for lists of 37 digits. The other 46 nodes provided roots for the 14 subnets of the discrimination net assumed to constitute EPAM's pre-existing semantic memory for digit groups. These subnets correspond to the 12 semantic interpretation categories used by DD to perceive and represent three- and four-digit sequences as meaningful chunks (Staszewski, 1990). These interpretations included running times (DD is an experienced runner, highly familiar with times for standard distances), people's ages, and some miscellaneous interpretations.

EPAM's 14 subnets are: (a) quarter-mile running times, (b) half-mile times, (c) three-quarter-mile times, (d) 1-mile times, (e) 3-km times, (f) fast 2-mile times, (g) slow 2-mile times (h) 3-mile times, (i) 10-km times, (j) 10-mile times, (k) dates, (l) ages, (m) double-ages, and (n) miscellaneous salient number patterns (such as 222 or 468). DD possessed his set of categories at the beginning of the experiments and EPAM, like DD, continued to use them throughout the simulations. As in the case of the other observed characteristics of DD that were incorporated in EPAM, these patterns are initial conditions for the system's behavior, and do not introduce new degrees of freedom into the system.

Pattern Codes

EPAM also was given, in the form of productions, a set of pattern codes — symmetries in number patterns that DD recognized and used to help recall the lists. The patterns incorporated in EPAM were based on evidence of DD's use of them. Again, these productions are initial conditions of the simulation, constraining it rather than adding to its degrees of freedom.

Results: Learning During Recall Trials

EPAM was run twice through the entire set of 865 sessions each time being given a different, randomly generated sequence of digits to recall. We have held strategy and availability of retrieval structures constant throughout the simulations. The improvement in EPAM's performance is due solely to the growth in EPAM's semantic memory for digit groups.

We discuss some details of the performance to reveal the kinds of real or apparent fluctuations that can occur in such processes.

As shown in Figures 4A and 4B, in both runs of EPAM its digit span is interleaved with DD's digit span over the first 500 sessions (the first 100 blocks), the average learning rates matching closely. Over the next 250 sessions (50 blocks), DD appears stuck at a digit span of about 80, whereas in the first of its two tests, EPAM continues to climb to a span of about 100, falling back later to 80, with some sizable fluctuations in span from blocks 100 through 140. Over the last 100 sessions (20 blocks), DD resumes his progress, climbing to a span of about 104 digits.

In the second run, EPAM appears to stay on a plateau with a span between 80 and 90 digits from about the 80th to the 150th block, then again increases its span to about 110, to match DD's final performance. As there is nothing in the EPAM mechanism of which we are aware to account for the long plateau, it is most likely simply a product of the particular random number string that EPAM encountered in the second run. The fact that EPAM's learning rate showed as large fluctuations as DD's suggests that the plateau in DD's data may also be an artifact, although we will see later that it can be provided with a possible explanation that does not depend on chance.

EPAM studies (and usually learns) the final group of three to five digits each time a list is presented. Thus, the number of nodes added to the net is a function of the number of lists presented to EPAM and is just about the same for both runs. The exact location of the groups learned will not affect performance noticeably; for when and where a particular group will next appear in a list is unrelated to the location it had when it was learned.

EPAM's semantic memory grows from an original net of 109 chunks to a net of about 3,400 chunks at the conclusion of the 865th session. Both runs follow almost the same growth curve. With the exception of the original 109 chunks, and the 100 or so additional retrieval-structure nodes that EPAM learns prior to list-presentation in the serial recall task, all of the chunks in EPAM's net are learned by studying new digit groups while performing the task. EPAM studies the list and attempts to learn new chunks each time a list is presented.

EPAM's basic explanation for DD's gradual growth in digit span is that DD added about one new node to the net per list presented, at the same time adding to the number of familiar patterns stored in his semantic store of familiar running times and ages. This explanation, however, may be ignoring at least three other factors:

1. Even though DD did derive some benefit from explanations of strategy provided to him by SF, it is likely that he continued to develop his own strategy over the course of the simulations. The need for strategy development would predict a slower start for DD than for EPAM, and this is not evident in the figures.

2. DD began the simulations with a sizable semantic memory for running times. These chunks would predict a quicker start for DD than for EPAM, balancing point 1.

3. It is likely that DD had problems with developing his retrieval structures. His long plateau at about the 80-digit level coincides with the point at which EPAM's retrieval structure first requires more than four branches at the top node. If DD was unable to handle retrieval structures with more than four branches at a node, he would have had to reorganize his structure at this point. This reorganization provides a second possible explanation for the resumption of learning after the plateau.

Despite these possible differences, EPAM's rate of growth in the length of the lists it recalled successfully matches closely DD's growth over the first 100 blocks (about 2 years of daily sessions). We discuss later DD's long plateau that is not simulated by EPAM.

Another similarity is a mild (and possibly illusory) tendency for both DD and EPAM to hit temporary ceilings around mean digit spans of 37, 58, 79 and 100 digits. These digit spans
correspond to points where both EPAM and DD have just added three additional four-digit groups to their retrieval structures. Both EPAM and DD make more errors with four-digit groups than with three-digit groups. EPAM makes more errors with four-digit groups because its discrimination net has learned a higher proportion of the chunks for the 1,000 possible combinations of three digits than for the 10,000 possible combinations of four digits. This can also account for DD's higher error rate with four-digit groups.

**Forgetting**

The match between EPAM's and DD's learning rates depends heavily on the forgetting rate, a parameter that is only moderately constrained by independent evidence. The forgetting parameter determines how much information will be lost after it is initially attached to the retrieval structure. In the simulations reported here the forgetting parameter has been set at 25%, so that one fourth of the information placed on the retrieval structure is forgotten. The EPAM simulation assumed no forgetting in the semantic net, but only in the retrieval structure slots, including the pattern cues.

If the forgetting parameter is set at 20%, EPAM climbs more quickly to a digit span above 100; if the parameter is set at 30%, EPAM may not reach a digit span of 100 over the entire course of 865 sessions. A parameter value of about 25% appears to provide the best fit with DD's performance.

With this parameter available for fitting, it might not be considered surprising that both curves have about the same average slopes, but we will report below closely similar rates (23.6%) for DD's forgetting of patterns, so that the rate we used is not arbitrary.

In Appendix A (not reprinted here), we present an analysis, developed by our colleague, Shmuel Ur, relating the probabilities of forgetting individual items with performance on the task. This analysis allows us to reach at least a qualitative judgment that the assumed forgetting rate of 25% is consistent with what is known about the stability of human memory and with such statistics of forgetting as we are able to extract from DD's performance.

The analysis of forgetting in the Appendix underlines strongly the importance for expert memory of encoding and retaining redundant information (the digits on the retrieval structure, the semantic memory, and the pattern codes) to limit the damage to performance from forgetting. If information were stored without redundancy, then, in lists of 30 digit groups (100 digits), the observed average success rate in recall of one complete list in two would require an average reliability of .977 per group. With a forgetting rate of 25% per digit, and without redundancy, a list of 30 digit groups could hardly ever be reproduced without error (about three times in a trillion trials). On the other hand, if retention of two independent items of information about a group is sufficient for recall of the group, and if the forgetting rate for items were 25%, the chance of retaining a digit group would be .984, in excess of the minimum requirement for reliability of .977, given above.

The actual situation for EPAM was a little more complex: (a) no forgetting was assumed for semantic memory, but at any given trial learning had not yet stored in semantic memory a complete chunk for all digit groups; and (b) information provided by the pattern codes did not fully define all the digit groups.

Nevertheless, our analysis shows to a first approximation why an assumed forgetting rate of 25% per digit in the presence of three independent information sources would produce roughly the observed level of error rates for digit lists.

The retrieval structures alone cannot explain DD's (or EPAM's) ability to retain long lists. The additional information from semantic memory and pattern codes, structures already present in the previous versions of EPAM, was absolutely essential for this feat.

**Serial Recall Task**

The serial recall task is quite simple: the experimenter reads DD a list of random digits at the rate of one per second, then when DD is ready, he repeats those digits back to the experimenter. Actually, the task involves DD in four separate stages of activity: (a) a preparation stage before list presentation, (b) a study stage during list presentation, (c) a rehearsal stage between presentation and serial recall, and (d) a serial recall stage when DD reports the digits in order. Each of these stages is simulated by EPAM.

**Preparation Stage**

Before the list is presented, the experimenter tells DD how many digits it will contain, and DD then tells the experimenter when he is ready for list presentation. DD reports that during the interval he is preparing a retrieval structure for the number of digits that will be presented.

In our simulations, EPAM uses this stage to prepare an instantiated retrieval structure. First it either finds a pre-existing pattern for a retrieval structure in memory or constructs and memorizes a new pattern.

For example, if the list of digits will have 25 digits, EPAM remembers or constructs the retrieval structure pattern that we earlier referred to as "d25." It then instantiates d25 as a node-link tree like the one pictured in Figure 3. Then it makes the root node of the retrieval structure its "focus". The system has a special cell called the RSF (retrieval structure focus) which holds a pointer to a retrieval structure cell. RSF, as used in the remainder of this article, refers to the retrieval structure node that is currently in focus.

**Study Stage**

The study stage occurs while a list of digits is spoken to DD at the rate of one digit per second. DD generally sits still while the digits are being presented, and imposes his own grouping upon them.

In all of our simulations, EPAM follows the same routine for each group of digits. The digits arrive at a simulated rate of one per second. We have estimated the retention of information in the auditory sensory store (often called the echo box) at 3,800 ms, so that about four digits are held there at any one time. This retention time and capacity are consistent with published estimates of the parameters of the auditory sensory store, not to be confused with the articulatory loop (Baddeley, 1981; Zhang and Simon, 1985).

A time parameter is assigned for each step in processing the digits. There are five basic parameters: (a) 100 ms to access a digit in a sensory store, (b) 200 ms to attach a digit to a symbol
structure, (c) 100 ms to notice a pattern cue, (d) 10 ms per node for a recognition search through the discrimination net, and (e) 250 ms per node for an associative search through the net. Parameter (a) is about half of a simple reaction time. Parameter (b) is consistent with earlier estimates that the bulk of EPAM's learning time is devoted to net elaboration, the insertion of information into leaf slots requiring only a short time (Simon, 1976).

Parameters (c) and (d) are consistent with times required for noticing and recognizing in the other tasks to which EPAM has been applied. The 250 ms of Parameter (e) is consistent with measurements of skilled performance. For example, moderately skilled typists transcribe at about this rate per character, and skilled readers read at about this rate per word (Newell, 1990, pp. 236-240). Thus 250 ms can be interpreted as the basic time for accessing well-learned nodes in memory.

The following steps, with the time parameters as indicated above, define the processing strategy:

1. Access the first digit in the auditory sensory store and insert it in the slot of the retrieval structure focus (RSF) node. It takes 100 ms to access a digit from the auditory sensory store and 200 ms to attach the digit to the retrieval structure node.
2. If not running short of time, check the auditory loop to find whether there were back-to-back digits.
3. If back-to-back digits are found, EPAM sorts that information on the image of the RSF node. (It takes 100 ms to check for back-to-back digits. The time required to attach the information to the retrieval structure is 200 ms.)
4. Repeat Step 1 with the second digit in the auditory sensory store.
5. Repeat Step 2 with the second digit, for pairs of back-to-back digits.
6. Repeat Step 1 with the third digit.
7. Convert the information about the three digits into a list, and then sort the list in the semantic net to find their interpretation. For example, if the digits are (4, 2, 4) and the RSF node will hold three digits, EPAM sorts the list (4, 2, 4) in the semantic net. (The semantic net was hand-constructed so as to match the semantic interpretation that DD gives to groups. The portion of the net that sorts lists beginning with a 4 is illustrated in Figure 5. The list (4, 2, 4) sorts to a node whose category is “1-mile time” (i.e., 4 min and 24 s); the list (4, 6, 4, _) sorts to a node with the category “10-mile time” (i.e., 46 min and 40-some s), and the list (4, 2) sorts to a node with the category “quarter-mile time” (i.e., 42 s). The semantic category that has been found is then associated to the image of the parent node of the RSF node. (It takes 10 ms per node to sort in the semantic net; 200 ms to attach to the parent node.)
8. If this RSF node is not the first of the group descended from the parent, and if not running short of time, then examine the RSF parent’s node to see if the RSF node has the same semantic interpretation as the previous node descended from the parent. If so, add to the image at the RSF node the information that the semantic categories are “back-to-back.” (It takes 100 ms to check for code and 200 ms to attach information to retrieval structure.)
9. If this RSF node is the last of the group descended from the parent, and if not running short of time, then examine the parent node in the retrieval structure to determine whether there are any numerical patterns. If a pattern is found, attach that information to the parent node. The patterns include identical categories (i.e., all are dates), symmetry (i.e., mile-time, then half-mile time, then mile-time), ascending progressions, and descending progressions. (It takes 100 ms to check for pattern and 200 ms to attach information to retrieval structure.)
10. Access the remaining digits in the auditory imagery store, and determine how many should be in a group by examining the structure of the RSF node. Nodes for three, four, and five digits each have different structures. (This takes 300 ms per digit accessed, as in Step 1.)
11. Convert the digits and semantic information associated with the RSF nodes and their parents into a list, and then sort that list in the discrimination net. [Eliminate leading zeros before sorting. For example if the digits in the image at the RSF node are (0, 5, 6, 9) and the semantic category is quarter-mile-time, the list (5, 6, 9) will be sorted to a semantic node that holds chunks for quarter-mile times. (It takes 10 ms per node to sort in the discrimination net.)]
12. If not running short of time, test the image at the RSF node to determine if there is a pattern code within the group of digits. If one is found, attach that information to the image of the RSF node. (It takes 100 ms to search for a digit pattern and 200 ms to attach to the retrieval structure.)
13. Note a single difference, if there is one, between the list that was sorted in Step 10 and the image that was reached. For example, if the digits were (5, 6, 9) and the image reached was (5, 6, 8), the difference would be the digit 9 in the third position. Attach information about any difference to the RSF node’s image. (It takes 100 ms to search for a difference and 200 ms to attach to retrieval structure.)
14. Use the association routine to place a pointer to the RSF node upon the image of the semantic node that was found in Step 10. [In EPAM IV, this association process takes about 1.75 s, but once initiated can be carried out in the background, without requiring attention, while the system proceeds with activities that do not require adding information to LTM. The required 1.75 s is available, for associations are only built once for each digit group in the serial recall task, and the digit groups are presented for 3-5 s each (1 s per digit).]
15. Finally, update the RSF by traversing the retrieval structure in a depth-first-search to find the next retrieval structure node and place a pointer to it in the RSF. This search will occur at the same time that Step 12 is occurring in the background. (Time charged is 250 ms per node traversed. At least two nodes are traversed to go from one retrieval structure node to the next.)

Rehearsal Stage

DD reports that during rehearsal he is reviewing the semantic category of each digit group, moving backward through the groups. He discontinues rehearsal when he comes to the four groups of four digits each that are at the beginning of the entire list.

During this stage, EPAM follows the following strategy:

1. First EPAM studies the last group of digits that were presented. In this respect, the simulation does not exactly match DD's behavior, for he consistently rehearses the last two groups and may be learning at other times as well. As we do not have direct evidence to show exactly when DD's learning took place, and as the effects of learning are distributed over the whole performance, we simplified matters by concentrating EPAM's learning at one point in the performance cycle. We are not aware
of any substantial consequences of this simplification for the simulation.

EPAM puts the digits into its auditory imagery store and then rehearses and studies them until they are fully familiar. EPAM also associates the fully familiar chunk with the RSF node. The time required is often extensive, (about 8-10 s if a new node is built in the discrimination net). The process is the same as in paired-associate experiments, with the same time parameters.

2. Next EPAM progresses backward through the list in depth-first search. The time required is about 250 ms per node traversed. Whenever it is found that a node has back-to-back digits or back-to-back categories, or back-to-back digit pairs, the corresponding information is placed upon the previous node.

Also, except when reviewing the first four groups of four digits, EPAM searches for the semantic category and a chunk representing the digit groups in semantic memory described in Steps 3 and 4 of the recall stage. During rehearsal, it discontinues the search for a missing semantic category if it does not find it on the first try.

Thus, throughout the rehearsal stage EPAM follows a procedure closely similar to the one Staszewski (1988a) observed DD following during this stage. This strategy is again one of the initial conditions, or "givens," of the simulation.

Recall Stage

During the recall stage, DD outputs the digits in order, generally group by group. If he can't find a digit, he sometimes goes back to the previous group and tries to see whether it may provide some clues about its successor. If he can't get a whole group, he sometimes skips it, goes on, and comes back to it later.

EPAM goes through the groups one at a time. It traverses the retrieval structure in depth-first search to focus on each successive node. The process iterates through the following steps:

1. Fill in the digits on the retrieval structure using digit pattern and back-to-back digits information.
2. Try to find the semantic category on the image of the RSF node's parent, using any available information about semantic categories that has been stored at the parent node. (DD appears to take this second step before the previous one.)
3. If you cannot find the pattern at the parent node, then conduct a search for the semantic category. Make an educated guess about the category based upon the information about digits at the RSF node's image. Guess what the missing digits could be, then choose a likely semantic category. Then search the subnet of the discrimination net for that category. (A parameter limits the amount of time that EPAM will spend searching for the semantic code of a digit group. These simulations were all conducted with a cutoff of 1 minute on the simulated clock.)
4. Now check the retrieval structure image to see if it gives a complete account of the digits. If not, search the semantic category (if one was found) for an active node that has a pointer to the incomplete RSF node. If the node is found, use its information, and any difference information that was stored with the RSF node image, to fill out the retrieval structure node.
5. Finally, vocalize the digits at a rate of 200 ms per digit.

Pause Latencies

Staszewski (1988a) has measured DD's pause times between digits during serial recall on 29 lists of 100 or more digits each. We have similarly computed two different sets of pause times for EPAM on the basis of different assumptions about the top-level organization of the retrieval structure. The first estimate is based on the retrieval structure for 100 digits that was induced from DD's reports (Figure 6A). The second estimate is based on a slightly modified retrieval structure derived directly from DD's pause times (Figure 6B).

EPAM tends to pause at about the same places for about the same median amounts of time as DD, as shown in Figures 7A and 7B. The Pearson coefficient of correlation between predicted and actual pause times is .932 for Figure 7A, and .964 for Figure 7B.

EPAM's pause time within group boundaries is set at 200 ms, which is the approximate time a person requires to vocalize a digit. (The retrieval structure addresses are used to recover the groups of digits, not the individual digits.) For the pause times between digit groups (except for the first three spikes), the 200 ms required to vocalize a digit is added to the 250 ms required to traverse each node when changing the retrieval structure focus.

Occasionally, EPAM must pause for much longer periods of time at these junctures while it searches the discrimination net for missing information about the digits or semantic category. EPAM's predictions for this part of the curve follow closely the predictions of a simple mathematical model earlier reported by Staszewski (1988a) that had just one parameter — 250 ms to traverse a node — but counted nodes traversed in a slightly different way from EPAM. Both Staszewski's model and Figure 7A are based upon the same diagram of DD's retrieval structure.

Times for the first four digit groups, those that DD did not rehearse before recalling the list, are treated differently in the EPAM model, and as a consequence, EPAM's first three spikes in Figure 7A are quite variable. At a minimum, each spike includes the 200 ms required to vocalize a digit and the 500 ms required to traverse the two nodes to move from one retrieval structure locus to the next. To these times are added the time required to search the discrimination net for information that is missing from the retrieval structure.

Effect of Pattern Encodings on Recall

Staszewski (1990) has reported an experiment to evaluate DD's use of pattern information in the serial recall task. In each of eight experimental sessions conducted on separate days, DD received six digit span trials of 50 digits each. In this experiment, list content was manipulated. Half of the lists were "depleted" so that their number sequences would not contain those patterns that DD consistently recognized and encoded as pattern codes. The rest were "enriched" to provide more than the usual number of opportunities to code pattern relations. DD and EPAM performed this task after achieving a digit span of over 100. The main results of the experiment are shown in Table 1.

The total recall time for both EPAM and DD is lower for the enriched lists than for the depleted lists, although there is a greater difference for DD than for EPAM. EPAM is faster with the enriched than with the depleted lists because the pattern information allows EPAM to note pattern codes in enriched lists during the study stage of the serial recall
task, hence to reconstruct information that was lost due to forgetting in its retrieval structure.

The fact that enrichment facilitates EPAM's recall less than it does DD's is probably due mainly to the fact that in these runs of the system we did not capture all of the patterns that DD uses. DD reports having used each of the patterns that were used in these EPAM runs, but other patterns appear in his protocols as well. For example, if 2-mile times appear back to back, DD will not only note that the categories are the same, but he will note which one is faster. Staszewski (1993), after a more thorough study of DD's numerical patterns, has achieved a more complete match to DD's data.

From DD's use of numerical patterns, we were able to obtain a rough estimate of his forgetting rate, which can be compared with our assumptions about EPAM's forgetting. When EPAM notices a numerical pattern, it attaches information about that pattern to its retrieval structure. This information is subject to the 25% forgetting parameter.

At the end of the presentation of a list, DD gave verbal protocols of his semantic interpretations and pattern codes. Assuming that DD noticed all of the same pattern codes that EPAM did, but then forgot a fraction of them, the proportion of codes reported by DD permits an estimate of the amount of his forgetting on these 50-digit lists.

Before making the comparison we had to eliminate some pattern codes from consideration. For example, in one case, EPAM classified a digit group differently than DD did, finding a pattern in the semantic codes that was not noticed by DD. Also in several cases larger pattern codes subsumed smaller ones even though both were reported by EPAM. For example, if EPAM noticed back-to-back digit pairs in the reverse direction (i.e., 7523 then 2321) it would also notice the back-to-back digits (the 3 at the end of 7523 and the 3 at the beginning of 3291). DD would either report having seen back-to-back digit pairs or back-to-back digits, but seldom both.

After such codes were deleted from consideration, 295 pattern codes remained that were noticed by EPAM during the study stage of the simulation. Seven of these were eliminated from consideration: five instances of a digit group consisting of three identical digits (e.g., 000 or 666) in which EPAM noticed a frontwards-backwards relation that was not reported by DD, and two instances (4051 and 2226) that EPAM described as "eleven-apart" and "four-apart." while DD described them as "add-ups." Of the 288 remaining cases, EPAM reported 218, for a forgetting rate of 24.3%; DD reported 220, for a possible forgetting rate of 23.6%.

Although this result appears to confirm our estimate that DD forgets about 25% of the information that is added to his retrieval structure, we should not put excessive weight on the estimate for at least two reasons. First, instead of forgetting these codes, DD may not have noticed all of the patterns. Second, there is much evidence that DD remembers many codes redundantly. Such extra redundancy would help DD recover pattern codes that might otherwise have been forgotten. The first of these possibilities would suggest that 23.6% would overestimate DD's forgetting rate whereas the second would suggest that 23.6% would underestimate it. Nonetheless, the fact that the observed rate was close to EPAM's justifies added confidence in the model.

**Effect of Trial Order on Recall**

Staszewski's (1990) analysis of recall times in the pattern encodings experiment revealed a general increase in median total recall times (rehearsal time + recall time) as a function of trial order during a single session. DD's results and EPAM's results appear in Table 2.

It is not surprising that EPAM shows a smaller time difference than DD between enriched and depleted lists, for EPAM does not notice as many pattern codes as DD. However, both DD and EPAM show a clear increase in total recall times with trial order. If the enriched and depleted lists are combined, the best fitting linear regression for EPAM's and DD's total times are nearly the same, as follows.

1. EPAM's total time = 6 X trial order + 53 s ($r^2 = .974$)
2. DD's total time = 7 X trial order + 56 s ($r^2 = .544$)

Each additional list during a session takes EPAM about 6 s longer in rehearsal and ordered recall than the previous list, while it takes DD about 7 s longer.

EPAM's recall time increases with trial order because each additional trial on a given day activates more nodes in EPAM's discrimination net, thereby expanding and slowing down the depth-first-searches. This explanation is quite different from the usual explanations of proactive inhibition. But as the experiments reported in the published literature deal with much shorter stimulus sequences than the experiment under discussion here, there is no reason to attribute the proactive inhibition seen under two quite different circumstances to the same mechanism. Proactive inhibition is the name of a phenomenon; it is not an explanation.

**Prediction of DD's Category Choices**

EPAM uses its discrimination net to categorize digit groups according to their semantic interpretation. The initial discrimination net was handcrafted to produce as high a match with DD's semantic interpretations as possible on the set of 798 digit groups that were presented to DD during the free-recall experiments. EPAM's classifications achieved a 99% agreement with DD's classifications for these data. Again, this net serves as an initial condition, representing DD's knowledge of classes of running times at the beginning of the experiment, thereby constraining EPAM's behavior. It is not a free parameter of the model.

It may not be possible to match DD's performance more closely, for DD does not appear to be completely consistent. In fact he reports that, during the study stage of serial recall, he sometimes chooses between two alternative categorizations of a digit group.

**Memory Scanning Task**

When DD had accumulated approximately 790 sessions of practice and his span stood at 90 digits, Staszewski (1988a) conducted a memory scanning experiment with two conditions, an after condition and a before condition. After a 50-digit list had been presented to DD in the usual fashion, groups of digits chosen from the list were presented to him visually. In the after condition, DD's task was to report the digit group that followed the probe group on the list. In the before condition his task was
to report the digit group that preceded the probe group on the list.

EPAM's strategy was the following:
1. Orient to task and turn digit group into a list (estimated
time 1500 ms).
2. Cut a leading zero off the group (if there is one) and use
the usual routine to determine the semantic category (10 ms
per node).
3. Sort the digit group (without the leading zero) in the
subnet for that semantic category in the discrimination net (10
ms per node).
4. Find the retrieval structures associated with the semantic
node that was found. If there are several such retrieval structure
nodes, pick the first one that represents the same number of
digits as the number presented. Make the chosen retrieval
structure the retrieval structure focus (no time charge).
5. If the direction is forward, conduct the usual depth-first
search through the retrieval structure to refocus upon the next
node; if backward, search to refocus on the previous node (usual
time charge of 250 ms per node traversed).
6. If the retrieval structure node has missing information, fill
it out just as it is filled out during the ordered-recall task. This
may involve searching for the corresponding semantic node and
other time-consuming processes (usual time charge of 250 ms
per node).

Both DD and EPAM were able to perform this task quite
accurately as indicated by DD's overall error rate of 5.8% and
EPAM's overall error rate of 1.9%. Both EPAM and DD paused
for median times proportional to the number of nodes traversed at
boundaries between digit groups, as estimated from Staszewski's
(1988a) diagrams of DD's retrieval structure for 50-digit lists.
There was essentially no difference in DD's times for forward and
backward search, and the same times were charged for EPAM's
searches in both directions. Median times for DD and EPAM are
depicted in Figures 8A and 8B for forward and backward search,
respectively. Combining both forward and backward times, we
find that EPAM's response times are well correlated with DD's,
the Pearson correlation between them being .70.

Free Recall Task

At the conclusion of many serial recall practice sessions,
Staszewski (1990) asked DD to recall freely, in any order, all of
the digit groups presented in that session. Staszewski has
studied DD's free recall of 27 lists of about 100 digits each
presented at three lists per session. Staszewski found that DD's
recall is organized by his semantic coding categories and that he
always reports running time categories in an ascending order
followed by dates, ages, and the miscellaneous category. Using a
very strict scoring criterion he found that 94% of the digit groups
recalled are clustered within categories and that the order of items
recalled within each category is from smallest to largest.

To recall the digit groups, EPAM simply searches the
semantic nodes of its discrimination net. Whenever an active
node is found, EPAM accesses the pointer or pointers to retrieval
structure nodes that were stored with its image. Then it fills out
the retrieval structure node's image with information from the
semantic node's image, outputs the digits, and resumes the
search. One hundred percent of the digit groups that EPAM
recalls are clustered within categories.

EPAM never misses a digit group, while DD missed 8% of
the groups. When the order in which the groups are produced by
DD was correlated with the order in which the groups are
produced by EPAM in each of the nine free-recall sessions, the
Spearman rank-order correlation ranged from .782 - .985. (see
Table 3). Four of the nine correlations were above .980 (.981,
.984, .984, .985).

Both DD and EPAM recall the digits by the same categories
and go through the categories in the same order. DD, unlike
EPAM, sometimes returns to a previous category and reports
digits that he had missed earlier. If digit groups appeared twice
within the three-list session, both DD and EPAM recall them
twice, in immediate succession.

EPAM takes less time on average with 100-digit lists than
does DD, as is shown in Table 4. DD's free recall required an
average of 492 s for a three-list session, while EPAM's free
recall took an average 388 s per session, a ratio of about 5 to 4.

As Table 4 shows, EPAM underpredicts the rehearsal and
ordered-recall times for these 100+ digit sessions. The rehearsal
time predicted by EPAM is only about one half of the time taken
by DD (92 s vs. 184 s), and the ordered-recall time taken by
EPAM is about 60% of the time taken by DD (118 s vs. 199 s).

It is possible that EPAM's underprediction of these times is
related to DD's long-lasting plateau at about 80 digits during the
learning stage. When DD tackles lists of more than 80 digits, he
appears to encounter some difficulties that are not explained by
the present version of EPAM.

The serial recall experiments provide various forms of
converging evidence for the existence and organization of DD's
retrieval structures and indirect evidence for the role of the EPAM
discrimination net and semantic memory in enabling his recall of
long strings of digits. The free-recall experiments provide direct
evidence for the existence and organization of DD's semantic
memory and the growth of that memory through learning over
successive trials. Thus, there is extensive converging evidence
that DD's expert performance depends both upon the mechanisms
modeled in EPAM IV's new components, and the mechanisms
already present in previous versions of EPAM.

Conclusion

The simulations that we have reported here provide a broad
outline of mechanisms that are used by DD. EPAM's current
behavior provides a close simulation to DD's performance on
four expert memory tasks, both at an aggregate level and in
many quite specific details. It does this over long stretches of
trials and time, during which substantial learning takes place.
Moreover, it achieves its fit to the data largely using
mechanisms that have already been validated in simulations of
behavior in quite different experimental paradigms and without
change in the parameters that determine the speed with which the
mechanisms operate. The new mechanisms that have been added
to EPAM IV are mostly specific to expert memory performance
and are supported by explicit empirical evidence from the
performance of DD and his predecessor, SF. (Reports on SF's
performance will be found in Ericsson et al., 1980, and Chase &
Ericsson, 1981.) Both EPAM and DD are similar in the
following aspects:

1. Learning. EPAM closely simulates DD's learning curve
for 3 years of daily learning and practice sessions by acquiring
new nodes in a discrimination net, but does not explain why DD
plateaus at a digit span of 80. In its ability to recall very long
lists with an average of only one error in every two lists, EPAM demonstrates the essentiality to expert memory of the redundant storage of information.

2. Pattern codes. EPAM closely simulates DD's documented use of pattern codes and provides an explanation of why lists that have been enriched with pattern codes are more quickly studied and recalled than lists that have been depleted of pattern codes.

3. Proactive inhibition. EPAM simulates the interference that accrues over a six-list session as partly due to growing regions of activation in the discrimination net that cause searches to take longer.

4. Memory scanning. EPAM closely simulates the median time required by DD to scan his retrieval structure from one digit group (given as a probe) to an adjacent group.

5. Pause times. EPAM closely simulates the median pause times in DD's ordered recall of digit lists of more than 100-digits.

6. Times required for various operations. EPAM closely simulates the overall time required for DD in rehearsal, ordered recall, and free recall. EPAM produces excellent fits to DD's data with the one exception that it predicts too brief rehearsal and ordered-recall times for 100-digit lists.

Although it is somewhat unconventional to build a theory of memory processes upon the behavior of a single individual, we were able to match data and theory in exquisite detail, avoiding the "smearing" of small but illuminating effects that occurs when we average data over many subjects. Of course, we do not advocate exclusive use of this strategy over the other.

This study is also limited to behavior in a single (and rather esoteric) task. To establish generality for the results, similar studies will have to be undertaken for a range of tasks. Such an extension has already been carried out for the memories of waiters (Ericsson & Polson, 1988b) and a master playing chess blindfolded (Ericsson & Oliver, 1984), and a model of expert chess memory that incorporates retrieval structures and templates has now been constructed and is being tested (Gobet, 1993). The fact that most of the mechanisms used (with the principal exception of the retrieval structures) have already served within EPAM to explain many perceptual, learning and memory phenomena in other domains gives reasons for being sanguine about generalization.

These results were obtained with an architecture extended from EPAM III by adding components based upon clear and explicit empirical evidence provided by DD's performance and his verbal reports of his memory structures and strategies. EPAM IV is highly responsive to context, and its behavior strongly "situated." It models an individual mind that is in close and constant communication with its physical and social environment.

Like other computer simulation models, EPAM IV attains a high standard of rigor. Its predictions can be tested in detail by running the computer and comparing its output with human behavior. This precision is not acquired at the expense of introducing numerous degrees of freedom. Most of the parameters of the system, and most of the mechanisms incorporated in it, are supported by converging evidence from a variety of experimental settings. Moreover, EPAM can be matched to immense numbers of observations.

When we combine the results reported here with previous findings that show EPAM's ability to simulate human behavior in detail in a wide range of experimental paradigms, we think it fair to claim that EPAM comes closer to providing a unified theory of perceptual and memory processes than any alternative theory that has been proposed to date.

But "unified," of course, is a relative term. A great deal of work lies ahead to extend and test EPAM in concept attainment and categorization, episodic memory, and other semantic memory paradigms to which we are now beginning to apply it.

Effort will be required to integrate EPAM's explanation of perception and memory with theories like Soar, which has approached the integration of cognition starting from the domain of problem solving, or Act*, which has taken semantic memory as its central structure.

References


Note on Figures:

Figures 5 and 9 from the original article were deleted. Here is a summation:

<table>
<thead>
<tr>
<th>Original</th>
<th>Condensed</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2*</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>deleted</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>8</td>
<td>7</td>
</tr>
<tr>
<td>9</td>
<td>deleted</td>
</tr>
<tr>
<td>10</td>
<td>8</td>
</tr>
</tbody>
</table>

Note on Tables:

Table 1 was deleted and the remaining renumbered. Summation:

<table>
<thead>
<tr>
<th>Original</th>
<th>Condensed</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>deleted</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
</tr>
</tbody>
</table>

* For the condensed version only the upper half (A and B) of figure 2 is being used. The lower half (C and D) should be omitted.
Figure Captions

Figure 1. Principal structures of EPAM. STM = short-term memory; LTM = long-term memory; WM = cache (or working memory).

Figure 2. Growth of discrimination net through learning. (See text for further discussion. NEC = not elsewhere classified.)

Figure 3. Retrieval structures for 25-, 50-, 75-, and 100-digit lists.

Figure 4. Learning curves for DD and two runs of Elementary Perceiver and Memorizer, Model IV (EPAM IV). (See text for further discussion.)

Figure 5. Discrimination net for semantic categories. NEC = not elsewhere classified; MISC. = miscellaneous.

Figure 6. Original (A) and revised (B) retrieval structures. (See text for further discussion.)

Figure 7. Comparison of predicted with actual pauses in serial recall for retrieval structures of Figures 6A and 6B. (See text for further discussion. EPAM = Elementary Perceiver and Memorizer, Model IV.)

Figure 8. Median response times in memory scanning task for forward (A) and backward (B) directions. (See text for further discussion. EPAM = Elementary Perceiver and Memorizer, Model IV.)
Table 1
DD's and EPAM's Mean Serial Recall Performance Scores (in Seconds) as a Function of List Type

<table>
<thead>
<tr>
<th>Measure</th>
<th>Enriched Lists</th>
<th>Depleted Lists</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DD</td>
<td>EPAM</td>
</tr>
<tr>
<td>Percent Correct</td>
<td>99.8</td>
<td>99.4</td>
</tr>
<tr>
<td>Rehearsal Time</td>
<td>29.1</td>
<td>39.5</td>
</tr>
<tr>
<td>Recall Time</td>
<td>43.3</td>
<td>46.8</td>
</tr>
<tr>
<td>Total Time</td>
<td>72.4</td>
<td>86.4</td>
</tr>
</tbody>
</table>

Note: EPAM = Elementary Perceiver and Memorizer, Model IV.
<table>
<thead>
<tr>
<th>Trial Order</th>
<th>Enriched Lists</th>
<th>Depleted Lists</th>
<th>Both lists Combined</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DD</td>
<td>EPAM</td>
<td>DD</td>
</tr>
<tr>
<td>1</td>
<td>53.5</td>
<td>57.8</td>
<td>67.5</td>
</tr>
<tr>
<td>2</td>
<td>55.5</td>
<td>62.9</td>
<td>102.5</td>
</tr>
<tr>
<td>3</td>
<td>55.0</td>
<td>69.1</td>
<td>113.0</td>
</tr>
<tr>
<td>4</td>
<td>54.5</td>
<td>69.9</td>
<td>102.0</td>
</tr>
<tr>
<td>5</td>
<td>73.5</td>
<td>79.6</td>
<td>109.0</td>
</tr>
<tr>
<td>6</td>
<td>77.0</td>
<td>77.9</td>
<td>112.0</td>
</tr>
</tbody>
</table>

Note: EPAM = Elementary Perceiver and Memorizer, Model IV.
Table 3
Results of Free Recall for Sessions of Three 100+ Digit Lists for EPAM Compared With DD

<table>
<thead>
<tr>
<th>Digit group no.</th>
<th>No. groups output by both</th>
<th>Same semantic interpretation</th>
<th>Spearman rank-order correlations</th>
</tr>
</thead>
<tbody>
<tr>
<td>858</td>
<td>76</td>
<td>76</td>
<td>.932</td>
</tr>
<tr>
<td>860</td>
<td>78</td>
<td>78</td>
<td>.914</td>
</tr>
<tr>
<td>861</td>
<td>85</td>
<td>83</td>
<td>.868</td>
</tr>
<tr>
<td>862</td>
<td>82</td>
<td>81</td>
<td>.862</td>
</tr>
<tr>
<td>863</td>
<td>80</td>
<td>80</td>
<td>.984</td>
</tr>
<tr>
<td>864</td>
<td>77</td>
<td>74</td>
<td>.792</td>
</tr>
<tr>
<td>865</td>
<td>76</td>
<td>75</td>
<td>.985</td>
</tr>
<tr>
<td>866</td>
<td>81</td>
<td>81</td>
<td>.985</td>
</tr>
<tr>
<td>868</td>
<td>78</td>
<td>77</td>
<td>.981</td>
</tr>
</tbody>
</table>

Note. 87 digit groups were presented each session. EPAM output all of the groups. One group output by DD that did not appear on the lists presented is omitted. EPAM = Elementary Perceiver and Memorizer, Model IV.
Table 4
DD's and EPAM's Mean Performance Times
(in Seconds) on 100+ Digit Lists

<table>
<thead>
<tr>
<th>Measure</th>
<th>DD</th>
<th>EPAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rehearsal Time</td>
<td>190</td>
<td>91</td>
</tr>
<tr>
<td>Ordered-Recall Time</td>
<td>200</td>
<td>118</td>
</tr>
<tr>
<td>Free Recall Time</td>
<td>504</td>
<td>388</td>
</tr>
</tbody>
</table>

Note: EPAM = Elementary Perceiver and Memorizer, Model IV.
Author Identification Notes

We gratefully acknowledge our debt to Shmuel Ur, author of Appendix A of this paper on the effects of redundancy in memory on forgetting. In addition, Dr. Ur and Dr. Fernand Gobet have participated for the past several years in the weekly discussions of our workgroup on expert memory, and their thinking on these matters has strongly influenced ours. In particular, Dr. Gobet contributed the insights he has gained in extending the EPAM IV model to expert behavior in chess, while Dr. Ur shared with us his analyses of the computational theory of discrimination nets.

This research was supported by the National Science Foundation Grant No. DB-912-1027; and by the Defense Advanced Research Projects Agency, Department of Defense, ARPA Order 3597, monitored by the Air Force Avionics Laboratory under contract F33615-81-K-1539. Reproduction in whole or in part is permitted for any purpose of the United States Government. Approved for public release; distribution unlimited.