COGNITIVE PSYCHOLOGY **30,** 221–256 (1996)
ARTICLE NO. 0007

Working Memory: Activation Limitations on Retrieval*

JOHN R. ANDERSON, LYNNE M. REDER, AND CHRISTIAN LEBIERE

Carnegie Mellon University

Two experiments which require subjects to hold a digit span while solving an equation and then recall the digit span are performed. The size of the memory span and the complexity of the equation are manipulated as well as whether the subject is required to substitute items from the digit span for constants in the equation. As either task (digit span recall or equation solving) gets more complex there are performance decrements (accuracy or latency) not only in that task but also in the other task. It is also shown that the majority of the errors are misretrievals. These results are consistent with the proposal that working memory load has its impact on retrieval from memory. These results are fit by the ACT-R theory (Anderson, 1993) which assumes that there is a limit on source activation and that this activation has to be divided between the two tasks. As either task increases in complexity there is less activation for retrieval of information from declarative memory. Subjects' misretrievals of associatively related information could be predicted by assuming a partial matching process in ACT-R.

As Baddeley (1992) notes there are several senses in which the term *working memory* has been used. The paper will be concerned with two of these senses. One is associated with the tradition that defines working memory in terms of paradigms which require the subject to maintain a memory load while performing a task (e.g., Baddeley & Hitch, 1974; Daneman & Carpenter, 1980). The second is associated with production system theories (e.g., Newell, 1991) where working memory is taken to be the currently available information against which production rules match. We are interested in relating these two senses because the ACT theory (Anderson, 1976, 1983, 1993) is associated with both. The ACT theory is associated with the first because of its strong roots in the human memory literature. It is associated with the second because it is a production system theory. ACT is a bit peculiar as a production system theory in that it does not have a working memory as that term is usually understood in production systems. Rather, the concept of capacity limitations is carried by the concept of activation. Elements in declarative memory have activation levels associated with them and access to these elements is a function of their level of activation. Roughly, working memory

This research is supported by Grants ONR N00014-90-J-1489 and NSF 91-08524 to the first author and NSF 89-08030 to the second author. We thank Marsha Lovett for her comments on a draft of the paper. Address correspondence and reprint requests to Dr. John R. Anderson, Department of Psychology, Carnegie Mellon University, Pittsburgh, PA 15213. E-mail: ja@cmu.edu.
* Accepted under the editorship of Douglas L. Medin.

0010-0285/96 $18.00

can be equated with the portion of declarative memory above a threshold of activation. Up until now we have never explored in detail how experimental manipulations of working-memory load would impact ACT's activation-based performance.

We have speculated about the role of working-memory load in studies of the effects of task complexity on skilled performance (Anderson & Jeffries, 1985; Anderson, Reder, & Ritter, in preparation). We have found that certain errors occur more frequently in the presence of greater complexity. For instance, Anderson et al. examined the frequency of errors while solving algebraic equations that varied in complexity. Consider the pair of errors below:

$$x + 6 = 9 \rightarrow x = 9 + 6$$

$$x + 6/5 = 9/4 \rightarrow x = 9/4 + 6/5.$$

Both errors reflect the failure to change the sign when taking a constant across an equation. This error occurred more frequently in the second case involving fractions even though logically the fractions are irrelevant to this transformation. We argued that the extra working-memory load associated with representing the fractions increased the rate of errors. However, we did not actually provide a mechanism for producing this error. This paper will provide a mechanism.

The example above is a case where the load in dealing with one aspect of a task (fractions) impacts on performance of another aspect of the same task. It is more typical to do experiments where working-memory load is manipulated in a separate task often requiring the subject to maintain some sort of span (e.g., Baddeley & Hitch, 1974). There are a number of current theories about how working-memory load impacts performance of such tasks. Baddeley (1986) has argued for a number of separate working memories (phonological loop, spatio-visual sketchpad) and that maintaining a span will impact target performance only if the span is so large that it will overflow into a central executive. Just and Carpenter (1992) argue that there is a certain amount of activation for performing a task which permits only so many things to be done. Their loading tasks are more cognitive in nature, but they too argue for separate spatial and linguistic capacities (Miyake, Shah, Carpenter, & Just, 1994). Our concept of working-memory limitations will be more continuous than either of these with task performance gradually degrading as load increases. Also, we will specifically localize the effect of load on retrieval from long-term memory.

THE ACT-R THEORY

ACT-R (Anderson, 1993) is a model of human cognition which assumes that a production system operates on a declarative memory. It is a successor to previous ACT production-system models (Anderson, 1976, 1983) and continues the emphasis on activation-based processes as the mechanism for relat-

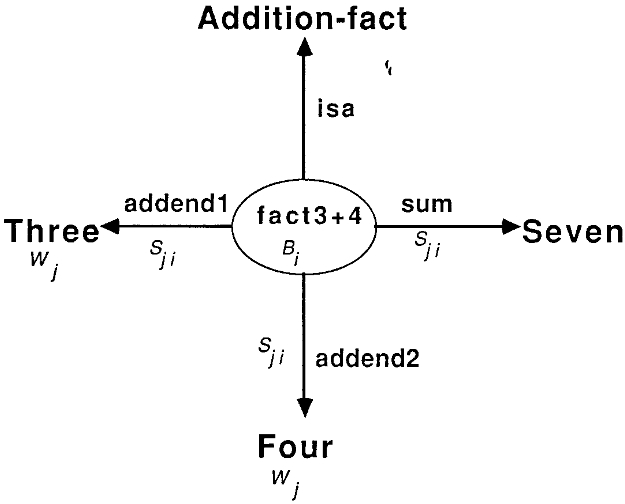

FIG. 1. A network representation of an ACT-R chunk.

ing the production system to the declarative memory. Different traces in declarative memory have different levels of activation which determine their rate and probability of being processed by the production rules. ACT-R is distinguished from the prior ACT theories in that the details of its design have been strongly guided by the rational analysis of Anderson (1989). Essentially, ACT-R is a production system tuned to perform optimally given the statistical structure of the environment.

According to the ACT theories, knowledge is divided into declarative knowledge and procedural knowledge. In ACT-R, declarative knowledge is represented in terms of **chunks** (Miller, 1956; Servan-Schreiber, 1991) which are schema-like structures, consisting of an *isa* slot specifying their category and some number of additional slots encoding their contents. Figure 1 is a graphical display of a chunk encoding the addition fact that $3 + 4 = 7$.

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

$$\begin{array}{r} 531 \\ +\ 248 \\ \hline 9 \end{array}$$

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

PROCESS-COLUMN

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

The first clause in this production matches the current goal to process the tens column; the second clause matches the digits in the tens column; and the third clause matches a fact or chunk from long-term memory. According to the ACT-R theory, an important component of the time for this production to apply will be the time to retrieve the long-term memories required to match the production rule. So, in this case where 3 and 4 are in the current column, the time to match the last clause will be determined by the level of activation of the chunk encoding 3 + 4 = 7 in Fig. 1. We explain how activation determines match time in the next subsection.

Activation

Activation of declarative structures has always been an important concept in the ACT theories. Basically activation determines how available information will be.[1] The activation of a chunk is the sum of source activation it receives from the elements currently in the focus of attention. Formally, the equation in ACT-R for the activation of element i is

$$A_i = \sum_j W_j S_{ji}, \qquad (1)$$

where W_j is the salience or source activation of element j in the focus of attention, and S_{ji} is the strength of association from element j to i.[2] For instance, in the context of retrieving the chunk that 3 + 4 = 7 in response to seeing 3 and 4 in a column, the W_j's would be the source activations of the elements 3 and 4 in the column and the S_{ji} would be the strengths of association from these elements to the chunk encoding 3 + 4 = 7. Figure 1 illustrates these quantities in the network encoding of the chunk. It is assumed in ACT-R, in contrast to early versions of ACT (such as in Anderson, 1976) but as in ACT* (Anderson, 1983), that these activations levels are achieved rapidly and that time to ''spread'' activation is not a significant contributor to latency. However, unlike ACT* there is no multilink spread of activation. Rather, activation is simply a direct response to source elements like j. As such, the theory is much like the SAM model (Raaijmakers & Shiffrin, 1981;

[1] According to the ACT-R theory the activation of a chunk reflects a preliminary estimate of how likely it is to match to a production at the current point in time. More precisely, activation reflects the log odds that the chunk will match to a production.

[2] The ACT-R theory allows each chunk to have a base level activation but we will simply assume that is zero.

Gillund & Shiffrin, 1984) except that our activations are like logarithms of SAM familiarities since they add rather than multiply. It will prove important to keep conceptually separate the quantities A_i and W_j. The former are activations, which control retrieval from declarative memory, while the latter reflect the salience or attention given to the cues.[3] The W_j's are referred to as *source activations.*

The levels of activation determine the odds that a chunk will be retrieved and the time to perform that retrieval. These measures are described by equations of the form

$$\text{Odds}_i = Ce^{cA_i} \tag{2}$$

$$\text{Time}_i = Be^{-bA_i}, \tag{3}$$

where A_i is the level of activation of the chunk i, and C, c, B, and b are constants mapping A_i onto the two performance measures.[4] The underlying model is one in which chunks are retrieved as candidates to match a chunk pattern in a production until one is matched (producing a latency) or until a give-up time is reached (producing an error). The exponential functions in Eqs. (2) and (3) allow for the kind of nonlinear mapping of activation onto behavior required in many activation models (e.g., McClelland & Rumelhart, 1986; Rumelhart & McClelland, 1986). For a justification of the exponential assumption in ACT-R, see Anderson (1993).

Working-Memory Limitation

It remains to specify a theory of working-memory limitation that can be related to manipulations of task complexity. In the context of the current theory, the natural assumption is that there is some limitation on total source activation. Formally, this limitation is

$$\sum_j W_j = \text{Constant}. \tag{4}$$

This reflects a limitation on the amount of attention one can distribute over source objects. This is a new assumption, not specified in Anderson (1993). This paper will explore how well we can account for working-memory phenomena by making this assumption.

This resource limitation has some similarity to the ideas introduced by Kahneman (1973) and has quite a bit of similarity to the Just and Carpenter (1992) CAPS theory which interprets working-memory limitation as a limitation on the total amount of activation available in a production-system archi-

[3] The W_j can be interpreted as measures of the validity of using that cue as a predictor of what will match to a production condition.

[4] The full ACT-R theory allows for a modulating effect of production strength but this is being ignored for sake of simplicity.

tecture. However, there are differences with the CAPS theory. Activation in the CAPS theory spreads by production firings rather than associations directly from sources to memory structures. Also the ACT-R limitation is not directly a limitation on activation but rather on the sources of activation. The total activation (A_i's in Eq. (1)) is a function of the strengths S_{ji} as well as the W_j. Finally and most important, our capacity limitation impacts retrieval from long-term memory.

Summary

It is worth reviewing the significant claims of this analysis of working-memory limitation:

1. The fundamental limitation is on amount of source activation (Eq. (4)).
2. This will impact on the activation of individual memory chunks (Eq. (1)).
3. This in turn will impact on probability and speed of successful retrieval (Eqs. (2) and (3)).

The unique aspect of this analysis of working-memory limitation is its localization of the limitation as impacting retrieval from declarative memory. We report research consistent with this localization. However, we do not mean to imply that there might not be other capacity limitations such as the rehearsal limitations in Baddeley's (1986) theory.

EFFECTS OF WORKING-MEMORY LOAD

One of the implications of the proposed extension (Eq. (4)) to the ACT-R theory is that there is a limited resource which is source activation. This would imply that two competing tasks, each of which required some source activation, would interfere with one another. This has been explored in experiments which require subjects to maintain a memory span concurrently while performing a primary task. Baddeley and Hitch (1974) found an interaction between memory span and complexity of the primary task (a reasoning task) such that there was a greater effect of primary task complexity at higher memory spans. Halford, Bain, and Maybery (1984) report such an interaction, both for performance of the primary task (an algebra-like task) and recall of the memory span. However, such interactions have not always been found (e.g., Evans & Brooks, 1981; Klapp, Marshburn, & Lester, 1983).

Carlson, Sullivan, and Schneider (1989) reported an experiment relevant to the issue of what determines whether there is a working memory interaction between a primary task and a concurrent memory load. We designed our experiments after their paradigm. During part of their experiment they presented their subjects with a memory span of three or six elements. The memory span involved the presentation of assignments of binary values to variables. In the three-span case subjects might be given $A = 1$, $B = 0$, $C = 1$. While holding this memory span subjects were required to predict the output for a logic gate given a particular set of input values. Then they were probed for their memory of the span by being presented with a question of the form $A = 0$ which they had to judge as correct or false. A critical manipulation in this experiment involved the relationship between the memory span and the judgment of the logic gate. In the **irrelevant** condition there was no relationship. In the other two conditions subjects knew they might need the information in the memory set to judge the gates. In the **access** condition, rather than seeing binary input to the gates subjects saw two variables and had to retrieve the values of these variables and predict what the gate would do for these values. In the **expect** condition, subjects thought they might see variables but in fact saw 0's and 1's as input.

Carlson et al. (1989) found little effect of size of memory span on irrelevant or expect trials but a large effect on access trials. The effect of three versus six memory load was 35 ms in the nonaccess conditions and 296 ms in the access condition. Also subjects were about 800 ms slower overall in the access condition. We were intrigued with this task for a number of reasons. First, Anderson (1989) argued that the large effect of access occurred because information in the memory span had to be used in the logic-gate task. Increased memory span would lower the activation of the individual elements in the memory span (as a fan effect) which would impact on the rate with which they could be used in the logic task. Thus, in effect, Anderson's argument was that there were separate working-memory limitations in the digit span and logic-gate tasks and the only way to get an effect of memory span was to integrate the memory span into the logic-gate task. This contrasts with Eq. (4) which proposes a single resource limitation.

We were also interested in the Carlson et al. (1989) manipulation because we thought this was a good way to explore the effects of working-memory load on a problem-solving task like algebraic equation solving. As we noted, rather weak effects of memory span have often been found on primary tasks (e.g., Klapp et al., 1983). The access condition of Carlson et al. seemed like a paradigm that was much more sensitive to effects of memory load. Carlson et al. did not report data that indicated whether complexity of the primary task interacted with the memory span task. The ACT* theory predicts such an interaction and one reason for the current experiment was to test this prediction.

While these were the motivations for choosing the paradigm, we should say at the outset that the results turned out to be rather different than we anticipated. As such, they proved more relevant to assessing and developing the ACT-R theory than the ACT* theory.

EXPERIMENT 1

We adapted the Carlson et al. (1989) paradigm to a task involving algebra problem solving in which we manipulated the complexity of the algebra task, the size of the memory set, and whether there was a requirement to access

TABLE 1
Example Problems Used in Experiment 1

	No substitution	Substitution
One transformation	$3x = 6$	$ax = b$
	$x/3 = 6$	$x/a = b$
	$3 + x = 9$	$a + x = b$
	$3 - x = 9$	$a - x = b$
Two transformations	$3x + 2 = 8$	$ax + b = 11$
	$3x - 2 = 7$	$ax - 2 = b$
	$x/3 + 2 = 8$	$x/a + b = 11$
	$x/3 - 2 = 7$	$x/3 - a = b$

the memory set in performing the algebra task. Table 1 illustrates the 16 types of material that were used. Half of the problems required one algebraic transformation to solve and half required two transformations. There were four basic types of one-transformation equations, involving multiplication, division, addition, or subtraction from both sides. The two-transformation equations consist of one multiplication or division and one addition or subtraction, giving four possible combinations. In the no-substitution condition, integers appeared in the equations whereas in the substitution condition the letters *a* and *b* replaced two of the integers. In the one-transformation condition both integers were replaced while in the two-transformation condition a random pair of the three integers were replaced. In the substitution condition, the subject was to substitute the first two digits from the digit span for *a* and *b* in the equation. This is the condition that requires subjects to integrate the contents of their digit span into the algebra problem-solving task. Crossed with these 16 types of materials, subjects were responsible for digit spans of two, four, or six digits. Thus, there were always two digits that could be integrated into the problem.

Method

Materials and procedure. Equations were randomly generated subject to the constraints that the constants all be one digit, that the intermediate result in the two-transformation equations be one digit, and that the final answer be an integer. All constants in the equations were greater than zero; also, when a constant was explicitly given as a coefficient of x, it was greater than 1. There were 48 problems required to realize all combinations of the 16 basic equation types in Table 1 and the three memory set sizes. The experiment consisted of five blocks of trials. In each block all 48 conditions were realized. Thus, there were 240 trials in all. The presentation of problems was random within a block. After each block there was a possibility for a break.

The problems were generated first and then the memory spans were randomly generated by the computer from the integers 0–9 with the constraint that in the substitution condition the first two digits had to come from the problem. Individual problems were presented in a different random order for each subject.

The experiment was administered by an IBM PC. Each digit string appeared on the PC screen for a time determined by its length ($1.5 + 0.5n$ s, where n is the number of digits) after which

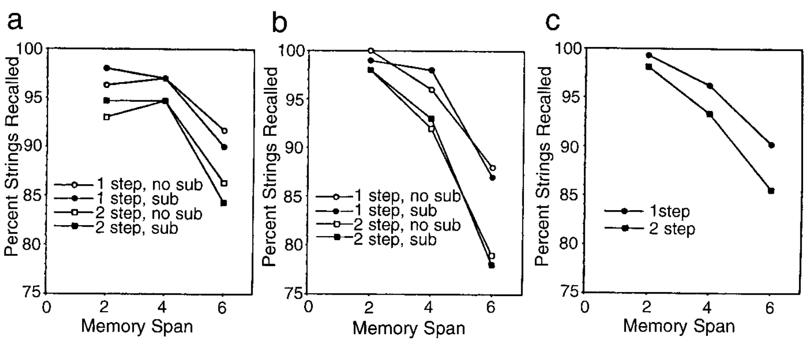

FIG. 2. Percentage of strings correctly recalled in Experiment 1: (a) Data; (b) Simulation; (c) Predictions of mathematical model.

it was replaced by an algebra problem. The subject solved the equation without benefit of paper and pressed a button when a solution had been reached, at which time the equation disappeared from the screen. Then the subject typed in the answer followed by the digit string. Immediate feedback was provided for both the answer and the digit string.

Subjects. The subjects were 15 CMU undergraduates, graduates, and staff who participated in the experiment either as part of a requirement of the introductory psychology course or for pay ($10). The experiment lasted about 2 h.

Results

The computer program recorded solution times, solution accuracy, and accuracy of string recall. Each of these dependent variables was subjected to a $2 \times 2 \times 3$ analysis of variance where the factors were equation complexity, whether substitution was necessary, and memory span.

Figure 2a displays the results for the percentage of memory spans perfectly recalled. There were significant effects of size of memory span ($F(2,28) = 6.02$, $p < .01$) and of equation complexity ($F(1,14) = 9.19$, $p < .01$). The effect of substitution was not significant ($F(1,14) = 0.77$), nor were there any significant interactions. The effect of two versus four digits was not significant ($t_{28} = 0.12$) but the effect of four versus six was highly significant ($t_{28} = 2.94$, $p < .01$). The difference between these two effects was marginally significant ($t_{28} = 1.70$, $p \sim .05$). Thus, it seems with respect to digit recall we need to account for the following facts:

1. There was a larger effect of four versus six digits in the memory span than two versus four.
2. There was a complexity effect of number of transformational steps.
3. There was no effect of substitution.

Figure 3a displays the results for percentage equations correctly solved. There were significant effects of equation complexity ($F(1,14) = 5.73$, $p <$

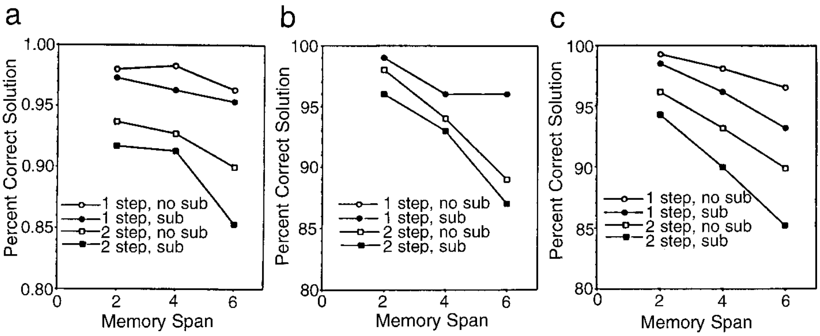

FIG. 3. Percentage of equations correctly solved in Experiment 1: (a) Data; (b) Simulation; (c) Predictions of mathematical model.

.05), memory span ($F(2,28) = 4.96$, $p < .05$), and substitution ($F(1,14) = 7.66$, $p < .05$). There were no significant interactions. The effect of two versus four digits was not significant ($t_{28} = 0.43$) but the difference between four and six digits was ($t_{28} = 2.59$, $p < .05$). The difference between the two effects was marginally significant ($t_{28} = 1.52$, $p < .10$). From the figure, it appears that the effect of complexity is larger than the effect of substitution and a contrast testing for this was significant ($t_{42} = 1.78$, $p < .05$). The only interaction that was even marginally significant was that between complexity and span ($F(2,28) = 2.26$, $p = .12$). The effect of span appears somewhat larger in the case of complex equations. A specific contrast testing whether the curves are steeper in the case of complex equations was significant ($t_{28} = 1.89$, $p < .05$). Thus, with respect to accuracy of equation solving, we need to account for the following facts:

4. There was an effect of memory span which may be greater for complex equations.
5. There was a complexity effect of number of transformational steps.
6. There was an effect of substitution but this effect was smaller than the complexity effect.

Figure 4a displays the results for time to solve the equations. There were significant effects of equation complexity ($F(1,14) = 41.71$, $p < .001$), memory span ($F(2,28) = 33.65$, $p < .001$), and substitution ($F(1,14) = 137.23$, $p < .001$). The effects of equation complexity and substitution are approximately equal. There was also a significant interaction between memory span and substitution ($F(2,28) = 4.00$, $p < .01$) such that the effect of memory span was greater in the case of substitution. This replicates the access effect found by Carlson et al. (1989). The increase with memory span was only marginally significant in the case of no substitution ($t_{70} = 1.49$, $p < .10$) but quite significant with substitution ($t_{70} = 5.93$; $p < .001$). The overall difference

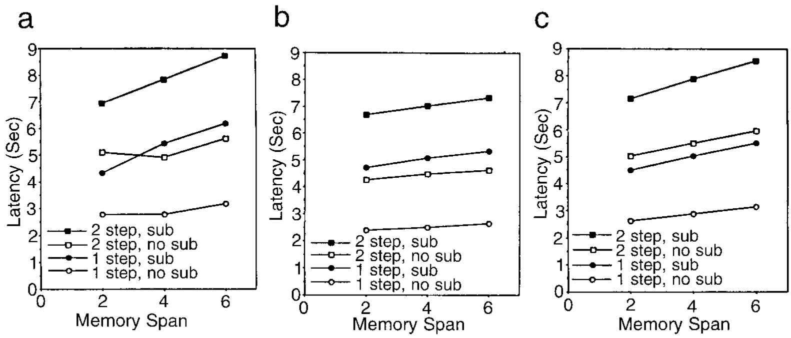

FIG. 4. Time to solve equations in Experiment 1: (a) Data; (b) Simulation; (c) Predictions of mathematical model.

between two and four items in the span was significant ($t_{28} = 3.31$, $p < .01$) as was the difference between four and six items ($t_{28} = 4.91$, $p < .001$). The difference between the sizes of the two effects was not significant ($t_{28} = 0.92$). Thus, with respect to latency of equation solution, we need to account for the following facts:

7. The effect of memory span was larger in the case of substitution.
8. There was a complexity effect of number of transformational steps.
9. There was an effect of substitution which is equal to the effect of number of transformational steps.

The equal effect of number of transformational steps and substitution in Fig. 4a allows us to rule out certain interpretations of two other aspects of the data. First, the interaction between memory span and substitution (point 7 above) is not simply a result of larger effects of memory span at longer latencies. The interaction is nicely displayed in these two overlapping curves. The one-step, substitution curve is significantly steeper than the two-step, no substitution curve ($t_{28} = 2.03$, $p < .05$) although both conditions have approximately the same latency. Second, the equal time effects of these variables affects our interpretation of the result that only equation complexity affected retrieval of the digit string (Fig. 2a). It means that the poorer recall of the memory span after solving two transformation equations could not be due to the greater elapsed time because substitution, which produced the same time increase, did not have an effect on span recall.

ACT-R Model

We developed an ACT-R simulation to account for equation solving and digit recall performance. The model for the equation solving involved productions to recognize the appropriate transformations to apply, solve the arithme-

TABLE 2
Productions Applying in the Solution of an Equation from Experiment 1

Initial state: $x/a - b = 4$

Substitute for *a*:
IF the goal is to solve an equation
and ''*a*'' appears in the equation
and *f* is the first element of the memory set
THEN substitute ''*f*'' for ''*a*''

Resulting state: $x/3 - b = 4$

Substitute for *b*:
IF the goal is to solve an equation
and ''*b*'' appears in the equation
and *s* is the second element of the memory set
THEN substitute ''*s*'' for ''*b*''

Resulting state: $x/3 - 2 = 4$

Invert-transformation:
IF the goal is to solve an equation of the form ''term op1 $c = d$''
where *c* and *d* are constants
and op2 inverts op1
THEN transform equation to the form ''term = *d* op2 *c*''

Resulting state: $x/3 = 4 + 2$

Collect-sum:
IF the goal is to solve an equation that contains ''$c + d$''
and *s* is the sum of *c* and *d*
THEN replace ''$c + d$'' by ''*s*''

Resulting state: $x/3 = 6$

Invert transformation:
IF the goal is to solve an equation of the form ''term op1 $c = d$''
where *c* and *d* are constants
and op2 inverts op1
THEN transform the equation to the form ''term = *d* op2 *c*''

Resulting State: $x = 3*6$

Collect-product:
IF the goal is solve an equation that contains ''$c*d$''
and *p* is the product of *c* and *d*
THEN replace ''$c*d$'' by ''*p*''

End State: $x = 18$

Type-out:
IF the goal is to solve an equation of the form ''$x = c$''
and *c* is a constant
THEN press the button and type *c*

tic relationships, substitute the correct values, and type out the answer. Consider the most complex case which involved solving equations of the form

$$x/a - b = 4$$

with $a = 3$ and $b = 2$. Table 2 gives the sequence of productions that applies in this case. Basically, there is one production for each operation in producing the answer. Our model for the recall of the memory span simply involved productions that encoded the incoming digits into successive serial positions and then retrieved from those positions.

Simulation

We ran a simulation of ACT-R which assumed fixed capacity for source activation which had to be divided between the terms in the equation and the terms in the digit span. This capacity was set at one unit.[5] We assumed that the capacity was equally divided among all of the symbols of the equation and the memory load. In the case above, there were seven symbols in the equation: x, /, a, -, b, =, 4. Thus the total number of elements was $7 + s$ where s is the size of the digit span. Therefore, in the case of a two-transformation equation, the total activation of any element was $1/(7 + s)$. In the case of a one-transformation equation, this element activation was $1/(5 + s)$ because there were only five symbols in the equation.

Figures 2b, 3b, and 4b show the predictions from 125 Monte Carlo simulation runs per condition for the data in Figs. 2–4 using the parameters $c = 8$, $b = 2$, $C = 1$, and $B = 1$ for Eqs. (2) and (3). The two one-step curves in Fig. 3b are, by chance, identical.[6] The time scale in Fig. 4 is arbitrary and could be changed by changing the constant B. These simulations were run assuming the only time spent was in the retrieval involved in matching conditions. There is no cost for the action sides of productions. These simulations reproduce some of the qualitative appearance of the data; however, it is difficult to assess the goodness of this fit or to understand why the model fits in some places and misfits in other places.

One of the sources of complexity in understanding the simulation is that the activation levels vary with the strengths S_{ji} among elements (see Eq. (1)) which are a function of the exact connectivity among elements (see Anderson, 1993, for details). Also, there is a random component such that the results are only Monte Carlo approximations to the pure ACT-R predictions. In order to generate more precise predictions of the simulation, we produced a

[5] Since c and A multiply in Eq. (2) and b and A multiply in Eq. (3), the exact setting of total source activation is not important since any change in it can be compensated by proportional changes in b and c.

[6] Since there are only 125 simulating runs, it is possible for conditions which are theoretically different to post the same percentage correct.

mathematical model of its application to this task and then optimized the fit of the mathematical model to the data. This is described below.

Mathematical Model

Retrievals from long-term memory are an important determiner of accuracy and latency in the ACT-R theory. Therefore, in developing a mathematical analysis of its predictions, it is important to identify which productions require retrieval from memory and how memory load impacts retrieval. Each production except *type-out* requires an long-term memory retrieval. That is, the *substitute* productions require retrieval of the digit from the memory span, the productions *collect-sum* and *collect-product* each require retrieval of an arithmetic fact, the production *invert-transformation* requires retrieval of an algebraic transformations such as ''+ *inverts* −'' or ''* *inverts* /'' which are used to undo the operation. Thus, in general, the number of retrievals for a problem is one less than the number of productions.

To determine the amount of activation arriving at a to-be-retrieved element i, we need to know the strengths of association, S_{ji}, in Eq. (1). It turned out in the simulation that if a memory span element was to be retrieved, there was only one source j of activation associated with that element. This was the span element itself and its strength of self-association was approximately 5. Thus, the source activation was multiplied by 5. In the case of retrieval of arithmetic facts like $3 + 4 = 7$ there were two sources of activation (e.g., 3 and 4) and their strength of association to the target fact was about 2.5. Thus, the source activation was effectively multiplied by $2.5 + 2.5 = 5$.[7] So, all elements had the source activation multiplied by about 5. In effect, since the total source activation was 1 and it was multiplied by 5, the net activation for any to-be-retrieved element was $5/(d + s)$ where d is the number of symbols in the equation and s is the span size. This is the value of A used in the equations below and allows us to avoid considering the S_{ji} values. According to the ACT-R theory, A should be related to time to retrieve by the equation

$$T_R = Be^{-bA}, \tag{5}$$

which is basically a repeat of the earlier Eq. (3), and to probability of retrieval by the equation

$$P_R = \frac{Ce^{cA}}{1 + Ce^{cA}}, \tag{6}$$

[7] In ACT-R, the default is to set the strength of self-association proportional to the log of the number of chunks in the database. With respect to an association from j to i the default is to set that strength to the difference between the log of the number of chunks and the log of the number of associations involving j. See Anderson (1993) for a discussion. The fact that the strength of self-association is equal to two j–i associations reflects the fact that the number of items associated to j turned out to be approximately a square root of the number of items in the database.

which is based on the earlier Eq. (2). The parameters B, C, b, and c were free to be estimated.

The time to complete a trial (solve the equation) is the sum of the amount of time associated with production firings, T_P, plus the component retrievals, T_R.

$$T(\text{Solution}) = mT_P + nT_R, \tag{7}$$

where m is the number of productions and n is the number of retrievals. In the example in Table 2, $m = 7$ and $n = 6$. Only the component T_R will be affected by memory span and activation. The component T_P reflects an ''average'' estimate of the time to do the other (non-long-term memory retrieval) matching of the production's condition and to execute the production action.

As for accuracy, we assumed that all errors are caused by failure to correctly retrieve information. Thus, the probability of a correct answer is the product of the probabilities of correct retrievals:

$$P(\text{Solution}) = P_R^n, \tag{8}$$

where n is the number of retrievals. P_R will be impacted by activation which will in turn be a function of equation complexity and memory span. This assumes that no errors were due to systematic bugs in the subjects' procedures. This seems a reasonable assumption given the high level of performance of all subjects on all equations.

Finally, we assumed memory span accuracy would simply reflect the accuracy of retrieval of the digits:

$$P(\text{String}) = P_R^S, \tag{9}$$

where S is the number of digits in the span. Note that it is the same P_R in Eqs. (8) and (9). Thus, the same basic effect is predicted for accuracy in equation solution as in digit span.

The model requires the estimation of five parameters: B, the scaling factor for retrieval latency; C, the scaling factor for retrieval odds; b, the exponent for latency; c, the exponent for accuracy; and T_P the non-retrieval time for a production. We fit 36 data points—12 for solution latency, 12 for solution accuracy, and 12 for string accuracy. We minimized a χ^2 statistic defined as

$$\sum_i (\text{Obs}_i - \text{Pred}_i)^2 / S_{Obs}^2,$$

where Obs_i are the observed means, Pred_i is the model's prediction, and S_{Obs}^2 is the variance of the means taken from the analysis of variance tables. S_{Obs}^2 is obtained from the overall interaction between subjects and conditions for that dependent measure. With 36 data points and five parameters, there were 31 degrees of freedom. The minimum χ^2 statistic obtained was 26.95 which indicated no significant deviations overall. The parameters estimated were $B = 1.88$ s, $C = 3.73$, $b = 3.16$, $c = 5.96$, $T_P = 0.74$ s.

Figure 2c displays the predicted accuracy of string recall. We have collapsed over the substitution factor because the model predicts no effect of substitution on string recall, replicating the third fact noted under Results. The model also predicts an effect of complexity of the appropriate size (fact 2). The mean effect of equation complexity in the data is 3.7% while it is 2.9% in Fig. 2c. The model also predicts the curvilinearity such that the effect of two digits versus four digits in the span is smaller than the effect of four versus six but the effect does not appear as dramatic as in Fig. 2a. Thus, the effect for memory span is partially reproduced by the model. It is worth noting why the model predicts increasing growth in error rate with larger span. As span increases two things happen—level of activation of the span elements goes down and more elements have to be recalled. To some degree these effects ''multiply'' in this error scale, producing the accelerated error rate. While the theory produces an accelerated error rate, it does not produce the apparently flat error rate from two to four digits. It is hard to know whether to attribute this deviation from prediction to chance (since it is not statistically significant). However, to the extent that it is real, we think it may reflect differential allocation of effort between the span and equation solution. A number of subjects report ''trying harder'' for the larger digit spans.

Figure 3c displays the predicted accuracy in equation solution. It reproduces all three main effects found in the data—effect of memory span (fact 4), effect of number of steps (fact 5), and the smaller effect of substitution (fact 6). The effect of substitution occurs because of potential errors in retrieval of the digits. The effect of number of steps occurs both because of potential errors in the extra memory retrievals (of an algebraic transformation and an arithmetic fact) and because of increased memory span (more symbols in equation). It also reproduces the greater effect of memory span for more complex equations because the two extra memory retrievals for complex equations will be impacted by the memory span.

Figure 4c displays the predicted latency in equation solution. It again reproduces all three effects found in the data. The effect of memory span is larger in the case of substitution (fact 7) because retrieval of the digits is slower for larger spans. However, the effect of substitution on the memory span effect is not as large as in the data. As for the data, in the case of no substitution, the difference between two and six items is 0.47 s while in the case of substitution it is 1.77 s. As for the predictions, the effects are 0.76 s and 1.19 s. There are approximately equal effects of number of steps (fact 8) and substitution (fact 9) and the size of the effects is very similar to the data.

All in all, the fit of the model is quite good in terms of significance of deviation with 36 data points defined on three dependent measures and in terms of accounting for the nine basic phenomena in the data. It is worth emphasizing that it achieves its success by localizing the capacity limitations in the retrieval of chunks from declarative memory to match production conditions. In this way it explains the effect of the substitution interaction with span which Carlson et al. (1989) found and we replicated. The effect of substitution occurs because it requires extra retrievals in solving the equation. However, we expect and find effects of span even when there are no substitutions. This is because other retrievals are required in equation solving. It is reasonable to conjecture that no retrievals other than substitution were required in the Carlson et al. task. Therefore, they found no effect of span in the no substitution condition.

Sensitivity Analyses

Given the good fit of the model it becomes interesting to inquire as to what aspects of the model are responsible for the fit and how the model would behave under alternative assumptions. There are four structural parameters of the model and five estimated parameters. The structural parameters are d (the number of digits in the span), s (the number of symbols in the equation), m (the number of productions), and n (the number of memory retrievals). While identifying d with the number of digits in the span is hard to question, there is room for questioning the other parameter assignments. Certainly, one could have produced production rule sets that involved different numbers, m, of productions. Since T_P is an estimated parameter in Eq. (7), the model is not sensitive to the exact number of rules, only the relative proportion of them in various conditions. In fact, we tried a fit where T_P was constrained to zero and got a χ^2 value of 28.72 which is only slightly larger than the unconstrained χ^2 of 26.95. Thus, the model fit is not particularly sensitive to any assumption about the number of productions. For similar reasons the model is also not sensitive to the exact value of n, the number of retrievals. However, here it is critical for the behavior of the model that there be twice as many retrievals in the case of complex equations as simple, which seems a reasonable assumption. Perhaps, the most substantial structured assumption was to identify s with the number of symbols in the equation. It seemed reasonable to us that subjects should have to process each symbol in the equation, but one might defend a model in which subjects had to process only the numbers which would mean $s = 2$ for simple equations and $s = 3$ for complex equations. We tried fitting this model and it did not fit as well ($\chi^2 = 32.80$). The reason for this is that the equation solving under these assumptions about s does not impose the same interference to the digit span. Therefore, in setting $s = 5$ for simple equations and $s = 7$ for complex equations we are establishing a certain critical trade-off between loads for the two tasks, i.e., requiring more of the activation to go to the equation representation and less to the digit span representation. This is not to deny, however, that some of our assumptions about symbol activation may only be approximate—for instance, that all symbols receive equal activation.

We also explored what would happen to the goodness of fit if the estimated parameters T_P, B, b, C, and c took on different values. What we did was to fix one of these parameters at 50% more or less than their best fitting values and then to estimate best fitting parameters under this constraint. Table 3

TABLE 3
Sensitivity Analysis of the Model Fits for Experiments 1 and 2

	Original model	T_p 50% smaller	T_p 50% larger	B 50% smaller	B 50% larger	b 50% smaller	b 50% larger	C 50% smaller	C 50% larger	c 50% smaller	c 50% larger	$b = c$
Experiment 1												
T_p	0.74	0.37	1.11	0.76	0.83	0.52	0.84	0.74	0.74	0.74	0.74	0.82
B	1.88	1.65	0.00	0.94	2.82	1.48	2.80	1.88	1.88	1.88	1.88	2.51
b	3.16	1.31	1.37	1.81	4.68	1.58	4.73	3.15	3.15	3.15	3.15	4.34
C	7.38	7.38	7.38	7.38	7.38	7.38	7.38	3.69	11.07	19.82	2.88	7.63
c	4.42	4.42	4.42	4.42	4.42	4.42	4.42	6.01	3.51	2.21	6.62	4.34
χ^2	25.05	26.12	40.56	27.28	25.21	25.57	25.22	27.19	25.91	30.73	28.9	25.16
Experiment 2												
T_p	0.48	0.72	0.24	1.06	1.02	0.00	0.96	0.48	0.48	0.48	0.48	1.13
B	5.82	6.07	5.65	2.91	8.73	4.47	7.27	5.82	5.82	5.82	5.82	8.19
b	2.71	3.14	2.34	1.42	4.72	1.36	4.06	2.71	2.71	2.71	2.71	4.70
C	3.70	3.70	3.70	3.70	3.70	3.70	3.70	1.85	5.55	8.93	1.63	4.56
c	5.36	5.36	5.36	5.36	5.36	5.36	5.36	7.54	4.15	2.68	8.04	4.70
χ^2	79.28	79.55	79.51	101.34	82.98	83.67	80.92	87.12	82.55	96.6	90.47	83.38

reports the results of these explorations. As can be seen, the quality of fit did not suffer much under these settings as compensating values could be estimated for the other parameters. The one exception was that the quality of fit decreased perceptibly when T_P was set to be 50% higher. The reason why these fits were so good generally is because B and b can trade off for latency and C and c can trade off for accuracy.[8] Larger values of the time scale parameter B can compensate for larger values of the exponent b and larger values of the odds scale parameter C can compensate for smaller values of the exponent c. It was apparent from this exploration that we could have a four-parameter version of this model in which the exponents are constrained to be the same ($b = c$) which is also reported in Table 3.

In summary, we think the model fits are sensitive to those aspects that were expected—the digit span (d), the symbolic complexity of the equation (s), and the relative number of memory retrievals (n) required for simple versus complex equations. With respect to the estimated parameters, the scale parameters B and C are being estimated to produce the average values observed of the latency and accuracy dependent measures. The exponents b and c are being estimated to produce the mapping of changes of activation onto changes in performance. Basically, the data are a function of how much the load produced by the combined tasks impacts the memory retrieval required in each task.

A Separate Capacity Model

We thought it would be informative to see how a model which assumed one capacity for digit span and a different capacity for equation solving would do at fitting these data. Thus, the activation available for doing the equation was $5/s$ and for the memory span $5/d$. Such a model without elaboration does poorly at fitting the data ($\chi^2 = 62.80$) since it fails to capture the task interactions. However, a reviewer pointed out to us that there was a fairly simple way to elaborate the model to produce some of the task interactions. There are two ways that the digit span might impact upon the equation solving despite the lack of shared capacity. First, in the case of substitutions, two retrievals from the span are required which will be impacted by the span activation. Second, it is possible that subjects were covertly rehearsing the span while solving the equation. (Although our subjects did not report doing this, research has shown it is rather difficult to assess implicit rehearsal; Reitman, 1974). More time would be taken away from equation solving for each digit that had to be rehearsed. Therefore, we estimated a mean time, r, for each second of equation solving that a subject would give to rehearsing a digit. Thus, if it took T s to solve the equation without a span and the span had d digits it would take $T*(1 + dr)$ s to solve the equation.

[8] Essentially what the exponent determines is how quickly changes in activation result in changes in time and accuracy, with larger values producing steeper functions while the scale parameters determine the average values of these functions.

Equation solving slowed the time before recall of the digits began and one might imagine that the activation of the digits decayed over this time. Assuming exponential decay, if it took T units to solve the equation, the activation would have decayed by an amount a^T where a is the fraction decayed each second. Thus, the parameter a becomes another parameter of the model.

We fit this seven parameter model to the data and achieved best fitting estimates of $r = 0.06$ s, $a = 0.95$, $T_P = 0.26$, $B = 0.84$ s, $b = 0.11$, $C = 14.16$, and $c = 1.83$. The χ^2 statistic was 36.14 which is about 10 larger than the original model which had two fewer degrees of freedom. This model has a major difficulty in explaining the effect of equation complexity on digit span. As we already noted with respect to Figs. 2 and 4, conditions with nearly identical latency in equation solving are showing considerable differences in digit accuracy depending on the complexity of the equation.

This effort is by no means definitive proof that some separate capacity model might not be capable of accommodating the data. A different framework might produce a different conclusion. A problem is that there is not a separate capacity model which is equivalently explicit in its predictions for this task as our version of ACT-R and we were left to transform our existing model into a separate capacity model.

The Predicted Three-Way Interaction

The theory does make predictions about other interactions with memory span which, while they tended to be in the right direction, were generally not significant in the data. As Figs. 3c and 4c illustrate, the effect of memory span is predicted to be stronger with either substitution or two-transformation equations. And indeed, there is a three-way interaction predicted such that the effect of span should be strongest in the case of two-transformation with substitution. Both substitution and complexity increase the span effect because substitutions or more complex equations require more retrievals, each of which will be impacted by the decreased activation with larger memory span. ACT-R predicts a three-way interaction because equation complexity lowers the source activation for performing the memory retrievals required by substitution. However, the complexity manipulation was rather weak, adding only two symbols to the equation. In Experiment 2 we decided to investigate a more substantial variation in equation complexity. We also used a greater variation in span size.

EXPERIMENT 2

The results from Experiment 1 were encouraging with respect to the ACT-R theory's analysis of the effects of memory load. We decided to see how it would apply to equations like those in the Anderson, Reder, and Ritter (in preparation) study where we found that the presence of fractions rather than additional transformations also increased the frequency of algebraic errors. This proves to be a much more substantial manipulation of the number of symbols in the equation. So, if our analysis localizing the effect of equation

TABLE 4
Example Problem Used in Experiment 2

	No substitution	Substitution
Simple	$3x = 7$	$ax = b$
	$3 + x = 7$	$a + x = b$
Complex	$-3/4x = 7/2$	$-(a/4)x = b/2$
	$3/4 + x = -7/2$	$a/4 + x = b/2$

complexity in number of symbols is correct, we should see larger interactions with equation complexity.

Table 4 illustrates the eight types of material that we used. All the problems were just one algebraic transformation removed from a solution. That transformation could involve either subtraction from both sides or division of both sides to isolate the variable. The arguments in the equation could be either simple positive integers or complex signed fractions. The fractional equations involve additional symbols for numerator, denominator, fraction bar, and number signs. Finally, the arguments could be numbers or the letters a and b. In the case of letters, the subject was to substitute the first two digits from the digit span for a and b in the equation. This is the condition that requires subjects to integrate the contents of their digit span into the algebra problem-solving task. On a given trial, subjects were responsible for remembering two, four, six, or eight digits. Digit span was crossed with the eight types of materials.

Method

Materials and procedure. Sixteen instances were created of each of the eight problem types illustrated in Table 4. Four instances were randomly assigned for each subject to each of the four lengths of digit string. Digit strings were randomly generated by the computer with the constraint that in the substitution condition the first two digits had to come from the problem. Individual problems were presented in a different random order for each subject.

The experiment was administered by an IBM PC. Each digit string appeared on the PC screen for a time determined by its length ($1.5 + 0.5n$ where n is the number of digits) after which it was replaced by an algebra problem. The subject signaled by pressing a button when a solution had been reached and then typed in the answer and the digit string. Immediate feedback was provided for both the solution and the digit string.

Subjects completed the experiment in two sessions, ranging in length from 40 to 75 min. Breaks were offered although most subjects preferred not to take them.

Subjects. The subjects were 20 CMU undergraduates and graduate students who participated in the experiment either to help satisfy a requirement of the introductory psychology course or for pay.

Results

The computer program recorded solution times, solution accuracy, and accuracy of string recall. Each of these dependent variables was subjected to

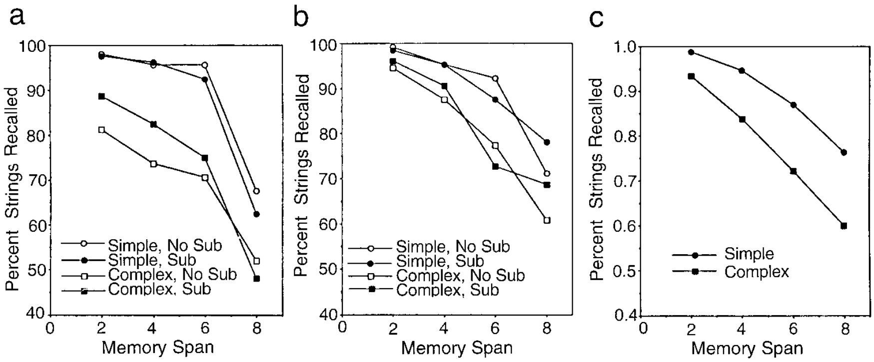

FIG. 5. Percentage of strings correctly recalled in Experiment 2: (a) Data; (b) Simulation; (c) Predictions of mathematical model.

a 2 × 2 × 4 analysis of variance where the factors were equation complexity, whether substitution was required, and memory span.

Figure 5a displays the results for percentage memory spans recalled. There were significant effects of size of memory span ($F(3,57) = 35.87$, $p < .001$) and of equation complexity ($F(1,19) = 46.75$, $p < .001$). The effect of substitution was not significant in this experiment as in Experiment 1 ($F(1,19) = 0.46$). However, there was a substitution by complexity interaction ($F(1,19) = 9.03$, $p < .01$) such that subjects were 2% less accurate when they performed substitution for simple equations and 4% more accurate when they performed substitution for complex equations. The substitution effect for complex equations is significant ($t_{19} = 2.4$, $p < .01$). There is also a significant span-by-substitution interaction ($F(3,57) = 2.87$, $p < .05$) such that the substitution advantage is mainly for small spans. There were no other significant interactions. There is again a curvilinear trend in the effect of span: The decrease from two to six digits is significant ($t_{57} = 2.21$, $p < .01$) but significantly less than the decrease from six to eight ($t_{57} = 3.53$, $p < .001$). Thus, with respect to digit recall we need to account for the following facts:

1. There is an advantage of the substitution condition for complex equations with short spans.
2. There is an effect of equation complexity.
3. There is a larger effect of six versus eight digits than two versus six.

Figure 6a displays the results for percentage of equations correctly solved. There were significant effects of equation complexity ($F(1,19) = 47.85$, $p < .001$) and substitution ($F(1,19) = 20.00$, $p < .001$). The effect of memory span was marginally significant ($F(3,57) = 2.56$, $p < .10$). A specific contrast for a linear trend was significant ($t_{57} = 2.53$, $p < .01$). Since Experiment 1 found an effect of memory span, it seemed likely that there would be one

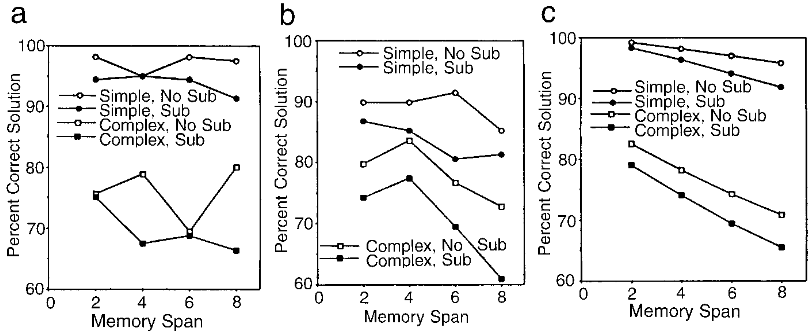

FIG. 6. Percentage of equations correctly solved in Experiment 2: (a) Data; (b) Simulation; (c) Predictions of mathematical model.

in this experiment too. This experiment also found a significant three-way interaction of complexity and substitution with memory span ($F(3,57) = 5.91$, $p < .005$). The data are definitely noisy but fitting linear functions reveals no effect of span in the case of simple, no substitution and a 1.24% increase in error rate per item in the digit span in the case of the complex, substitution. This is the predicted three-way interaction that failed to be significant in the first experiment. Thus, with respect to accuracy of equation solving we need to account for the following facts:

4. There is a weak effect of memory span which is largest in the case of complex equations with substitution.
5. There is an effect of equation complexity.
6. There is an effect of substitution but smaller than the effect of equation complexity.

Figure 7a displays the results for time to solve the equations. There were

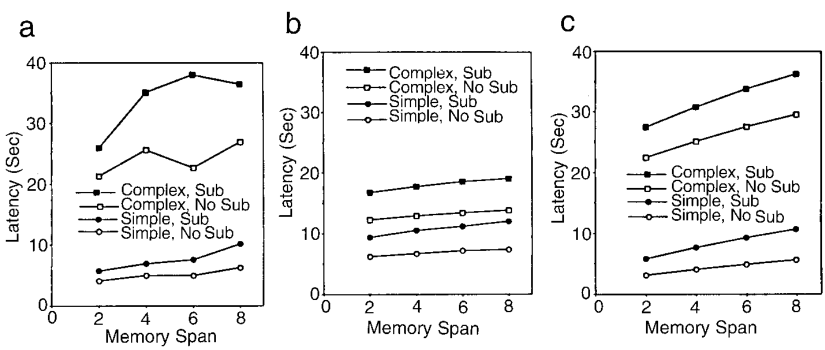

FIG. 7. Time to solve equations in Experiment 2: (a) Data; (b) Simulation; (c) Predictions of mathematical model.

significant effects of equation complexity ($F(1,19) = 4169.33$, $p < .001$), of memory span ($F(3,57) = 10.65$, $p < .001$), and of substitution ($F(1,14) = 110.27$, $p < .001$). There was also a significant interaction of memory span and substitution ($F(3,57) = 5.74$, $p < .01$) such that the effect of memory span is greater in the case of substitution. This replicates the interaction found by Carlson et al. (1989). The increase in latency with increased memory span was only marginally significant in the case of no substitution ($t_{57} = 1.61$, $p < .10$) but quite significant with substitution ($t_{57} = 3.16$, $p < .001$). The two experiments both found marginal effects of span in the case of no substitution. Combining the two experiments, the effect is significant ($z = 2.19$, $p < .01$). Thus, unlike Carlson et al. we conclude that there is an effect of span on latency in the absence of the substitution requirement. In this experiment all the other interactions were significant as well—complexity by substitution ($F(1,19) = 38.85$, $p < .001$); complexity by memory span ($F(3,57) = 6.55$, $p < .001$); and complexity by substitution by memory span ($F(3,57) = 4.89$, $p < .001$). This again is the predicted three-way interaction that failed to be significant in the previous experiment. In particular, the effect of memory span increases from simple, no substitution (slope = 0.33 s per item) to simple, substitution (0.70 s per item), or complex, no substitution (0.69 sec per item) to complex, substitution (1.88 s per item). It is true that the effect of span tends to be larger in conditions with longer latency; however, this effect cannot be simply an artifact of larger effects for conditions with larger base RTs: the complex, no-substitution condition has the same slope as the simple substitution condition, yet the former has a much higher base RT. Thus, with respect to solution time, we need to account for the following effects:

7. The effect of memory span is larger in the case of substitution or in the case of complex equations.
8. There is an effect of complexity.
9. There is an effect of substitution but it is smaller than the effect of complexity.

The nine effects reported above substantially correspond to the results from the first experiment. Equation complexity, manipulated by use of fractions, produced effects similar to the effects of equation complexity, manipulated by number of transformations. Complexity in this experiment, though, produced larger effects particularly on the time to solve equations where subjects took almost five times longer to solve the complex equations.

Finally, almost as an aside, we note that there may be something of a speed-accuracy trade-off in the condition of solving complex equations with no substitution. This is the condition that produced the greatest deviations from monotonicity in Figs. 6 and 7. What is striking about these data is that they mirror each other—every time there is a dip or rise in accuracy (Fig. 6a), there is a compensating dip or rise in latency (Fig. 7a).

TABLE 5

Productions Applying in the Solution of an Equation from Experiment 2

Initial state: $x + 3/4 = -7/6$

Invert-transformation:
IF the goal is to solve an equation of the form ''term op1 $c = d$''
where c and d are constants
and op2 inverts op1
THEN transform the equation to the form ''term $= d$ op2 c''

Resulting state: $x = -7/6 - 3/4$

Collect-two negatives:
IF the goal is to solve an equation that contains ''$-a-b$''
THEN replace ''$-a-b$'' by ''$-(a + b)$''

Resulting state: $x = -(7/6 + 3/4)$

Collect-sum-fraction:
IF the goal is to solve an equation that contains ''$a/b + c/d$''
THEN replace ''$a/b + c/d$'' by ''$(a*d + b*c) \div b*d$''

Resulting state: $x = -((7*4*3*6) \div 6*4)$

Collect-product:
IF the goal is to solve an equation that contains ''$c*d$''
and p is the product of c and d
THEN replace ''$c * d$'' by ''p''

Three applications of the above production produces

Resulting state: $x = -((28 + 18) \div 24)$

Collect-sum:
IF the goal is to solve an equation that contains ''$c + d$''
and s is the sum of c and d
THEN replace ''$c + d$'' by ''s''

Resulting state: $x = -(46 \div 24)$

Simplify-fraction
IF the goal is to solve an equation of the form ''$x =$ sign $(a \div b)$''
and $a = x*c$
and $b = x*d$
THEN encode this as ''$x =$ sign c/d''

End state: $x = -23/12$

Type-out:
IF the goal is to solve an equation of the form ''$x = c$''
and c is a constant
THEN press the button and type c

ACT-R Model

We extended the ACT-R model from Experiment 1 to account for these data. The same sequence of production rules was used for the simple equations but a different and more complex set of productions was used for the complex equations to do the fractional arithmetic. Table 5 illustrates the sequence of productions that would be required to solve a problem involving addition of fractions.

This sequence for addition of fractions in Table 5 involves nine productions and six memory retrievals of arithmetic facts and one retrieval of an algebraic fact (*invert-transformation*) yielding seven memory retrievals. The corresponding sequence for multiplication involves seven productions and five memory retrievals (ignoring substitutions). In contrast, for simple equations there are always three productions and two retrievals for both addition and multiplication as in the past experiment. This contrast implies subjects should take longer to solve addition problems than multiplication problems for the complex equations but not for the simple equations. Indeed, there was an interaction between complexity and type of operation ($F(1,19) = 38.85$, $p < .001$). In the case of simple equations there was no difference between addition and multiplication (6.25 versus 6.35 s) while addition took much longer in the case of complex equations (35.39 versus 23.29 s).

We ran the ACT-R simulation of this task 160 times per condition and the average data are illustrated in Figs. 5b, 6b, and 7b.[9] For the sake of brevity, we will proceed directly to describing the fit of the mathematical model of the application of the ACT-R theory to this task. In calculating activation sources, we assumed that each term in the equation plus each digit in the memory span was an element. Setting a bound on source activation of 1, as in Experiment 1, meant that each to-be-retrieved chunk had an activation of $5/(5 + s)$ in the case of simple equations and $5/(11 + s)$ in the case of complex equations where s was the number of digits in the memory span. Each fraction term required four symbols to represent the sign, the fraction bar, and the denominator as well as the numerator. This contrasted with the simple equations which required only one symbol per integer. Thus, with two fractions in a complex equation, there were six more symbols than in the simple equation. The other parameters of the model were the same as those in the previous experiment and were estimated to be $B = 5.82$ s, $C = 3.70$, $b = 2.72$, $c = 5.36$, and $T_P = .48$ s. With five estimated parameters and 48 data points, the χ^2 measure of goodness of fit had 43 degrees of freedom and had

[9] The simulation code can be found at http://sands.psy.cmu.edu/ by following the paths ''ACT-R architecture,'' ''ACT-R software,'' ''models,'' and ''algebra.'' It can also be obtained by ftping to ftp.andrew.cmu.edu and logging in as anonymous, in the directory /pub/act-r/ftp/models/algebra.

the value 79.3. This indicates a good fit but there is significant residual variance not predicted.

The fit of the mathematical model is shown in Figs. 5c, 6c, and 7c. Figure 5c displays the predicted accuracy of string recall. The major discrepancy involves fitting the effect of the memory span. While the model does produce accelerated drop off with span size, it does not capture the magnitude of the acceleration. For instance, the data show a drop of 26% from span 6 to span 8 while the model predicts only a 12% drop. It is also the case that the model fails to predict the interactions with substitution noted in the results section. If we ignore the span recall, the overall fit of the model to the equation solving data is adequate ($\chi^2_{31} = 34.33$, $p > .05$). We discuss memory span performance further in a later section.

Figure 6c displays the predicted accuracy in equation solution. It reproduces all three results found in the data—effect of memory span that varies with complexity and substitution (fact 4), large effect of complexity (fact 5), and the smaller effect of substitution (fact 6). Figure 7c displays the predicted latency in equation solution and should be compared with Fig. 7a. It also reproduces all three results found in the data. The effect of memory span is larger in the case of substitution or complexity (fact 7). The model reproduces this because increased span slows retrieval of the digits in the case of substitution and of the arithmetic facts in the case of fractional complexity. There is an effect of complexity (fact 8) and a smaller effect of substitution (fact 9).

By way of summary, except for the memory span data (Fig. 5) the ACT-R model seems to be capturing all the significant effects in the data. Even in the case of the span data, it is capturing much of the effects (the R^2 between observed and predicted indicates that the model is predicting 85.4% of the variance for even these data).[10] Again this supports the localization of working-memory limitations in memory retrieval.

Other Analyses

As in Experiment 1, we also fit the separate capacity model to these data. This model fit with a comparable χ^2 of 79.7 but it required two additional parameters. The best fitting values of the parameters were $r = 0.03$ s, $a = 0.94$, $T_P = 0.00$ s, $B = 5.34$ s, $b = 0.98$, $C = 10.11$, and $c = 2.16$. The reason that this model was relatively more successful in this experiment than in the previous experiment was that the simple and complex equations were sufficiently separated by time that one could use time-decay of activation to fit the effect of complexity on span recall. However, the first experiment did cast doubt on time as the correct explanation for this effect. We obtained substantial differences between simple and complex equations even when there was not a time difference.

[10] And 93.1% for equation accuracy and 96.8% for equation latency.

Also as we did in Experiment 1, we performed a sensitivity analysis of our model. This is reported in the lower half of Table 3. As can be seen, the model is again relatively insensitive to the actual parameters estimated, trading off T_P, b, and B for predicting time and trading off c and C for predicting accuracy. Again, the model fits reasonably well with the constraint that the exponents b and c be equal.

The actual values estimated for these parameters are quite different in the two experiments. However, the trade-offs that exist suggest that we could constrain these estimates to be the same between the two experiments with relatively little effect on overall goodness of fit. We were able to fit the data of both experiments with the same set of parameters except that we needed to estimate two values of B, the time scale parameter. To see why this is necessary, consider performance on the simple equations with spans of 2 through 6 which were the common conditions in both experiments. For these comparable conditions there is little difference in accuracy between the two experiments: subjects solved 97% of the equations in Experiment 1 and 96% in Experiment 2; they recalled 95% of the spans in Experiment 1 and 96% in Experiment 2. On the other hand, there was a large difference in latency with subjects taking 4.1 s to solve these equations in Experiment 1 but 5.7 s in Experiment 2. Perhaps because Experiment 2 required a greater variety of facts to be retrieved, the time to retrieve any one fact was lower due to the less frequent repetition. We fit the experiments allowing two time scale parameters B_1 and B_2. The best fitting parameters were $T_P = 0.73$ s, $B_1 = 1.92$ s, $B_2 = 6.01$ s, $b = 3.16$, $C = 2.87$, and $c = 6.06$. The χ^2 values for this 6 parameter model is 121.8 which compares with 106.7 for the full 10 parameter model. This is a relatively modest increase in the misfit for a substantial reduction in number of parameters.

Memory Span Performance

It is worth comparing performance of subjects in our experiment with other reports of percentage recall of strings of various lengths. Our subjects recalled over 90% of six digit strings when solving simple equations and almost 75% when solving complex equations. Other reports have subjects recalling only 75% of six digits without a dual task (Crannell & Parrish, 1957). The very high performance of our subjects may reflect those who choose to select to be in an experiment that so heavily emphasized mathematics.[11]

Generally, research has found results that agree with our model's prediction of a gradual drop off in percentage correct reproduction for the entire

[11] To recruit subjects for this experiment, we advertised on a Carnegie Mellon electronic bulletin board with the title ''test your math prowess'' and noted that (unlike many experiments) one did not have to be a native speaker of English.

span with span size. For instance, Crannell and Parrish (1957) found a gradual drop off in accuracy as the digit span increases from 4 (nearly 100%) to 10 (nearly 0%). Unlike frequent popular characterizations, there does not appear to be a discontinuous ‘‘drop-dead’’ size. Indeed, typical span recall looks very much like that predicted by our model for the simple equations.

Because of their dual-task structure our experiments are not ideal if one’s real goal was to study the nature of the memory span. As we noted in the previous experiment there is the possibility that subjects systematically allocate more of their capacity to the span and away from equation solving as the span gets larger. This would produce a flattening of the curve until high spans. There are other complications in the span task not accounted for by our model. These include effects of acoustic confusion (minimal for digits), time-based forgetting, and confusion among serial positions. The next section will show that serial position confusion was a significant factor in our experiment. Thus, our model for the span task in no way captures all of the complexities of what is occurring. The model is just complicated enough to accommodate the basic interactions between the processing demands of the two tasks.

We suspect that time-based forgetting was behind the interactions with substitution that we found in this experiment. Complex equations were taking on the order of 30 s to solve. Substitution offers an opportunity to rehearse the part of the span and this may have significant benefit in bridging this interval. This benefit would be greatest for short spans where substitution served to rehearse a significant fraction of the span. This may be why subjects showed a substantial advantage in Fig. 5 when they had to substitute from a short span for a complex equation.

NATURE OF ERRORS

The model as described so far has treated errors as resulting only from a failure of retrieval. One might assume that such failures would be just omissions. However, this assumption does not fit well with the observed errors. Subjects almost never failed to enter an answer to an equation and many of their failed recalls in the memory span were of correct length. Thus, these were errors of commission rather than errors of omission. Of course, these errors could just reflect guesses on the subjects’ part; however, the significant observation was that subjects’ errors were systematic.

We developed an automatic analysis program for classifying errors in the algebra task. The program considered an error to be a substitution error if it could be produced by substituting a digit incorrectly recalled in position 1 or 2 of the span for a or b in the equation. If the error was not so classifiable, it was classified as a transformation error if it could be produced by one of the following eight transformation templates:

$$\text{term} + a = b \rightarrow \text{term} = b + a$$

$$\text{term} - a = b \rightarrow \text{term} = b - a$$

$$a{*}\text{term} = b \rightarrow \text{term} = a{*}b$$

$$\text{term}/a = b \rightarrow \text{term} = b/a$$

$$\text{term} + a = b \rightarrow \text{term} = b/a$$

$$\text{term} - a = b \rightarrow \text{term} = b{*}a$$

$$a{*}\text{term} = b \rightarrow \text{term} = b - a$$

$$\text{term}/a = b \rightarrow \text{term} = b{*}a.$$

If an error could not be classified as one of the above, it was classified as an arithmetic error if it could be produced by transforming a true addition fact $a + b = c$ or a true multiplication $a{*}b = c$ into a false fact by incrementing or decrementing one of the a, b, or c by 1 (e.g., $3 + 4 = 8$; $3 \times 5 = 12$). In our opinion this classification underestimates the frequency of systematic errors since some appeared to result from multiple slips of this sort. Errors classified by our scheme occurred much more frequently than chance. We calculated chance by the following procedure: For each experiment, we took all the problems for which subjects made errors and randomly permuted assignment of wrong answers to the problems. Then we applied our categorization program to these permuted wrong answers. In Experiment 1 actual subject errors could be classified by this scheme 72% of the time compared to 20% of the permuted errors. In Experiment 2, 64% of the actual errors were systematic in Experiment 2 compared to a 5% chance level.

The most common transformation errors were

$$\text{term} + a = b \rightarrow \text{term} = b + a$$

or

$$\text{term} - a = b \rightarrow \text{term} = b - a.$$

This ‘‘transposition’’ error occurred for 3.0% of the problems in Experiment 1 and 2.0% in Experiment 2. (In both experiments it was applicable on half the trials.) All other transformation errors occurred 0.5% of the time in Experiment 1 and 1.5% of the time in Experiment 2. Arithmetic errors occurred 0.5% of the time in Experiment 1 and 4.1% of the time in Experiment 2. The greater frequency in Experiment 2 reflects the larger number of opportunities for making arithmetic errors with fractions. Substitution errors occurred in 0.1% of the problems in Experiment 1 and 0.9% of the problems in Experiment 2.

Many errors in the digit span task were also errors of commission. Some of these errors were systematic. Subjects showed a tendency to recall a digit from a near position in the span producing a generalization effect (Nairne, 1992). We aggregated the data for the digit-span of 6 from Experiments 1

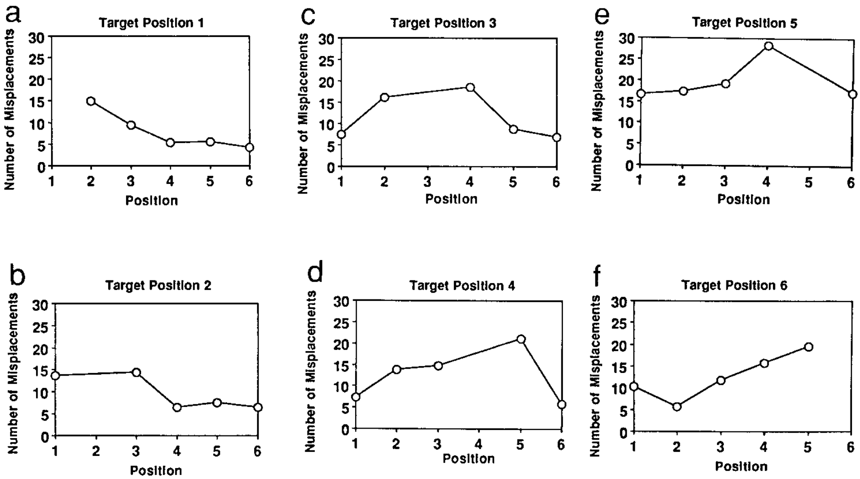

FIG. 8. Number of digits misplaced at each position for each target position.

and 2 to provide a representative illustration of these effects. Forty-six percent of the errors involved omission of the target digits. Such an omission could occur in a number of ways: Sometimes subjects did not recall the string at all, sometimes they recalled a fragment of the string, and sometimes they recalled a full six digit string but did not include the target digit. However, the other 54% of the time, the digit was recalled in the string but not in the right position. Figure 8 provides the data to show that these were not wild guesses and graphs the frequency of misplacements for each target position. As can be seen, there are positional uncertainty gradients such that subjects are most likely to misplace the digit in an adjacent position.

Partial Matching and Errors of Commission

The facts that error frequency was impacted by memory load and errors were misretrievals reinforces the localization of working-memory limitations in memory retrieval. The question remains of whether we can account for the exact nature of these errors in ACT-R. In ACT-R the most elegant way to account for these errors of commission is to allow chunks which are quite active but only partially match to be retrieved instead of the correct ones. Partial matching has received support, both empirically and computationally, in recent work of Reder (Kamas, Reder, & Ayers, in press, 1994; Reder & Cleeremans, 1990; Reder & Kusbit, 1991; Reder & Ritter, 1992; Reder, Schunn, Nhouyvanisvong, Richards, & Stroffolino, in preparation). There is a straightforward modification to ACT-R that will enable it to accept partial matches: Rather than simply rejecting a chunk if it matches we lower its match score. The initial match score of a chunk *i* is just its activation A_i but this is decremented for each mismatch. Thus, if a chunk $3 + 5 = 8$ is retrieved as an answer to a pattern for $3 + 4 = ?$, its match score is its activation minus a measure of the mismatch between 5 and 4. The degree of mismatch will be a function of the similarity between 4 and 5.

This scheme can be justified from a rational perspective if we allow that things do not have to match perfectly to be useful. Certainly, in matching real world things like faces which can vary in their dimensions and grow objects like mustaches, this makes sense. Only in highly formal domains like mathematics do things have to match perfectly to be used.

To get some variability in the responding we then added Gaussian noise to the activation levels. The odds formula in Eq. (2) was also based on the assumption of a Gaussian noise added to activation values which would cause them to sometimes fall below a threshold activation. In the current version, the Gaussian noise will occasionally cause the activation of the correct chunk to fall below the activation value of a distractor.

Let us first see how this mechanism can account for the pattern of arithmetic errors. When retrieving the sum of 2 and 5, both numbers are made sources and contribute activation to the correct fact: $2 + 5 = 7$. Many other facts also receive activation from these and other sources and might gather more activation because of a Gaussian noise in the activation values. To favor close matches, similarity values between numbers are set to reflect their absolute difference.[12] Therefore, the penalty for $2 + 6 = 8$ matching $2 + 5$ will be less than the penalty for $2 + 1 = 3$, because 6 is more similar to 5 than 1 is. All things being equal, the former will be more active than the latter and will have a better chance of being retrieved (if a misretrieval occurs), which explains the predominance of close matches. This will produce the pattern of arithmetic errors documented by Siegler (1988) and which we saw in our algebraic mistakes.

To see how this mechanism accounts for algebraic transformation errors, recall that transformations are all produced by the production *invert-transformation.* Faced with the equation

$$x - 3 = 4$$

it should retrieve the fact that ‘‘+ inverts −’’ and transform this to

$$x = 4 + 3.$$

However, if it retrieved ‘‘− inverts +’’ the following transposition error will be produced:

$$x = 4 - 3.$$

Since the memory chunk for ‘‘− inverts +’’ receives a mismatch penalty, it should be retrieved only infrequently, when a large random fluctuation in activation levels occurs which favors it.

[12] Specifically, the similarity between i and j is $\exp(-|i - j|)$.

TABLE 6
Distribution of Errors (ACT-R Predictions in Parentheses)

Algebra task	Experiment 1	Experiment 2
Transformation errors	61% (55%)	26% (23%)
Arithmetic errors	8% (7%)	31% (24%)
Substitution errors	2% (2%)	6% (6%)
Unclassified errors	29% (37%)	36% (47%)

ACT-R predicts the interaction observed between such errors and memory load. The strength of association, S_{ji}, between ''−'' and the erroneous ''− inverts +'' is not as strong as that between ''−'' and ''+ inverts −'' because ''−'' is not as predictive of the erroneous chunk. Therefore, the erroneous chunk receives less activation than the target chunk. The amount of activation either receives is determined by the product of the source activation, W_j, and strength, S_{ji} (see Eq. (1)). Thus, the initial difference between the two (before the mismatch penalty) is greater in the case of smaller memory load.

Table 6 compares the distribution of error classifications in the two experiments and the results of an ACT-R simulation augmented to allow errors of commission. The correspondence is quite good. The position effect in the digit-span recall task can be explained in the same way as the arithmetic errors: since each memory is accessed by its positional index, small differences in position will carry a lesser penalty than large ones.

Note that the error mechanism producing the predictions of the mathematical model in Figs. 2c, 3c, 4c, 5c, 6c, and 7c is one that assumes error of omission resulting from failure to retrieve in a fixed interval. This is similar to but different from the current proposal where an incorrect production instantiation beats out a correct instantiation. The simulation with errors of commission produces patterns of data similar to those in Figs. 2–7. However, we have yet to find a mathematical form of it to fit to the data.

GENERAL DISCUSSION

These experiments have looked at the interaction between the complexity of an algebra task and the size of a concurrent memory span. Basically, manipulations of the complexity of either task impacted on the performance of the other. Thus, we have created a situation where performance is limited by what would traditionally be considered working-memory capacity. There were several pieces of evidence consistent with localizing the impact of these capacity limitations in memory retrieval. First, the impact of span size on the algebraic task was greater when substitutions were required, consistent with the results of Carlson et al. (1989). Second, there were three-way interactions among span size, substitution, and complexity in the algebra task which were predicted by this view. Third, the errors were primarily errors of misretrieval.

Finally, we achieved good fits to the data within the ACT-R model which localizes the effect in memory retrieval.

In ACT-R the limitation is in source activation (Eq. (4)). This in turn limits the ability to get declarative chunks sufficiently active so that they can be retrieved or reliably discriminated from partially matching chunks. If we take working memory to mean the amount of declarative memory that ACT-R can reliably and quickly access, then limitations on source activation imply limitations on working memory. The ACT-R concept of working memory is rather nontraditional for production systems where working memory is normally thought of as some fixed set of information. The graded character of the effects in these experiments is clearly consistent with the ACT-R conception.

In addition to fitting the performance measures of error rate and latency, an extension of the ACT-R model was shown capable of accounting for the qualitative pattern of errors. Most of these errors were errors of commission and could be explained by incorrect retrieval of memories which were similar to the target memory and which were active in the experimental context.

It is interesting to try to characterize the implications of this research for Baddeley's (1986) model of working memory. One might try to map the separate-capacity model described earlier onto this theory, identifying the digit capacity with the phonological loop and the equation capacity with the visuo-spatial sketchpad. If one accepts this mapping, the results would not be very favorable for Baddeley's model because the separate capacity model's fit to our two experiments does less well with more parameters. Moreover, we view the assumptions required to get even this good a fit as somewhat strained. Subjects do not report the use of covert rehearsal while equation solving which was how the model produced an impact of digit span on equation time.

However, we do not think that these identifications are the appropriate ones for the Baddeley model. It seems more likely that all of our tasks were tapping what Baddeley calls the central executive. The delays were too long to be mediated by a 2-s phonological loop even with occasional covert rehearsal. Certainly, algebra equation solving does not seem like an activity that can be supported by a peripheral slave system. So perhaps the best view of these data is that they are not relevant to Baddeley's theories of the phonological loop and spatio-visual sketchpad which have received the most attention. Rather, they are really concerned with the central executive where the notion of a single limited resource might be reasonable. The research in our paper shows that the activation limitations have an important impact on long-term retrieval. Just and Carpenter (1992) have similarly emphasized activation as the capacity limitation corresponding to Baddeley's central executive.[13]

[13] Interestingly, Miyake, Shah, Carpenter, and Just (1994) argue for a separate spatial and verbal working memory.

Finally, we want to comment on the research strategy reflected in this paper, which is to develop a simulation of a phenomenon, determine that it reproduces the basic qualitative character of the data, find a mathematical model of that simulation, and then optimize the fit of that model to the data to determine just how well the theory accounts for the results. We feel this reflects a powerful research strategy which is emerging in a number of efforts to test and develop large scale theories (e.g., McClelland, 1991). There is a real need to develop an integrated theory which is capable of accounting for a broad range of phenomena (Newell, 1991). Such theories offer the only real hope of transferring results from the laboratory to the real world where phenomena are not packaged into neat laboratory categories. On the other hand, there is a need to have such theories address the details of empirical phenomena that are the traditional tests of theoretical accuracy. By starting with a general-purpose simulation such as ACT-R, we achieve the desired broad generality. By producing a simulation for a specific task we achieve a detailed mapping of the theory to the experimental situation. By developing a mathematical model, we facilitate calculation of goodness of fit and identify the essential aspects of the large theory responsible for accounting for the phenomena in the experiment at hand.

REFERENCES

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

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

Anderson, J. R. (1989). Practice, working memory, and the ACT* theory of skill acquisition: A comment on Carlson, Sullivan, and Schneider. *Journal of Experimental Psychology: Learning, Memory, and Cognition,* **15,** 527–530.

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

Anderson, J. R., & Jeffries, R. (1985). Novice LISP errors: Undetected losses of information from working memory. *Human Computer Interaction,* **1,** 107–131.

Anderson, J. R., Reder, L. M., & Ritter, S. (in preparation). Complexity and algebraic competence.

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

Baddeley, A. D. (1992). Working memory: The interface between memory and cognition. *Journal of Cognitive Neuroscience,* **4,** 281–288.

Baddeley, A. D., & Hitch, G. (1974). Working memory. In G. A. Bower (Ed.), *Recent advances in learning and motivation, Vol. 8.* New York: Academic Press.

Carlson, R. A., Sullivan, M. A., & Schneider, W. (1989). Practice and working memory effects in building procedural skill. *Journal of Experimental Psychology,* **15,** 517–526.

Crannell, C. W., & Parrish, J. M. (1957). A comparison of immediate memory span for digits, letters, and words. *The Journal of Psychology,* **44,** 319–327.

Daneman, M., & Carpenter, P. A. (1980). Individual differences in working memory and reading. *Journal of Verbal Learning and Verbal Behavior,* **19,** 450–466.

Evans, J. St. B. T., & Brooks, P. G. (1981). Competing with reasoning: A test of the working memory hypothesis. *Current Psychological Research,* **1,** 139–147.

Gillund, G., & Shiffrin, R. M. (1984). A retrieval model for both recognition and recall. *Psychological Review,* **91,** 1–67.

Halford, G. S., Bain, J. D., & Maybery, M. T. (1984). Does concurrent memory load interfere with reasoning? *Current Psychological Research and Reviews,* **3,** 14–23.

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

Kahneman, D. (1973). *Attention and effort.* Englewood Cliffs, NJ: Prentice–Hall.

Kamas, E., & Reder, L. M. (1994). The role of familiarity in cognitive processing. In E. O'Brien and R. Lorch, *Sources of coherence in text comprehension: A festschrift in honor of Jerome L. Myers.* Hillsdale, NJ: Erlbaum.

Kamas, E., Reder, L. M., & Ayers, M. (in press). Partial matching in the Moses illusion: Response bias not sensitivity. *Memory and Cognition.*

Klapp, S. T., Marshburn, E. A., & Lester, P. T. (1983). Short-term memory does not involve the ''working memory'' of information processing: The demise of a common assumption. *Journal of Experimental Psychology: General,* **112,** 240–264.

McClelland, J. L. (1991). Stochastic interactive processes and the effect of context on perception. *Cognitive Psychology,* **23,** 1–44.

McClelland, J. R., & Rumelhart, D. E. (Eds.). (1986). *Parallel distributed processing: Explorations in the microstructure of cognition* (Vol. 2). Cambridge, MA: MIT Press/Bradford Books.

Miller, G. A. (1956). The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review,* **63,** 81–97.

Miyake, A., Shah, P., Carpenter, P., & Just, M. (1994). Individual differences in spatial and verbal working memory. In *Proceedings of the 35th Annual Meeting of the Psychonomics Society.* St. Louis, MO.

Nairne, J. S. (1992). The loss of positional certainty in long-term memory. *Psychological Science,* **3,** 199–202.

Newell, A. (1991). *Unified theories of cognition.* Cambridge, MA: Cambridge University Press.

Raaijmakers, J. G. W., & Shiffrin, R. M. (1981). Search of associative memory. *Psychological Review,* **88,** 93–134.

Reder, L. M., & Cleeremans, A. (1990). The role of partial matches in comprehension: The Moses illusion revisited. In A. Graesser & G. Bower (Eds.), *The psychology of learning and motivation* (Vol. 25, pp. 233–258). New York: Academic Press.

Reder, L. M., & Kusbit, G. W. (1991). Locus of the Moses Illusion: Imperfect encoding, retrieval or match? *Journal of Memory and Language,* **30,** 385–406.

Reder, L. M., & Ritter, F. (1992). What determines initial feeling of knowing? Familiarity with question terms, not with the answer. *Journal of Experimental Psychology: Learning, Memory, and Cognition,* **18,** 435–451.

Reder, L. M., Schunn, C., Nhouyvanisvong, A., Richards, D. R., & Stroffolino, P. (in preparation). Rapid feeling of knowing: An activation-based account.

Reitman, J. S. (1974). Without surreptitious rehearsal, information in short-term memory decays. *Journal of Verbal Learning and Verbal Behavior,* **13,** 365–377.

Rumelhart, D. E., & McClelland, J. L. (Eds.). (1986). *Parallel distributed processing: Explorations in the microstructure of cognition* (Vol. 1). Cambridge, MA: MIT Press/Bradford Books.

Servan-Schreiber, E. (1991). *The competitive chunking theory: Models of perception, learning and memory.* Doctoral dissertation, Carnegie Mellon University.

Siegler, R. S. (1988). Strategy choice procedures and the development of multiplication skill. *Journal of Experimental Psychology: General,* **117,** 258–275.

(Accepted May 7, 1995)