Psychological Review
1995, Vol. 102, No. 2, 305–330

0033-295X/95/$3.00

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 E. A. Feigenbaum in 1959, that 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 Ericsson (1981, 1982) and Staszewski (1988a, 1988b, 1990). 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 great interest and much empirical research has focused on the abilities of experts in several domains 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).

What is it at stake here is not just the vast store of knowledge about a domain that experts in the domain generally hold in memory, but in particular the experts' abilities to store new information rapidly for later retrieval. Extrapolation from the results of verbal learning experiments and theories of memory based on them (see Baddeley, 1981) would predict that short-term memory (STM) has insufficient capacity to retain the information presented and also that the information could not be stored in long-term memory (LTM) at the rapid rates at which it was presented (Bugelski, 1962; Simon, 1976). The problem is to explain how, under these circumstances, the information can be acquired and retained within the narrow time limits allowed.

This article shows how these and similar phenomena, which have not been dealt with by other broad-gauged models of memory with which we are familiar, can be explained by the Elementary Perceiver and Memorizer, Model IV (EPAM IV) program. EPAM IV is 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) and an application to chess has been programmed by Gobet (1993), but the mechanism has not previously been incorporated in a comprehensive model of perception and memory.

A well-known example of a retrieval structure is that used in the so-called Method of Loci, used by orators and others since Greek and Roman times to facilitate retention of lists (Chase & Ericsson, 1982; Luria, 1968; Mitchell, 1939; Yates, 1966). In one version of this method, the plan of a building ("memory palace") is memorized thoroughly, usually including major pieces of furniture in each room, until the subject can traverse the building and the rooms mentally, systematically, and without hesitation. When a list of items is now presented for memorization, successive items are associated with successive loci in the building; and when recall is required, it is achieved by visiting these loci and noting what items are stored there. The slots referred to in the description of retrieval structures are the loci where information can be stored rapidly and retained.

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

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

This research was supported by National Science Foundation Grant 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 U.S. Government. Approved for public release; distribution unlimited.

Correspondence concerning this article should be addressed to Herbert A. Simon, Department of Psychology, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213. Electronic mail may be sent via Internet to has@cs.cmu.edu.

It appears that experts in a domain that makes heavy demands on memory commonly possess retrieval structures that they have learned, deliberately or incidentally, while acquiring expertise in the domain. Experimental data reveal considerable detail about the form that such schemas assume in several task domains in which expert memory has been studied, 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, it appears that the schemas are based on the pattern of a chess board on which squares have slots with which pieces, patterns of pieces, and other information can be associated and on the typical patterns of pieces in chess openings that are known to master players (Gobet, 1993). The latter patterns, gradually acquired in the course of study and play, can be called *templates* to distinguish them from the deliberately acquired retrieval structures that are of principal concern in the present study. Templates contain information about a familiar situation, as well as slots for new information. The retrieval structures discussed here contain only slots.

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 reported that the mnemonists made use of previously acquired semantic memory, retrieval structures, or both and that retention of sequences, although not quite as long as those reported here, exceeded 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. We have noted that retrieval structures also appear to be essential to expert memory in chess and in other domains, but 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, their interactions with LTM, and their contribution to the redundancy, and hence reliability, of the information stored.

A detailed model of a single subject carrying out a single, rather unusual, task would not be of great 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 general theory of this kind of expertise and to test it in great detail with the data of one subject. As the data are very extensive, they permit an unusually close investigation of even the details of the theory and the processes it embodies.

Introduction and Overview

In the first section of the article, we briefly describe the tasks performed by the subject DD 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 in particular 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, who learned to perform the task at a very high expert level through more than 3 years of nearly daily training.

We mention, from time to time, other explanations for memory phenomena that have been proposed in the literature; but as we are not acquainted with any other theories that undertake to account, either in quantitative or even qualitative detail, for the data of 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 (even those of experts) in a general model of memory, we comment on other theories only cursorily. This is not to deny the importance of attempting such an extension for other theories, but the attempt is best made by those who have developed those theories and who are thereby most likely to be able to elaborate and generalize their mechanisms to account for the new phenomena.

In discussing the changes introduced into EPAM from its earliest versions through EPAM IV, 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 the contents of the subject's memory at each subsequent stage. They represent the subject's learning prior to and during the experiment, learning the nature and extent of which is substantiated by extensive empirical evidence. 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. To do this, he created and learned a sequence of larger and larger retrieval structures, which he used as mnemonic aids to storing and recalling the digits. At the same time he learned, and stored in semantic memory, three- and four-digit series 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. However, he recalled them not in the 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 task casts further light on the cues on which he relied for recall.

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

The history of the EPAM system goes back more than 30 years. 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, accurately and usually in quantitative detail, 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 multitrial learning, and so forth. A list and discussion of the principal successful predictions of EPAM, as of a decade ago, can be found in Feigenbaum and Simon (1984, pp. 150–152).

Although EPAM's ability to model verbal learning in the standard experimental paradigms attracted considerable attention to it, that attention faded with the shift of experimental activity to free-recall tasks and subsequently to categorization tasks. Theories dealing with these latter tasks have generally taken a more abstract and algebraic form. (For recent surveys see Estes, 1991 and 1994.) It was not widely recognized that the mechanisms of EPAM, if they represented a basically correct picture of perceptual and memory mechanisms, should be able to give an account of behavior in these other kinds of experiments also.

With the improvements in the theory incorporated in EPAM IV, EPAM can now account for expert mnemonic performance like that discussed in this article, performance in several concept-attainment (categorization) paradigms and performance in a substantial number of standard short-term memory paradigms. Concept attainment and STM paradigms and expert performance in chess-playing environments will be examined in separate articles; this article is limited to expert mnemonic performance.

Core EPAM Structures

The EPAM theory was developed to identify a basic set of mechanisms that would give a unified account of diverse perceptual and memory phenomena. The phenomena surrounding verbal learning were selected as the core, and the program has been gradually extended, while retaining its central mechanisms, to encompass other task domains. From EPAM's first versions to the present one, the core of the EPAM system consists of a small STM that can hold a few familiar chunks of knowledge and a long-term semantic memory accessed from a discrimination net. These three structures are operated on by programs of information processes, also stored in long-term memory. In particular, 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 that are invoked by task instructions.

Performance Processes

Stimuli for EPAM are represented by lists of features or attribute values (Estes, 1994) or, in the case of more complex stimuli, by objects that have subobjects (and possibly sub-subobjects, etc.) and lists of features associated with the objects or subobjects at each level. Features in EPAM do not have weights. Similarity between stimuli depends on the number of features, and which ones, stimuli have in common. We believe that none of the phenomena examined in this article would permit us to choose between alternative representations of similarity.

In performing perceptual and memory tasks, 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 in the net through which a stimulus is sorted contain tests on the values of the stimulus attributes that select the subsequent node to which it will be passed. Ultimately, the stimulus reaches a *leaf node*, which serves as an interface between the discrimination net and semantic memory. A leaf node contains no tests, but instead stores a partial image of the stimulus (a *chunk*) together with links (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 (1956) celebrated "magical number" paper and has been a central feature of all versions of EPAM.

The discrimination net in EPAM performs essentially the same function as the hidden layers in connectionist schemes or the more abstract computational schemes that measure similarity in other memory models. It is not necessary in this particular study to address the question of whether empirical tests can be devised to choose among these mechanisms or computational schemes. 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. Through the associative links, 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 other structures in EPAM IV's 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 *cat* may have the attribute *color* with value *black* and subobjects like *head, legs, body, tail,* and so on. Each value in an attribute-value pair and each subobject on the list of subobjects can represent either another object or a primitive (unanalyzed) feature.

Some objects are categorical: For example, a node may represent the class of cats. Other leaf nodes represent specific objects, for example, my marmelade cat, Mehitabel. The term *object* also comprehends larger structures that represent complex situations (scenes or scripts) involving relations among sets of objects. In the current implementation, the ASCII characters serve as the system primitives.

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. A complete copy of an object contains enough information to reproduce the object, but 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. An uninstantiated slot, then, does not have a wholly definite image associated with it, but can be filled by any one of a set of chunks, system primitives, or both. For example, the allowable values for a particular slot may be any one of the digits from 0 to 9.

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*). The discrimination net and associated images are part of LTM, as are the processing mechanisms that EPAM uses for discriminating and learning.

Several kinds of indirect evidence indicate that adding a branch to the discrimination net is a slow process, requiring 5 s or more; but adding a feature to an image or supplying the value of a variable feature in an associative structure is a relatively rapid process, requiring 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 limited STM whose capacity, in previous versions of the theory, was limited to a fixed (small) number of chunks. In contrast, in EPAM IV it is limited by the need to rehearse chunks before they fade from memory (Baddeley, 1981; Zhang & Simon, 1985). For the range of experiments run on previous versions of EPAM, these two forms of STM limitation produce nearly identical effects on performance.

Outline of Changes in EPAM IV

EPAM IV, the version that is reported here, elaborates and improves the earlier model of STM by including specialized auditory and visual components in STM, 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 they have only secondary effects on the findings reported in this article.

As we have already seen, 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 idea that instantiating existing memory structures might require less time than learning new discriminations dates back to Simon (1976). He observed that the unexpected absence of retroactive inhibition in Charness's chess memory experiments (Charness, 1976) could be 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 and their efficacy in accounting for the performance of skilled mnemonists; but EPAM IV represents the first detailed specification and computer implementation of such a scheme in the context of a general model of perception and memory.

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, which incorporate in EPAM mechanisms that have received strong empirical support from the work of, among others, Quillian (1968) and Anderson (1983), do play a significant part in the performance reported here.

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

A. LTM, consisting of semantic memory, a growing network of list structures (also known as node-link structures, schemas, frames, scripts)—(a) retrieval structures and (b) other seman-

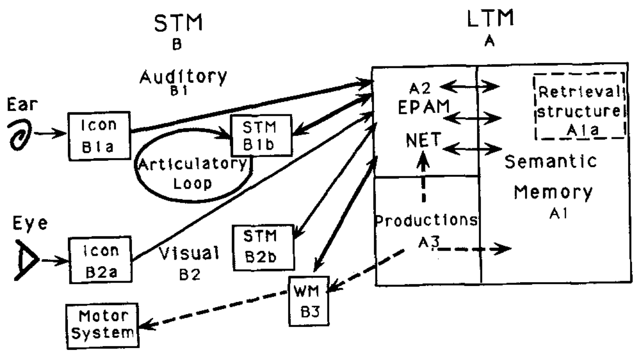

Figure 1. Principal structures of EPAM. STM = short-term memory; LTM = long-term memory; WM = cache (or working memory).

tic memory, organized associatively; a discrimination net that learns, providing a growing "index" that 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.

A glance at this list and at Figure 1 shows that the basic architecture of EPAM, although not always the details of its design, 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 elaborations of STM were introduced to take account of phenomena, noted earlier, that are well established in the experimental literature, and comparisons of the EPAM predictions for these phenomena with human data will be undertaken in later articles. But the expert memory performances considered in the present article would not be materially altered if the simpler and more primitive STM of earlier versions of EPAM had been retained, for the phenomena addressed here are predominantly LTM phenomena.

Strategies for the Evolution of EPAM

EPAM contributes toward the long-run strategy of building a unified theory of cognition (Newell, 1990). It incorporates major STM and LTM mechanisms, learning mechanisms, and the perceptual (but not the sensory) "front end" required for such a theory. Chapter 6 of Newell (1990), especially pages 341–343, shows how EPAM can be accommodated in Soar, a leading proposal of a comprehensive architecture for a unified theory. However, Soar, at its present stage of development, cannot simulate the perceptual and memory tasks that are central to EPAM.

In general, the modifications made to EPAM over the years 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. Although strategies and knowledge are not part of the core theory, they set boundary conditions to the performance of particular tasks. 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. The importance of distinguishing architecture from strategies in cognitive theories is discussed in chapter 14 of Newell and Simon (1972).

In making these extensions, the perceptual and memory architecture of EPAM remains mostly unaltered (exceptions that constitute genuine changes in the theory will be mentioned explicitly). The same parameter values for architectural features are retained from one experiment to another, and the extended EPAM is always tested on the old as well as the new tasks to verify that it performs consistently with the earlier versions over the whole range of tasks previously simulated. This cumulative consistency is essential, of course, if we are to construct a genuinely unified theory.

EPAM IV

We have already sketched the architecture of EPAM IV, but we must now provide more complete information about the changes that distinguish it from EPAM III, and especially about those structures and processes that are important for its performance in expert memory tasks.

To ensure 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 & 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).

The performance of EPAM IV demonstrates that these memory data and the mechanisms that have been postulated to account for them are wholly consistent with the mechanisms used to explain the wide range of perceptual and memory phenomena for which EPAM has previously been tested. 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]

[1] EPAM IV can be obtained on 3½-in disk from Howard Richman, RD 2, Box 117, Kittanning, Pennsylvania 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 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 STM mechanism (Figure 1, B), which was quite skeletal in previous versions of EPAM; (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 that 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 the result rather directly of either 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, but first we describe how EPAM accomplishes the skilled memory task: recalling long strings of rapidly presented digits.

EPAM IV in Skilled Memory Performance

EPAM IV's performance on the skilled memory task 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 could be handled in exactly the same way, but in the current version they are implemented by a separate set of recognition processes. There is direct evidence in the data from DD, and also from other subjects whose expert memory performance has been studied, 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. (This information was also supplied to EPAM at the outset of its runs.) 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 digits 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.

For reasons that will become clear in the course of our analysis, 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. Reliability in performance requires using in combination the information stored in the regular EPAM semantic memory (including the pattern codes) and the information stored in the retrieval structure.[2]

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. Storing information about the items to be retrieved in the successive slots of the retrieval structure preserves, at least for the duration of the task, the information about both the items and their sequence. 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. Later, we describe in detail the EPAM retrieval structure, which was modeled directly on the one that DD intentionally and explicitly learned and used.

As we shall see, DD is also able to recite the digits under free-recall instructions. However, with these 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. This proves to be essential for DD's ability to perform the serial retrieval task at a high level of accuracy.

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 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.

[2] On the underlying theory of reliability, see Von Neumann (1956). For the application of the theory to the present task, see the Appendix of this article.

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 organizes and uses semantic memory and 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 over the whole sequences of trials, using EPAM's standard learning processes, as EPAM tries to recall lists of greater and greater length. 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.[3] These codes contain information about perceptual patterns of digits, describable as symmetries within and between groups. When such patterns are detected in a digit group presented to EPAM, 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. DD creates the codes to augment further the information about each digit group that he stores in the retrieval structure and semantic net. For example, DD 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. The pattern codes, whenever they are detected, are associated with the corresponding digit groups in the retrieval structure. The specific pattern codes that DD (and EPAM) used are described later in more detail.

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 the interference that builds up within and across trials 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. Hence these slots are filled (at less than 1 s per digit) as each new list is encountered. 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 progresses through a list trying 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, gradually acquired through extensive practice, and their role in skilled memory performance are strongly supported by empirical evidence in all of the task domains in which such performance has been studied.

Learning Mechanism

In EPAM IV the learning mechanism has been simplified. Whereas in EPAM III learning a new node in the net was often accompanied by automatic learning of several other nodes including empty imageless nodes, 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 uses the same discrimination net architecture as did previous versions of EPAM. The discrimination net (Figure 1, A2) is a tree structure, somewhat like a family tree. The top node in a net, the root node, is the ancestor of all of the nodes below it. It has children, and its children have children, all the way down to the leaf nodes at the bottom of the net, which have no children of their own but link the net to semantic memory (A1). An object from the outside world is recognized by being sorted through the net using tests that are associated with each node along the way. The principal mode of learning involves growing new nodes in the net.

In the studies of expert memory reported here, EPAM simulates the discrimination net of DD, an experienced runner who often recognized groups of digits as running times. Figures 2A, 2B, 2C, and 2D 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 com-

[3] Contrary to our initial expectation, there now seems to be no need to distinguish pattern codes from the other contents of semantic memory, and in subsequent versions of EPAM we will undertake to remove their special status.

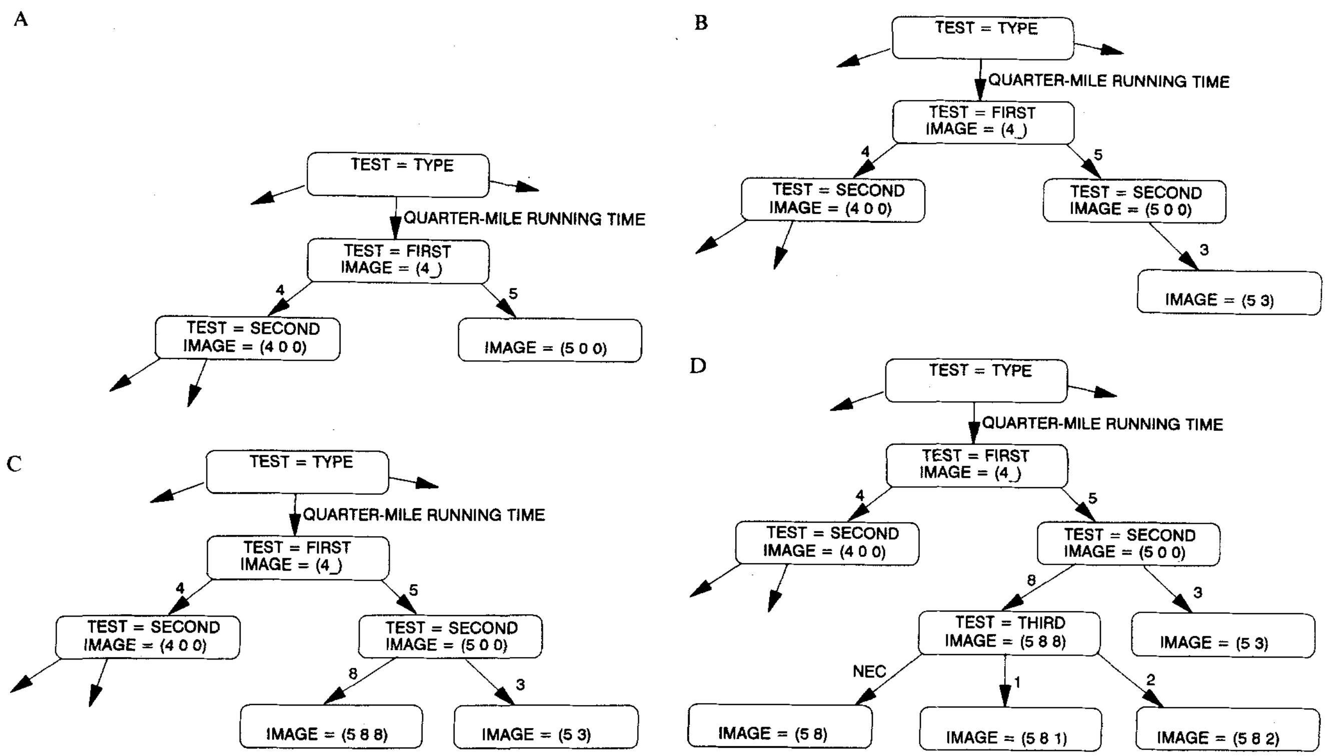

Figure 2. Growth of discrimination net through learning. (See text for further discussion. NEC = not elsewhere classified.)

pletely. The root node in the net has a test for type; this root node holds the various subnets together. Other subnets sort 1-mile running times, ages, and other types of stimuli. Once EPAM knows that a particular stimulus is a quarter-mile running time, it sorts that stimulus to the top node for this type of stimulus. The top node in the net for quarter-mile running times is the node pictured in Figure 2A with a test for first digit. The arrows without destinations are included in this figure to indicate that there are other areas of the net not pictured.

If the quarter-mile running time 50.0 is presented, it will be sorted to the node with the image (5 0 0).[4] 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. As the first digit is 5, 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). At this point, the running time 53 s would be sorted to the node having the image (5 3). If the quarter-mile running time 58.8 s were now presented it would be sorted to the node having the image (5 0 0). As there is no branch at this node for the digit 8, the sorting routine outputs this node even though it is not a terminal node.

Figure 2C shows the same net after a chunk for 58.8 s has been completely learned. At this point, any running time beginning with the digits 58 would sort to the chunk with the image for (5 8 8).

Figure 2D shows the same net after chunks for 58.1 s, 58.2 s, and 58 s have been learned. The branch to the node with the image (5 8) has been labeled *NEC*, which stands for *not elsewhere classified.* The NEC branch is followed in EPAM IV when the stimulus cannot be tested by the test at a node. Because 58 has no third digit, the NEC branch is followed from the node with the test for third digit to the node with the image (5 8).

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. This mechanism also enabled SAL to simulate changes in response time over the course of a simple paired-associate experiment. 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.

[4] For clarity of presentation the association list belonging to each image has been omitted from Figure 2.

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), that 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). Some of these structures and the processes that manipulate them were borrowed from SHORT (Gilmartin, Newell, & Simon, 1976), a model of STM under strategic control.

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 visuospatial sketch pad is referred to by others as a mental image, or the mind's eye. The references cited above provide strong empirical evidence for these components of STM. The contents of a store are a list of objects (in the EPAM IV sense) with tags attached to each object to indicate how much longer it can remain in the store before it will disappear and how it has been grouped.

The push-down stack is used to hold the inputs and outputs of various processes, which get their inputs and then leave their outputs, if any, at the top of this stack. For example, to rehearse (i.e., refresh) a group of objects (such as words) in the articulatory loop, EPAM IV first recognizes the objects in the group (using the discrimination net), puts pointers to the leaf-node chunks for these objects at the top of the push-down stack, and then articulates the images of the chunks, creating 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. If rehearsal in the articulatory loop were to pause while EPAM is engaging in long-term learning, then information would be lost from the loop. Instead 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). We are currently conducting additional tests of this amplified short-term memory in EPAM IV with simulations of key STM experiments in the literature. The findings will be discussed in subsequent articles.

With EPAM's current parameters each step in LTM learning takes at least 1.75 s. In nonsense syllable learning, several steps are required to learn new chunks. Building new chunks and adding new nodes to the discrimination net are 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). In a paired-associate experiment, EPAM can be given the strategy of rehearsing the stimulus and response objects, thus keeping them available, while it engages in the required learning steps. If other incoming stimuli are ignored, EPAM can persevere until the stimulus elicits the complete response. 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

Previous versions of EPAM have used the discrimination net only for recognition processes. 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. Of course total associative reaction times in response to external stimuli are much longer than this, but we are dealing here with a single step in a wholly internal search with no time required for sensory processing. 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 uses the following 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. In particular, the test can be used to find a node in the discrimination net that points to (associates to) a particular node in the retrieval structure.
3. A continue-test routine, used to determine whether search will continue down a particular path. This test can be used to keep the search within activated regions of the discrimination net.
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. If the number (45_) is a member of this schema, then the third slot causes EPAM to try the digits 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9 in the third position when it searches the net.

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

Consistent 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 uses two kinds of activation: for branches (the arrows of Figure 2) and for nodes. EPAM's simulation of activation is currently quite simple, but adequate for the cognitive tasks to which it has been applied here. 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, and the next day it will no longer be activated. 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 that EPAM has done thus far.

Branches and nodes are activated automatically by EPAM processes. One process activates branches, and a different one activates nodes. A branch is activated whenever EPAM sorts through it, thereby activating pathways through the discrimination net. 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. The exit-test routine in depth-first search can consider nonactivation of a node as one of the reasons to exit there.

Retrieval Structures

The specific retrieval structures used in EPAM IV have already been described. We need to examine them now in a more general context of research on memory and mnemonics. 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. A schema with slots for numbers cannot store letters, and vice versa; a schema with slots for chess pieces cannot store playing cards.

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, 1952, chapter 14; Simon, 1974; Simon, 1976.)

The experiments with EPAM IV were designed to test whether adding retrieval structures to EPAM, and assuming (for both retrieval structures and other semantic memory) that filling an open slot takes much less time than is required to add a new node to a discrimination net, would enable EPAM, using the redundant memory resources of the retrieval structure and the semantic memory, to match the performance of an expert mnemonist. The hypothesis that filling a slot is a more rapid process than building a node was initially proposed in 1976 (Simon, 1976, pp. 87, 91) to explain unusual memory performances using mnemonic schemes and was more fully developed by Chase and Ericsson (1982) and Ericsson and Staszewski (1989), but this is its first implementation in a comprehensive theory of perception and memory.

DD's Retrieval Structures

Chase and Ericsson (1982) and Staszewski (1988a, 1990) have mapped out the characteristics of the retrieval structures that DD uses when he recalls, in order, up to 104 digits presented to him at a rate of about 1 per second. As we have already seen, DD's retrieval structures are hierarchically organized. 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, as the retrieval structures shown in Figure 3 indicate, 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 sorted through the discrimination net to code and store it as a meaningful chunk. Second, as each such sublist 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 coordinate his list encoding with his list retrieval processes, constituting a highly effective implementation of the encoding specificity principle (Tulving & Thomson, 1973).

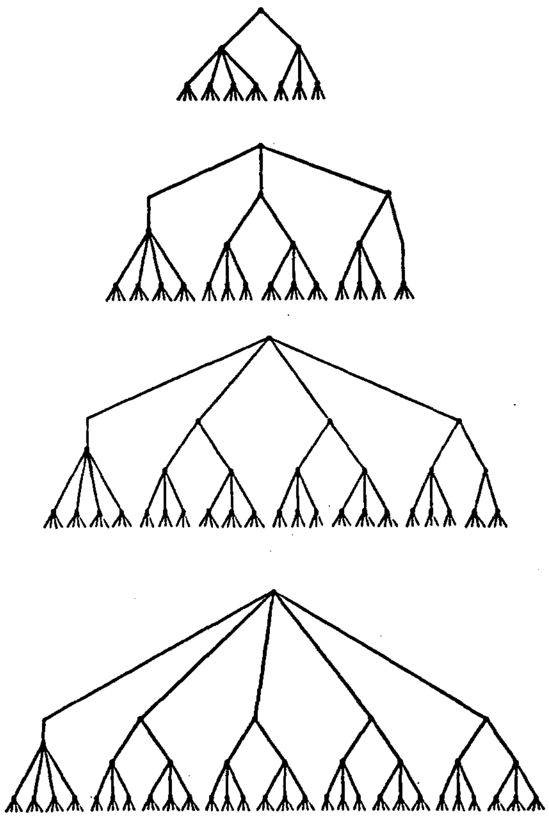

Figure 3. Retrieval structures for 25-, 50-, 75-, and 100-digit lists.

EPAM IV's Retrieval Structures

In speed of storage, retrieval structures resemble STM structures like Baddeley's (1981) articulatory loop; but in more fundamental ways, they are LTM structures at the ends of the discrimination net branches, traversable by (slow) association processes. EPAM IV's model of retrieval structures for memory of number sequences 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 & Simon, 1985).
2. Each node in the structure can have several typed slots, holding information about the digits stored at that node. Information that fills these slots constitutes an image attached to the node.
3. Each slot on a node can hold only one of a small set of values of a particular attribute. For example, most of the slots on DD's retrieval structure nodes can hold only a single digit each.
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. Unlike information stored in the discrimination net itself, 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. The loss of information at a particular node takes place as soon as the system moves its focus of attention away from that node and its children. 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, each on a different day; (c) the detail of 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 when performing that task and then programmed that strategy as part of the EPAM model. The strategies that were built for EPAM 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 behavioral data and verbal reports on which they were based are separate from the quantitative task performance that we measured. The strategies are best viewed as (learned) initial conditions for the simulation that do not add degrees of freedom to the theory. This section reports our simulations and compares EPAM's behavior with DD's behavior.

Most of the numerical parameters used in the simulation are permanent features of EPAM's structure, estimated from experimental data during early simulations with EPAM (e.g., Gregg & Simon, 1967; Simon & Feigenbaum, 1964) and held constant over simulations of behavior in the numerous different task environments in which EPAM has been tested. To the extent that they derived from evidence that is independent of the current experiments on expert memory, they are not free parameters. Hence they do not introduce additional degrees of freedom into the theory, but instead add constraints to which it must conform. When we make use of particular parameters, we will provide some indication of the sources for our estimates of them.

Creating Retrieval Structures

DD developed his retrieval structures on the basis of advice given to him by SF, the first subject in Chase and Ericsson's pioneering study of skilled memory (Chase & Ericsson, 1982). 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 and about the locations in the structures of specific groups of digits. Staszewski's diagrams of these structures appear in Figure 3. As we shall see later, these postulated structures agree closely with the retrieval intervals between successive digits, although we will also see that we can improve the fit by altering the 100-digit structure somewhat.

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. (DD's protocols suggest that he may also use a two-digit chunk and that his five-digit chunk is a composite of a two-digit with a three-digit chunk. The current version of EPAM IV differs from DD in this respect, but the difference will not have an observable effect on performance of the tasks analyzed here.) EPAM IV's chunks can be represented by the following images:

d3 = (digit digit digit)

d4 = (digit digit digit digit)

d5 = (digit digit digit digit digit)

EPAM puts together larger retrieval structures by chunking smaller structures. Following Staszewski's (1988a) diagrams, it is given the following patterns for structures of 6, 7, 8, 9, 10, 11, 12, 13, 14, 16, 21, 22, and 23 digits. (These patterns again serve as initial or boundary conditions for the theory and do not add degrees of freedom.)

d6 = (d3 d3)	d7 = (d3 d4)	d8 = (d4 d4)
d9 = (d3 d3 d3)	d10 = (d3 d3 d4)	d11 = (d3 d3 d5)
d12 = (d4 d4 d4)	d13 = (d4 d4 d5)	d14 = (d4 d4 d3 d3)
d16 = (d4 d4 d4 d4)	d21 = (d9 d12)	d22 = (d9 d13)
d23 = (d9 d14)		

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: The initial component of the structure is always d16; this is followed by a series of structures of the form d21 = (d9 d12). The other productions group any remainders of less than 21 digits, using structures built up mainly from d3 and d4. In other words, EPAM begins with groups of 3 or 4 or occasionally 5 digits, then assembles these into groups of 2 to 4 digits, 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:

d25 = (d16 d9)	d50 = (d16 d21 d13)
d75 = (d16 d21 d21 d17)	d100 = (d16 d21 d21 d21 d21)

Many other patterns are also possible. In fact, according to Chase and Ericsson's (1982) diagrams, DD's predecessor, SF, combined components to produce somewhat different retrieval structures from those chosen by DD. Despite these differences of detail, all of these structures are trees with two to four branches at the nodes below the root node (the highest branching level).

EPAM's branching factors are four or less, even at the root node, for structures less than 82 digits. For structures 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 on 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). Mandler (1967, 1975) has found evidence for groups of five, but we believe that there is a preponderance of evidence for the values three or four.

Later, when we compare DD's performance with EPAM's, we will have more to say about the large branching factor at the top of the tree, and 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 long (more than 1 year) 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. A consequence of this procedure is that DD and EPAM will each make, on average, about one error for every two lists attempted—a little less when performance is improving through learning and a little more in periods when performance is deteriorating.

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, as shown in Table 1. In our simulations the same number of lists were presented to EPAM as were presented to DD in the corresponding session.

We now examine in more detail how this generally close correspondence of EPAM's learning with DD's came about. First, we consider the learning that was postulated to have occurred before the test trials, that is, what DD and EPAM were assumed to know when the experiments began. Next, we analyze the test

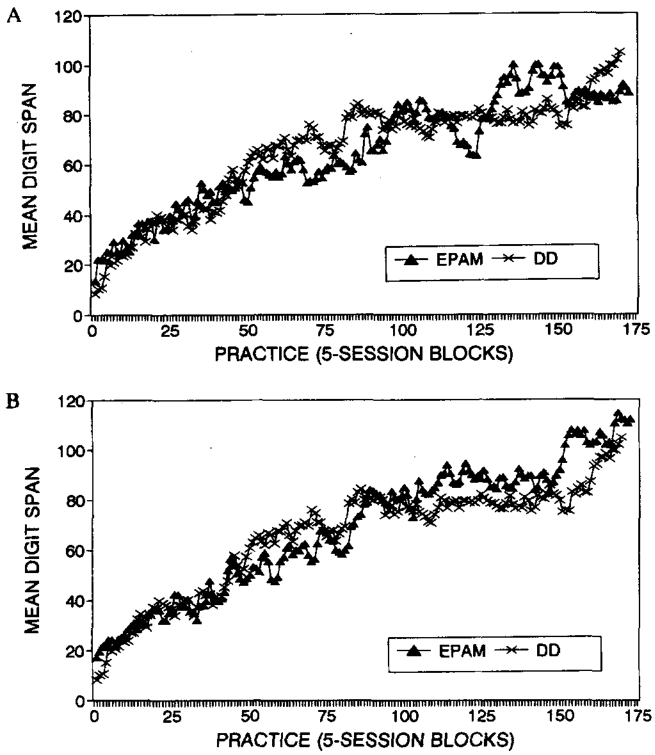

Figure 4. Learning curves for DD and two runs of Elementary Perceiver and Memorizer, Model IV (EPAM IV). (See text for further discussion.)

trials. Then, we discuss the assumptions that were made about forgetting during the course of the trials.

Presimulation 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 (0, 1, 2, 3, . . .) into their spoken forms as represented by words ("oh," "one," "two," "three," . . .). 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 preexisting 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.

Table 1
Number of Lists Presented to DD and EPAM Each Session

Sessions	Lists per session
1–15	25
16–20	15
21–25	10
26–40	8
41–220	6
221–265	5
266–475	4
476–865	3

Note. EPAM = Elementary Perceiver and Memorizer, Model IV.

Two of the categories defined by Staszewski (1990) have been split in two to simplify computations. Two-mile times were split into fast 2-mile times (below 10 min) and slow 2-mile times. Ages were split into two- and three-digit ages (such as 29 or 29.2) and four-digit ages (such as 8969), the latter called "double ages" by DD because he treated them as pairs of two-digit ages. 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, and they were of the kinds that most people would notice in digit sequences (e.g., that 3663 is "frontwards-backwards"). Thus, they represent basic human capabilities for recognizing various kinds of symmetries in perceptual inputs. Again, these productions are initial conditions of the simulation, constraining it rather than adding to its degrees of freedom.

Results: Learning During Recall Trials

Before looking at the details of their learning, we will summarize the performance of EPAM in comparison with that of DD over the entire run of 865 sessions. EPAM was run twice through this whole task, each time being given a different, randomly generated sequence of digits to recall. To simplify the comparison of EPAM's with DD's results, 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 not to claim that EPAM simulates DD at an almost microscopic level, but to provide a feeling for how the learning processes of both EPAM and DD proceed and to reveal the kinds of real or apparent fluctuations that can occur in such processes. We also speculate a

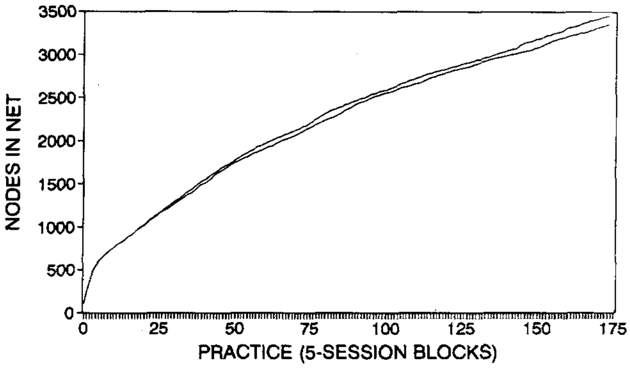

Figure 5. Growth of Elementary Perceiver and Memorizer, Model IV's (EPAM's) discrimination net over two runs.

bit on the mechanisms that might account for fluctuations—without ruling out the possibility that they are wholly random.

As shown in Figures 4A and 4B, in both runs of EPAM its digit span is interlaced 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.

Figure 5 shows the growth of EPAM's discrimination net over the course of its two runs, represented on the same chart. The two curves are so similar as to be almost indistinguishable. As will be explained in our description of the "rehearsal stage" of the serial recall task, the learning that the curves record occurs as soon as an entire list of digits has been presented to EPAM. 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. We have no evidence, one way or the other, whether DD's learning is focused on digit groups at the ends of his lists. Furthermore, 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 at the beginning of the first session 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. As is explained in our description of the rehearsal stage of the serial recall task, this learning occurs as soon as the presentation of a list of digits to EPAM has been completed. EPAM studies the list and attempts to learn new chunks (i.e., to add to semantic memory the last group of three to five digits) 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. At about block 145 (i.e., near the end of the plateau in DD's learning), his protocols began to include more frequent reports of using pattern codes during list encoding. (DD had originally been told by SF that taking time to search for pattern codes would hurt performance.) It may be that DD's eventual climb beyond a mean digit span of 80 was at least partly a result of using pattern codes more frequently than he had used them previously. EPAM used its pattern codes to the same extent on all trials.

2. As DD is an experienced runner, he began the simulations with a sizable semantic memory for running times. Perhaps he had 500 or more in memory. 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 to handle longer lists successfully. This reorganization provides a second possible explanation for the resumption of learning after the plateau. (Of course each of the proposed explanations may tell part of the story.) We come back to these points later, when discussing the data on response pause latencies.

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). In 16 blocks on Figure 4a their mean digit spans are equal (blocks 5, 6, 12, 17, 18, 25, 42, 46, 50, 55, 56, 57, 58, 72, 86, and 87). In Figure 4B the two curves overlap in a similar manner. 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 at which 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; on a set of 30 lists of 100+ digits each, for example, EPAM erred on 3.0% of the four-digit groups but on only 1.5% of the 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 finding 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 and hence can be set to match DD's overall learning rate without contradicting other data. 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 (including information from the pattern codes) 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 report below closely similar rates (23.6%) for DD's forgetting of patterns, so that the rate we used is not arbitrary. What is even more compelling is that the mechanisms of EPAM, and especially those that provided redundancy, were sufficient to account for the gradual acquisition of the extraordinary final memory performance, using memory parameters that have some prima facie plausibility.

In the Appendix we present an analysis developed by our colleague, Shmuel Ur, relating the probabilities of forgetting individual items with performance on the task. The analysis also relates the total forgetting rate to the component forgetting rates for digit groups in the retrieval structure, pattern code information, and information in the semantic net. 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 one 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.

This illustration of the importance of redundancy explains why 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

In broad outline, the serial recall task we have been discussing 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 preexisting pattern for a retrieval structure in memory or constructs and memorizes a new pattern. As explained earlier, because the structures for lists of different lengths are highly stereotyped, with their components already familiar, and because the progression to lists of new lengths is slow, this process does not consume much time.

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, that is to the cell of the structure to which the symbol in the special RSF cell points. Now the system is ready for presentation of the list.

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. The digits are not grouped in any way by the experimenter when they are presented to DD, but DD imposes his own grouping on them.

In all of our simulations, EPAM follows the same routine for each group of digits. The digits arrive at the auditory sensory store 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 & 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 to perform a recognition search through the discrimination net, and (e) 250 ms per node to perform an associative search through the net. The first two parameters fall well within the 1 s per digit presentation rate, leaving time for other learning activities. Parameter (a) is about half of a simple reaction time (no external reaction is called for). 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). If we assume that characters are the basic chunks in transcription typing (given that each character must be produced) and that words are the basic chunks in reading, then 250 ms can be interpreted as the basic time for accessing well-learned nodes in memory. (For further discussion of parameters in this time range for simple processes, see, for example, chapter 5 of Newell, 1990.) Modest changes in these parameters have little or no effect on performance, so that EPAM's performance is not sensitive to their exact values, but would be to mis-estimates of an order of magnitude. We would claim that the parameter values we have used are well within this limit of error.

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 (if there is no more than one digit that has not yet been attended to in the auditory loop), check the auditory loop to find whether there were back-to-back digits.
3. Digits are back to back when the first digit of one group is the same as the last digit of the previous group. If back-to-back digits are found, EPAM puts that information on the image of the RSF node. (It takes 100 ms to check for back-to-back digits. When such digits are found, the time required to attach that 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, add blanks at the end of the list to mark the places of future digits, delete an initial 0 if there is one, 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; if the RSF node will hold four digits, EPAM sorts the list (4, 2, 4, _). [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 6. The list (4, 2, 4) sorts to a node whose category is "1-mile time" (i.e., 4 min, 24 s), the list (4, 6, 4, _) sorts to a node with the category "10-mile time" (i.e., 46 min, 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 and 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 (as described in Step 1), 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 (i.e., quarter-mile time, then half-mile time, then three-quarter-mile time), and descending progressions (i.e., three-quarter-mile time, then half-mile time, then quarter-mile time). (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 be-

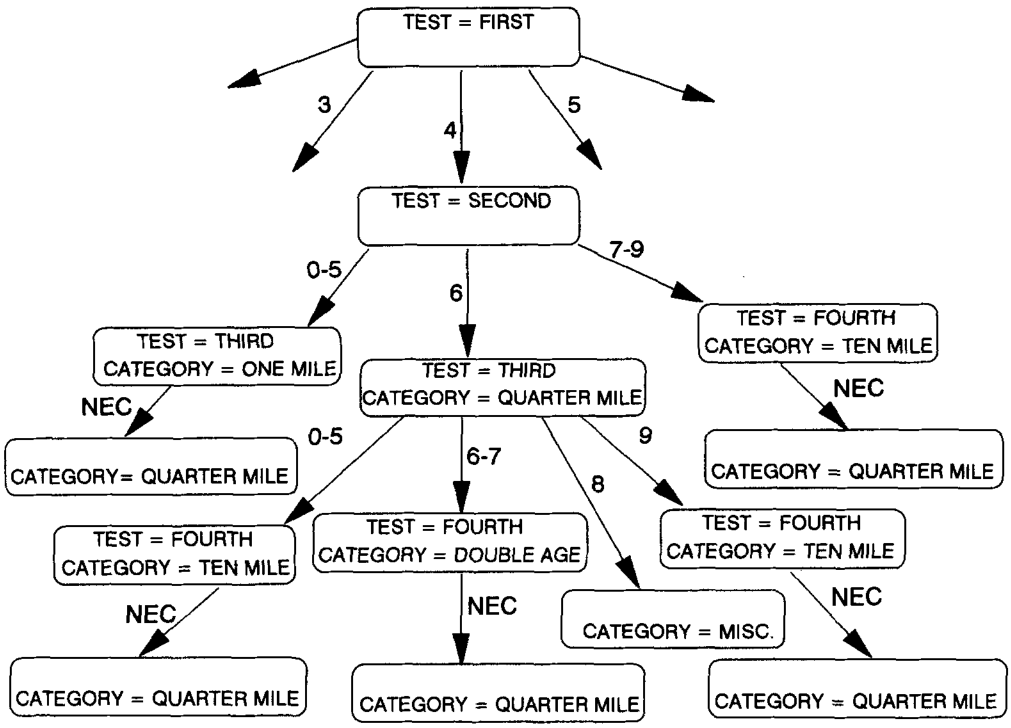

Figure 6. Discrimination net for semantic categories. NEC = not elsewhere classified; MISC. = miscellaneous.

fore 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 time. The chunk sorted to may hold an identical image (5, 6, 9) or a similar image such as (5, 6, 8). (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 on the image of the semantic node that was found in Step 10. This is the only part of the serial recall strategy that adds information to LTM, rather than to the retrieval structure. [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

In the time between list presentation and serial recall, DD rehearses. He is quite animated during this stage, in contrast with his passivity during the previous stage. DD reports that during rehearsal he is reviewing the semantic category of each digit group, moving backward through the groups. He also reports that he tries not to spend too much time with any digit group and that 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, which means that the image of a node in the discrimination net includes all of the digits in the group. EPAM also associates the fully familiar chunk with the RSF node so that there will be a pointer to the RSF node on the image of the chunk. This is where EPAM learns the one new chunk per list that increases its discrimination net. The time required is often extensive (about 8–10 s if a new node is built in the discrimination net). The process of rehearsing a stimulus while learning it 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, focusing on each node and then the previous one. 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 on 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. The difference is that, 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 cannot find a digit, he sometimes goes back to the previous group and tries to see whether it may provide some clues about its successor, for example, pattern cues that relate the two groups. If he cannot 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. Sometimes the process requires extensive time-consuming associative searches of the discrimination net at the rate of 250 ms per node traversed during the search. Traversal of the discrimination net and the retrieval structure during depth-first searches accounts for most of the time spent in this stage, as well as in the rehearsal stage and the optional free-recall stage. 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 to help determine the semantic category for the given RSF 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 on the information about digits

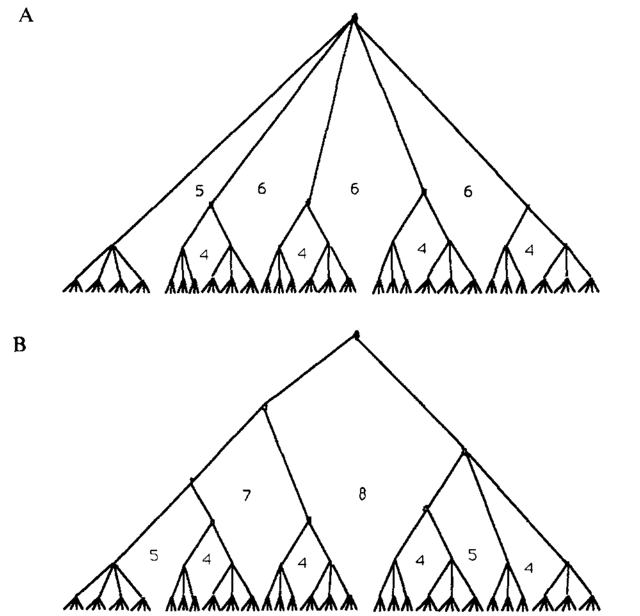

Figure 7. Original (A) and revised (B) retrieval structures. (See text for further discussion.)

at the RSF node's image. Guess what the missing digits could be, then, based on that guess, choose a likely semantic category. Then search the subnet of the discrimination net for that category in hopes of finding an active semantic node that points to the RSF node. (A parameter limits the amount of time that EPAM will spend searching for the semantic node of a digit group. These simulations were all conducted with a cutoff of 1 min on the simulated clock—if the system could not find the category after searching for 1 min, it would give up on that digit group.)

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. (The digits in the retrieval structure serve as a partial image to guide the search.) 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 EPAM's pause times between digits for the same lists. We have 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 7A). The second estimate is based on a slightly modified retrieval structure derived directly from DD's pause times (Figure 7B). A rationale for the second estimate, related to the yearlong plateau in DD's learning, is provided later.

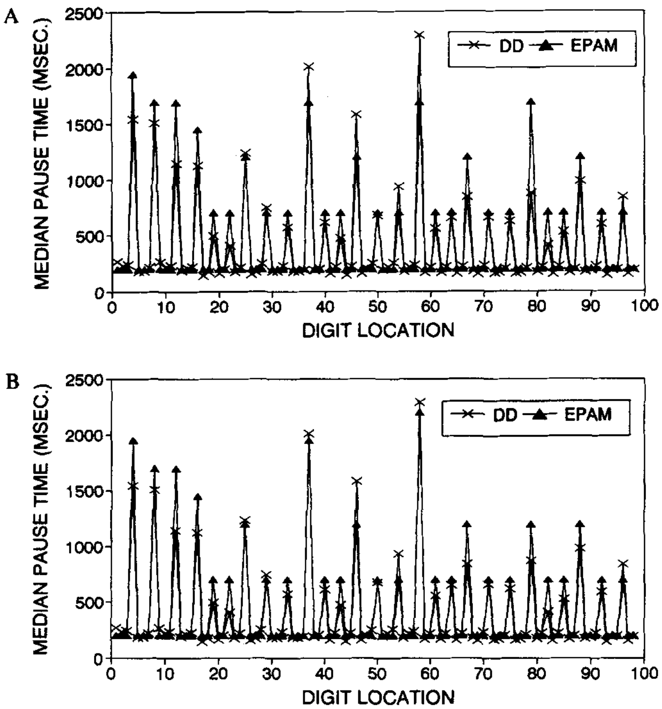

Figure 8. Comparison of predicted with actual pauses in serial recall for retrieval structures of Figures 7A and 7B. (See text for further discussion. EPAM = Elementary Perceiver and Memorizer, Model IV.)

EPAM tends to pause at about the same places for about the same median amounts of time as DD, as shown in Figures 8A and 8B. The fit in Figure 8A, based on the retrieval structure in Figure 7A, is poorer than that in Figure 8B, based on the structure in Figure 7B. The Pearson coefficient of correlation between predicted and actual pause times is .932 for Figure 8A and .964 for Figure 8B.

To match DD's data, EPAM's pause time within group boundaries is set at 200 ms, which is the approximate time a person requires to vocalize a digit. (This implies that the retrieval structure addresses are used to recover the groups of three or four digits, not the individual digits.) For the pause times between digit groups (except for the first three spikes, at digit locations 4, 8, and 12), the 200 ms required to vocalize a digit is added to the 250 ms required to traverse each node in the retrieval structure when changing the retrieval structure focus.

Occasionally, EPAM must pause for much longer periods at these junctures while it searches the discrimination net for information about the digits or semantic category that are missing from its retrieval structure images. However, these occasions occur rarely, as EPAM does most of its searching for these digit groups during the rehearsal stage. 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 8A are based on 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 8A are quite variable, ranging in our simulations from 950 to 2,700 ms each. 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 in this region of the retrieval structure. To these times are added the time required to search the discrimination net for information that is missing from the retrieval structure. For both DD and EPAM, without rehearsal the information stored in the retrieval structure is often incomplete and searches in semantic memory are needed to supplement it.

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 performed this task after he had achieved a digit span of over 100. EPAM also performed the task when it had achieved a digit span of 100. The main results of the experiment are shown in Table 2. (The results reported for EPAM are the average of 12 runs through all of the lists.)

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. DD takes an average of 115.6 s for the depleted lists and 72.4 s for the enriched lists, whereas EPAM takes an average of 102.6 s for the depleted lists and 86.3 s for the enriched lists.

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. These codes permit EPAM to reconstruct information that was lost due to forgetting in its retrieval structure, allowing it to avoid some searches and shorten other searches.

As we have already noted in our description of EPAM's serial recall process, the system notes 10 different types of numerical patterns during the study stage (see the description of the study stage for more detail). It notices five patterns

Table 2
DD's and EPAM's Mean Serial Recall Performance Scores (in Seconds) as a Function of List Type

	Enriched lists		Depleted lists	
Measure	DD	EPAM	DD	EPAM
Percentage correct	99.8	99.4	98.8	99.6
Rehearsal time	29.1	39.5	48.9	41.2
Recall time	43.3	46.8	66.7	61.4
Total time	72.4	86.3	115.6	102.6

Note. EPAM= Elementary Perceiver and Memorizer, Model IV.

in semantic categories (i.e., back-to-back categories, identical categories, symmetric categories, ascending progressions, and descending progressions), two types of patterns in which identical digits are found at the end of one digit group and at the beginning of the next group (back-to-back digits and back-to-back digit pairs), and three types of patterns within a group of digits ("frontwards-backwards," "differences," and "add-em-ups"). 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 (although we probably also incorporated a few that he does not use consistently).

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. Similarly, if a very similar mile time appeared in the identical location of the retrieval structure in the previous list, DD will note that fact as well as 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 for information that is attached to the retrieval structure.

At the end of the presentation of a list, DD gave verbal protocols of his semantic interpretations and pattern codes. We compared EPAM's noticing and reporting of pattern codes with DD's protocols. 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., 75*23* then *32*91), 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. In such cases we would count the larger unit as being forgotten if it was not reported, but we would not count the smaller unit as forgotten if the larger unit were reported.

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-em-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. For example, he often notices when the same codes appear in sequence; when 232 follows 545 he notes that the two groups are "back-to-back frontwards-backwards." 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. This effect could be labeled "proactive inhibition" as the study and recall of earlier lists apparently resulted in longer times to study and recall later lists. DD's results and EPAM's results appear in Figure 9 and Table 3. (Each point in DD's graph is the median of four data points, and each point in EPAM's graph is the median of 48 data points given that EPAM was run 12 times with these data.)

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 × trial order + 53 s (r^2 = .974)
2. DD's total time = 7 × 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, whereas it takes DD about 7 s longer.

Table 3
DD and EPAM's Median Total Recall Times (in Seconds) by Trial Order for Enriched and Depleted 30-Digit Lists

Trial order	Enriched lists		Depleted lists		Both lists combined	
	DD	EPAM	DD	EPAM	DD	EPAM
1	53.5	57.8	67.5	67.1	61.0	60.8
2	55.5	62.9	102.5	71.8	61.5	63.3
3	55.0	69.1	113.0	69.7	96.5	69.3
4	54.5	69.9	102.0	83.3	78.0	74.6
5	73.5	79.6	109.0	92.9	77.5	84.7
6	77.0	77.9	112.0	97.2	103.5	88.5

Note. EPAM = Elementary Perceiver and Memorizer, Model IV.

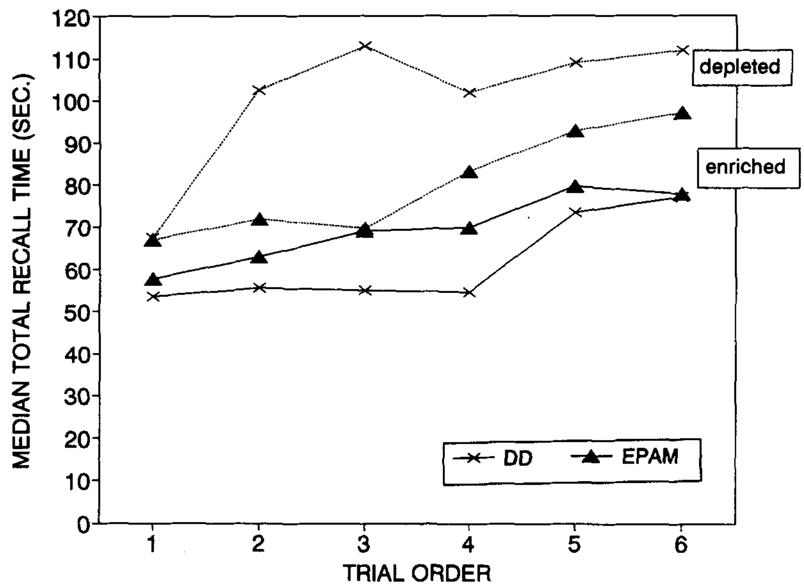

Figure 9. Effects of contextual coding on serial recall times. (EPAM = Elementary Perceiver and Memorizer, Model IV.)

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 (a portion of which is pictured in Figure 6) 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. With an earlier program, Staszewski (1990) was also able to obtain a 97% match with DD's categorizations of the same data.

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.

We did not simulate the visual presentation of the digit groups, but we did simulate EPAM's processes after it had put the digit group in a form suitable for sorting in its semantic net. EPAM's strategy was the following:

1. Orient to task and turn digit group into a list (estimated time = 1,500 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 on 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).

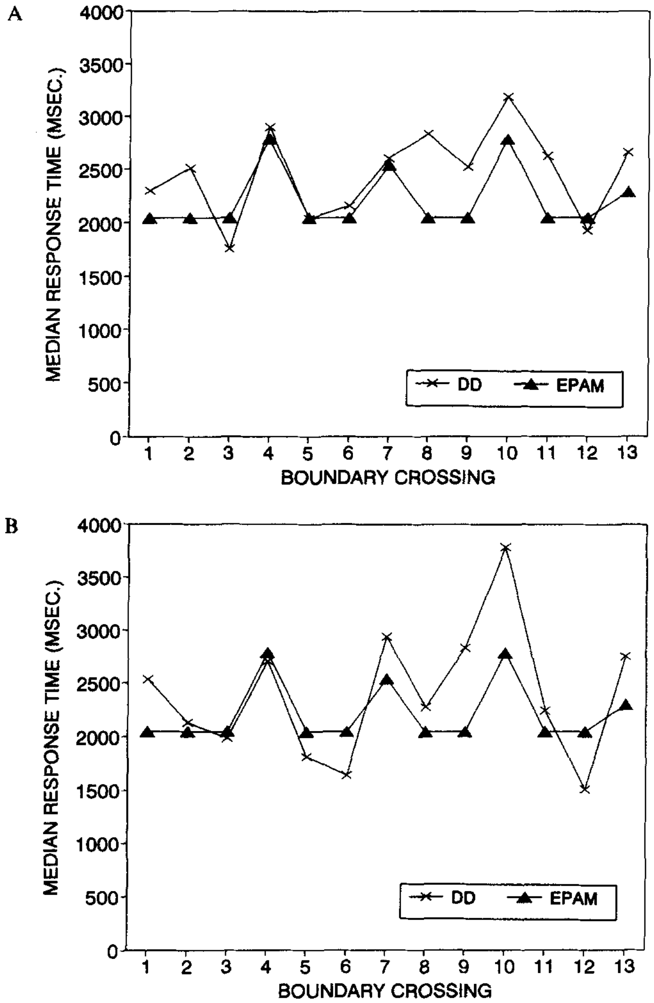

Figure 10. Median response times in memory scanning task for forward (A) and backward (B) directions. (See text for further discussion. EPAM = Elementary Perceiver and Memorizer, Model IV.)

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 graphed in Figure 10A and 10B 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. (A 10th session was discarded from his analysis because DD erred on the recall of one of the lists.) 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. (With a slightly more generous, but quite defensible, scoring criterion, the measure of fit rises close to 100%.)

To recall the digit groups, EPAM simply searches the semantic nodes of its discrimination net beginning in turn with the top nodes of each category. 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, whereas 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

Table 4
Results of Free Recall for Sessions of Three 100+ Digit Lists for EPAM Compared With DD

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

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 5
DD's and EPAM's Mean Performance Times (in Seconds) on 100+ Digit Lists

Measure	DD	EPAM
Rehearsal time	190	91
Ordered-recall time	200	118
Free-recall time	504	388

Note. EPAM = Elementary Perceiver and Memorizer, Model IV.

Spearman rank-order correlation ranged from .782–.985 (see Table 4). Four of the nine correlations were above .980 (.981, .984, .985, and .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 can do this because it is able to associate two retrieval structure nodes to the same semantic node.

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

As Table 5 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 on the mechanisms modeled in EPAM IV's new components (principally the retrieval structure) and on the mechanisms already present in previous versions of EPAM (principally the discrimination net and its associated semantic information).

Conclusion

The simulations that we have reported here provide a broad outline of mechanisms that are used by DD. There are many details that remain to be filled in, but EPAM's current behavior already 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 can 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. This aspect of learning does not explain why DD plateaus for well over a year at a maximum digit span of 80. It may be that DD has trouble at this level because of difficulties with trying to build a retrieval structure with more than four branches at a node or that he did not at first use pattern codes, or both reasons. 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 provided by retrieval structures, semantic codes, and pattern codes.

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 depth-first searches to become more extensive and hence 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 (in either the forward or backward direction).

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 as composed largely of depth-first searches, requiring 250 ms per node searched. 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 Ebbinghaus provides us with a good precedent, it is somewhat unconventional to build a theory of memory processes on the behavior of a single individual (or two, when we include SF). However, by following this strategy 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, but a judicious selection in each instance of the one likely to provide the deepest insights in a given experimental situation.

This study is also limited to behavior in a single (and rather esoteric) task: retaining and repeating back long digit strings that were presented at a rapid rate. To establish generality for the results, similar studies will have to be undertaken for a range of tasks in which this kind of special proficiency is exhibited by experts. 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). But though the range of tasks studied in detail is very limited to date, 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 on clear and explicit empirical evidence: the evidence provided by DD's performance and his verbal reports of his memory structures and strategies. With its rich memory structures and contents, through which previous experience interacts with current stimuli to determine behavior, EPAM IV is highly responsive to context, and its behavior is strongly "situated." Using its semantic memory, its retrieval structures, and its task-dependent strategies, 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. There is nothing ambiguous about the behavior it predicts, because the predictions can be tested in detail by running the computer and comparing its output with human behavior. We have taken pains to show that 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 tested against extended stretches of human behavior; as we have seen, it 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. We are not aware that any other theory has provided an explanation of the kind of expert performance exhibited by DD, which makes a direct comparison impossible in this case.

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. Until that is done, full contact cannot be made with current lines of experimentation that use these paradigms or with the models, some symbolic, others algebraic and stochastic, that have been used to explain the findings (Estes, 1991, 1994).

Even more 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. But all of that is work for the future.

References

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

Atkinson, R. C., & Shiffrin, R. (1968). Human memory: A proposed system and its control processes. In K. W. Spence & J. R. Spence (Eds.), *The psychology of learning and motivation: Advances in research and theory* (Vol. 2, pp. 89–195). New York: Academic Press.

Baddeley, A. D. (1981). The concept of working memory: A view of its current state and probable future development. *Cognition, 10,* 17–23.

Bellezza, F., Six, L. S., & Phillips, D. S. (1992). A mnemonic for remembering long digit strings. *Bulletin of the Psychonomic Society, 30,* 271–274.

Broadbent, D. A. (1975). The magical number seven after fifteen years. In A. Kennedy & A. Wilkes (Eds.), *Studies in long-term memory* (pp. 3–18). New York: Wiley.

Bugelski, B. R. (1962). Presentation time, total time, and mediation in paired-associate learning. *Journal of Experimental Psychology, 63,* 409–489.

Charness, N. (1976). Memory for chess positions: Resistance to interference. *Journal of Experimental Psychology: Human Learning and Memory, 2,* 641–653.

Chase, W. G., & Ericsson, K. A. (1981). Skilled memory. In J. R. Anderson (Ed.), *Cognitive skills and their acquisition* (pp. 141–190). Hillsdale, NJ: Erlbaum.

Chase, W. G., & Ericsson, K. A. (1982). Skill and working memory. In G. H. Bower (Ed.), *The psychology of learning and motivation* (Vol. 16, pp. 141–190). New York: Academic Press.

Darwin, C. J., Turvey, M. T., & Crowder, R. G. (1972). An auditory analogue of the Sperling partial report procedure: Evidence for brief auditory storage. *Cognitive Psychology, 3,* 225–267.

Dirlam, D. K. (1972). Most efficient chunk sizes. *Cognitive Psychology, 3,* 355–359.

Ericsson, K. A., Chase, W. G., & Faloon, S. (1980). Acquisition of a memory skill. *Science, 208,* 1181–1182.

Ericsson, K. A., & Oliver, W. (1984). *Skilled memory in blindfold chess.* Paper presented at the Annual Psychonomic Society Meeting, San Antonio, TX.

Ericsson, K. A., & Polson, P. (1988a). An experimental analysis of the mechanisms of a memory skill. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 14,* 305–316.

Ericsson, K. A., & Polson, P. (1988b). Memory for restaurant orders. In M. T. H. Chi, R. Glaser, & M. J. Farr (Eds.), *The nature of expertise* (pp. 23–70). Hillsdale, NJ: Erlbaum.

Ericsson, K. A., & Staszewski, J. (1989). Skilled memory and expertise: Mechanisms of exceptional performance. In D. Klahr & K. Kotovsky (Eds.), *Complex information processing* (pp. 235–268). Hillsdale, NJ: Erlbaum.

Estes, W. K. (1991). Cognitive architectures from the standpoint of an experimental psychologist. *Annual Review of Psychology, 42,* 1–28.

Estes, W. K. (1994). *Classification and cognition.* New York: Oxford University Press.

Feigenbaum, E. A., & Simon, H. A. (1984). EPAM-like models of recognition and learning. *Cognitive Science, 8,* 305–336.

Gilmartin, K. J., Newell, A., & Simon, H. A. (1976). A program modeling short-term memory under strategy control. In C. M. Cofer (Ed.), *The structure of human memory* (pp. 15–30). San Francisco: Freeman.

Gobet, F. (1993). A computer model of chess memory. *Proceedings of the 15th Annual Meeting of the Cognitive Science Society,* 463–468.

Gregg, L. W., & Simon, H. A. (1967). An information-processing explanation of one-trial and incremental learning. *Journal of Verbal Learning and Verbal Behavior, 6,* 780–787.

Hintzman, D. L. (1968). Explorations with a discrimination net model for paired-associate learning. *Journal of Mathematical Psychology, 5,* 123–162.

Luria, A. R. (1968). *The mind of a mnemonist.* Chicago: Henry Regnery.

MacGregor, J. N. (1987). Short-term memory capacity: Limitation or optimization? *Psychological Review, 94,* 107–108.

Mandler, G. (1967). Organization and memory. In K. W. Spence & J. T. Spence (Eds.), *The psychology of learning and motivation: Advances in research and theory* (pp. 328–372). New York: Academic Press.

Mandler, G. (1975). Memory storage and retrieval: Some limits on the

reach of attention and consciousness. In P. M. A. Rabbit & S. Dornic (Eds.), *Attention and performance* (Vol. 4, pp. 499–516). New York: Academic Press.
McLean, R. S., & Gregg, L. W. (1967). Effect of induced chunking on temporal aspects of serial recitation. *Journal of Experimental Psychology, 74,* 455–459.
Miller, G. A. (1956). The magical number seven, plus or minus two. *Psychological Review, 63,* 81–97.
Mitchell, J. M. (1939). Mnemonics. In *The encyclopedia Britannica* (14th ed., Vol. 15, pp. 623–625). Chicago: Encyclopedia Britannica.
Newell, A. (1990). *Unified theories of cognition.* Cambridge, MA: Harvard University Press.
Newell, A., & Simon, H. A. (1972). *Human problem solving.* Englewood Cliffs, NJ: Prentice Hall.
Payne, D. G., & Wenger, M. J. (1992). *Skilled memory development using word lists under conditions of limited practice.* Paper presented at the Fourth Annual Convention of the American Psychological Society.
Quillian, M. R. (1968). Semantic memory. In M. Minsky (Ed.), *Semantic information processing* (pp. 227–270). Cambridge, MA: MIT Press.
Richman, H. B., & Simon, H. A. (1989). Context effects in letter perception: Comparison of two theories. *Psychological Review, 96,* 417–432.
Simon, H. A. (1974). How big is a chunk. *Science, 183,* 482–488.
Simon, H. A. (1976). The information storage system called "human memory." In M. R. Rosenzweig & E. L. Bennett (Eds.), *Neural mechanisms of learning and memory* (pp. 79–96). Cambridge, MA: MIT Press.
Simon, H. A., & Feigenbaum, E. A. (1964). An information-processing theory of some effects of similarity, familiarization, and meaningfulness in verbal learning. *Journal of Verbal Learning and Verbal Behavior, 3,* 385–396.
Sperling, G. (1960). The information available in brief visual presentations. *Psychological Monographs, 74* (Whole No. 498).
Staszewski, J. (1988a). The psychological reality of retrieval structures: An investigation of expert knowledge (Doctoral dissertation, Cornell University, 1987). *Dissertation Abstracts International, 48,* 2126B.
Staszewski, J. (1988b). Skilled memory and expert mental calculation. In M. T. H. Chi, R. Glaser, & M. J. Farr (Eds.), *The nature of expertise* (pp. 71–128). Hillsdale, NJ: Erlbaum.
Staszewski, J. (1990). Exceptional memory: The influence of practice, and knowledge on the development of elaborative encoding strategies. In W. Schneider & F. E. Weinert (Eds.), *Interactions among aptitudes, strategies, and knowledge in cognitive performance* (pp. 252–285). New York: Springer-Verlag.
Staszewski, J. (1993). A theory of skilled memory. *Proceedings of the 15th Annual Meeting of the Cognitive Science Society,* 971–975.
Thompson, C. P., Cowan, T. M., & Frieman, J. (1993). *Memory search by a memorist.* Hillsdale, NJ: Erlbaum.
Tulving, E., & Thomson, D. M. (1973). Encoding specificity and retrieval processes in episodic memory. *Psychological Review, 80,* 350–373.
Von Neumann, J. (1956). Probabilistic logics and the synthesis of reliable organisms from unreliable components. In C. E. Shannon & J. McCarthy (Eds.), *Automata studies* (pp. 43–98). Princeton, NJ: Princeton University Press.
Wicklegren, W. A. (1964). Size of rehearsal group and short-term memory. *Journal of Experimental Psychology, 68,* 413–419.
Woodworth, R. S. (1938). *Experimental psychology.* New York: Holt, Rinehart.
Yates, F. A. (1966). *The art of memory.* Chicago: University of Chicago Press.
Zhang, G., & Simon, H. A. (1985). STM capacity for Chinese words and idioms: Chunking and the acoustical loop hypothesis. *Memory & Cognition, 13,* 193–201.

Appendix
Effects of Redundancy of Information on Retention

This appendix provides a somewhat more detailed account than is given in the body of the article of the probabilities associated with recalling long lists without error and the role of redundancy of information in keeping this probability at the level observed in DD's behavior.

In the digit span experiment, the length of the list presented to DD was increased by one digit each time he recalled a list correctly and decreased by one each time he made an error. This means that, with his observed moderate learning rate, he recalled entire lists correctly a little more than 50% of the time and made errors a little less than 50% of the time. With lists of 100 digits each, and assuming equal and independent probabilities of recall for each digit, to achieve a recall probability of .50 for the list would require a recall probability for each digit of .993 ($.993^{100} = .50$).

However, we know that DD did not memorize his lists as strings of individual digits, but as strings of groups of digits, with a group size of three or four. Assuming that a string of 100 digits consists of about 30 groups, to achieve a recall probability for the list of .50 would require a recall probability for each group of .977 ($.977^{30} = .50$).

If .007 is a wholly unrealistic forgetting rate, then .023 is only a little more plausible. In fact, we have estimated (with a little corroboration from evidence in DD's protocols) that DD forgets about 25% of the information he is placing in the retrieval structure as he hears the digits read.

However, we must take into consideration that DD has two additional sources of information besides the digits stored in the retrieval structure: namely, the semantic code (running times and ages) and the pattern code. Of course, even apart from forgetting, the semantic code or the pattern code will not always provide complete information about a digit group. DD's semantic net is incomplete—many digit groups are not represented in it—but it gets larger as the trials continue. We estimate that by the time the experiment ended DD had 3,000 digit groups in the semantic net, enough to represent substantially all three-digit groups but only a fraction of the 10,000 possible four-digit groups. The pattern codes, too, are incomplete; many groups do not have special features that are represented by pattern codes.

When we add a probable forgetting rate of 25% for digits and pattern codes in the retrieval structure and for the associative links from the semantic codes to the retrieval structure, matters become much worse. No single one of the information sources is nearly sufficient to account for the high retention rate of .977 per group that is required for the average 50% performance on long sequences.

The fallibility of the individual information sources is compensated by strong effects of redundancy—by the relatively low probability that all the sources will fail simultaneously. Let us create a rough model of the situation to calculate the effect of redundancy. Under reasonable circumstances, DD could reconstruct a three-digit group if information were retained in at least two, or perhaps three, of the information sources. For example, information about two of the three digits might be stored in the retrieval structure. This would provide enough information to retrieve the semantic code, where the remaining information could be found. Or the missing information might be recovered if one or two digits and a pattern code had been stored.

If we count the three digits in the retrieval structure (four for four-digit groups), the semantic code, and the pattern code as constituting, collectively, five (six for four-digit groups) independent but fallible sources of information, and assign the same probability of loss to each of them, we can calculate the probability that at least k pieces of information will be retained for each digit group, where k is the number of pieces that must be available to guarantee recall. If this probability is in excess of .977, then the entire list can be recalled with a probability of at least .5.

Table A1 displays these probabilities for $k = 1,5$ pieces of information retained for three-digit groups and $k = 1,6$ for four-digit groups. To show how this table is to be interpreted, let us consider how its entries are calculated in a simple case, with a forgetting rate of 30% and only one piece of information required for reliable recall of the group ($k = 1$). The probability that no one of the five pieces of information will be retained is .3 raised to the fifth power, or .00243. Hence the probability of recalling this group will be .9976, as shown in the third column of the last row of Table A1.

DD divides the 100-digit lists into about 30 groups, some of three and some of four digits. From Table A1, we see that the retention rate is above .977 (except for 30% forgetting in the three-digit groups), provided that only one or two pieces of information are required to recall each group, and is not much below this critical value even if three pieces of information are required.

From Table A1 it can be seen that, because there is more opportunity for redundancy of information with four-digit than with three-digit groups, the former have the larger probability of recall. However, this will only be the case if semantic codes are stored for the same fraction of both sets of groups. For three-digit groups, a complete set of codes requires a semantic net of only 1,000 nodes, but for four-digit groups, 10,000 would be required. The fact that DD and EPAM, in the course of their learning, acquired nearly complete nets for three-digit groups but much less complete nets for four-digit groups provides a possible explanation of why both make fewer errors with the three-digit than with the four-digit groups.

This model of redundancy only approximates EPAM's performance. It does not take into account the activation mechanism, the mechanism for searching semantic memory, the fact that different pieces of data contain more or less information (e.g., the semantic code is more complete than a single digit in the retrieval structure), and interlist interference. Nevertheless, it shows the relation of redundancy to the reliability of recall and provides estimates of the general magnitudes of the probabilities involved.

Table A1
Probability of Retaining Digit Group as a Function of Information Retention and Group Size

Minimum no. of digits retained	Group size 3			Group size 4		
	20%	25%	30%	20%	25%	30%
6				.262	.178	.117
5	.328	.237	.168	.655	.534	.420
4	.737	.633	.528	.901	.830	.744
3	.942	.897	.837	.983	.962	.929
2	.993	.984	.969	.998	.995	.984
1	.9997	.999	.9976	.9999	.9998	.9993

Received June 7, 1994
Revision received November 2, 1994
Accepted November 7, 1994 ■