Pervasive Episodic Memory: Evidence from a Control-of-Attention Paradigm

Erik M. Altmann (altmann@gmu.edu)
Wayne D. Gray (gray@gmu.edu)
Human Factors & Applied Cognition, George Mason University,
Fairfax, VA 22030

Abstract

Events appear to be represented distinctly in memory in large numbers at a fine grain, even in tasks in which memory retention is not a primary performance measure. In Experiment 1, participants classified character strings in sequences governed by randomly-alternating instructions. Response times were fastest near the start of a sequence, slowed gradually throughout the sequence, then sped up again near the start of the next sequence. This speedup and gradual slowdown were modeled in the ACT-R architecture as a combination of priming and interference effects in episodic memory. The model correctly predicts the absence of these effects in Experiment 2, in which the instruction must be inferred from the trial stimulus and hence is not a source of priming. These findings suggest (a) that episodic encoding is a pervasive side effect of cognitive performance; (b) that elements of episodic memory interact through priming and interference—effects traditionally associated with semantic memory; and (c) that brief interruptions of task performance have more complex effects than previously documented.

Introduction

Episodic memory, broadly defined, is memory with a temporal or contextual aspect, as distinct from a decontextualized semantic memory for a fact or concept (Tulving, 1983). Episodic memory has been implicated in a broad range of higher-level cognitive tasks, including software design (Jeffries, Turner, Polson, & Atwood, 1981), navigating large amounts of externally-represented information (Altmann & John, in press), learning interfaces by exploration (Rieman, Young, & Howes, 1996), discourse comprehension (Kintsch, 1998), and in general the efficient search of problem spaces (Howes, 1993). In each of these examples, episodic memory captures a history of events that influences task performance. However, the complexity of task performance in such domains makes the nature of episodic memory difficult to assess. What kinds of events are stored in memory, and at what temporal grain-size?

To gain control over such questions, researchers have typically adopted memory-oriented paradigms, in which performance is measured in terms of success at recalling or recognizing elements studied in a particular context (e.g., Tulving, 1983; Logan, 1988). However, such paradigms ignore the interaction of memory and other aspects of cognition. What is the role and nature of episodic memory within a wider cognitive framework?

This paper examines the role of episodic memory in a paradigm otherwise used to study the control of attention (Allport, Styles, & Hsieh, 1994; Gopher, Greenshpan, & Armony, 1996; Rogers & Monsell, 1995). The paradigm requires participants to classify simple stimuli according to varying instructions, allowing us to study the role of memory (for the current instruction) in a task with a decision-making component (judging the class of a stimulus according to the current instruction). In Experiment 1, classification of stimuli was interrupted at random times with a new instruction. These interruptions were found to have complex effects on subsequent trials, including a transient improvement in response time occurring soon after an interruption. We describe a computational cognitive model that accounts for this speedup in terms of priming between elements of episodic memory. From the model we predict that this speedup will be absent in tasks without distinct instructions. This prediction is confirmed in Experiment 2, in which the instruction is implied by the stimulus and is not presented separately.

Our results suggest that fine-grain events are stored pervasively in memory as a side effect of performance, even when retention of information in memory is not itself a performance measure. Conversely, episodic memory appears to influence cognition through priming and interference, effects typically associated with semantic and perceptual memory (Tulving, 1983; Tulving & Schacter, 1990).

Experiment 1

In Experiment 1, participants classified one string of letters per trial, based upon an instruction appearing at the start of that sequence of trials. On each trial, the string consisted of one letter repeated one or more times. Two classification tasks were used. For *Groupsize*, participants judged the number of elements in the string of letters. The correct response was *low* if Groupsize was fewer than five (1, 2, 3, 4) and *high* if it was greater than five (6, 7, 8, 9). For *Place*, participants judged the string as *low* if the constituent letter

was near the beginning of the alphabet (a, b, c, d) and *high* if it was near the end of the alphabet (w, x, y, z). The response to a trial caused the next stimulus to be displayed immediately.

Trials occurred in blocks of 20 grouped into two sequences or *runs*. The first run was governed by an instruction appearing at the start of the block (the *start instruction*). The first run continued until a second instruction appeared (the *interrupt instruction*). One interrupt instruction per block appeared at a randomly-selected point near the middle of the block (ranging from after trial 7 to after trial 13). Participants responded to an instruction by pressing the space bar, after which the first trial of the following run was displayed immediately. At the end of the block, participants received latency and error feedback encouraging them to work quickly and accurately. Participants completed 192 blocks each, of which the first 16 were excluded from analysis to factor out learning effects. Also excluded were trials falling more than three standard deviations from the mean response time of each participant.

One independent variable was the *Position* of a trial within the block. The *Baseline* level of Position was the mean of the response times for the four trials immediately before the interrupt instruction. The *I+1* and *I+2* levels were response times for the first and second trials (respectively) after the interrupt instruction.

A second independent variable was *Instruction*, meaning the kind of instruction presented at interrupt. This variable manipulated which task to perform after the interrupt (Gopher, et al., 1996) and served to require attention to the interrupt. One level was *Switch*, meaning that the interrupt and start instructions were different. A block was a Switch block if the start instruction was Groupsize and the interrupt instruction was Place or vice versa. The other level was *Noswitch*, meaning that the start and interrupt instructions were the same. Switch and Noswitch blocks were presented randomly within participants.

We predicted that the interrupt instruction would generate two kinds of cost (after Gopher et al., 1996). *Interrupt cost* is the performance penalty due simply to interrupting a task, as measured once the task is resumed. *Switch cost* is an extra penalty on Switch blocks due to resuming a different task. We predicted that interrupt and switch costs would be localized to I+1 (after Rogers & Monsell, 1995), and hence that I+2 performance should be the same as Baseline.

The dependent measure was response time (RT). Twenty George Mason University undergraduates participated in the study for course credit.

Results and Discussion

We tested our predictions with a 3x2 ANOVA on Position (Baseline, I+1, I+2) and Instruction (Switch, Noswitch). Figure 1 summarizes the results.

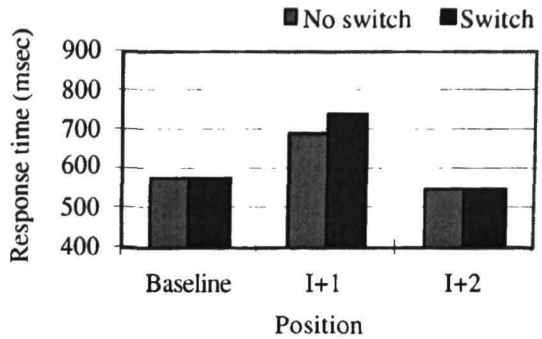

Figure 1: Interrupt and switch costs on I+1 and interrupt benefit on I+2.

Interrupt cost was indicated by the main effect of Position, $F(2, 9939) = 803.0$, $p < .001$.[1] To identify which positions were affected, we applied orthogonal contrasts to Baseline (571 msec) and I+1 (706 msec) and to Baseline and I+2 (541 msec). Baseline was faster than I+1, $t = 30.7$, $p < .00001$, replicating Gopher et al. (1996). Baseline was slower than I+2, $t = -6.8$, $p < .00001$.

Switch cost was indicated by the Position x Instruction interaction, $F(2, 9939) = 27.8$, $p < .001$. To identify which positions were affected, we examined simple effects of Instruction at each level of Position. There was no effect at Baseline, $F(1, 9939) = 1.1$, n.s., as one would expect given that the task switch had not yet occurred. On I+1, Noswitch (689 msec) was faster than Switch (744 msec), $F(1, 9939) = 71.6$, $p < .01$, again replicating Gopher et al. (1996). On I+2 there was no effect of Instruction, $F(1, 9939) = 1.7$, n.s. Thus both switch cost and interrupt cost were localized to I+1, as predicted.

We did not predict that Baseline would be slower than I+2. To explore this effect, we looked for intertrial trends in response time. A systematic slowing trend within a run would explain how Baseline (at the end of one run) could be slower than I+2 (near the start of another).

Response times for the first through seventh trials in a run are plotted in Figures 2 and 3. Figure 2 shows trials after the start instruction, and Figure 3 shows trials after the

[1] Degrees of freedom for this design were calculated as follows. Participant was treated as an independent variable to remove between-participants variance from the error term (Howell, 1997). Block was not treated as an independent variable, leaving 88 observations per level of Instruction (Switch, Noswitch) per Participant. There were 501 outlying observations. Total df were computed from Position, Instruction, Participant, and observations per cell, minus outliers: (3 * 2 * 20 * 88) - 501 1 = 10058. Treatment df were allocated to Position (2), Instruction (1), Participant (19), Position x Instruction (2), Position x Participant (38), Instruction x Participant (19), and Instruction x Participant x Position (38), leaving 10058 - 119 = 9939 df for the error term.

interrupt instruction.[2] Both figures show an initial speedup between the first and second trials, reflecting recovery from processing the instruction. This speedup is followed by a linear slowing trend, t = 12.8 and 10.1, respectively, $p < .00001$.[3] We refer to this speedup and subsequent slowdown as *within-run slowing*.

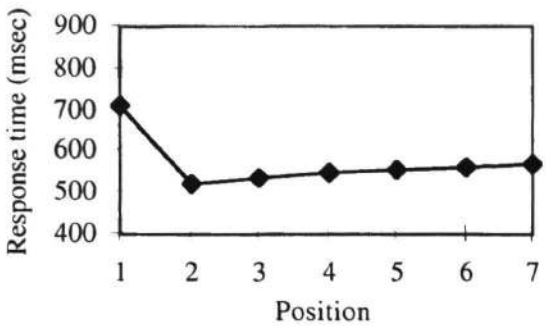

Figure 2: Empirical response times for trials in the 1st through 7th positions after the start instruction.

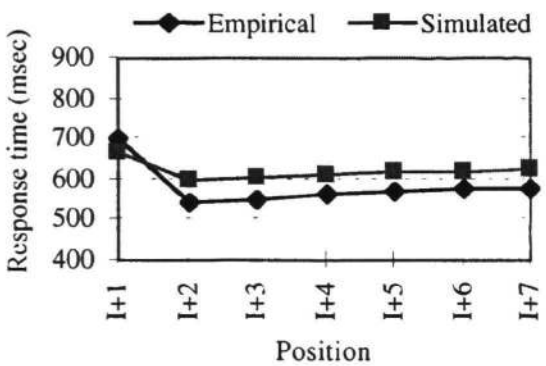

Figure 3: Empirical and simulated response times for trials in the 1st through 7th positions after the interrupt instruction.

The semantic content of the instruction apparently played no role in within-run slowing. There was no Position x Instruction interaction, $F(5, 20861) = 1.2$, n.s., meaning that the effect occurred regardless of whether the interrupt presented a new task or presented the old task over again. The event of instruction processing alone, independent of semantic content, generated a benefit on I+2 that attenuated gradually across the rest of the run.

[2] Response times for positions 1 through 7 were 713, 524, 535, 550, 556, 562, and 568 msec, and for I+1 through I+7 were 703, 542, 549, 561, 571, 575, and 577 msec.

[3] t-values reflect a significant linear term in an orthogonal polynomial contrast applied to the second through seventh trials after the start (2 through 7) and interrupt (I+2 through I+7) instructions. No higher-order terms approached significance.

A Model of Within-Run Slowing

To account for within-run slowing we constructed a model using the activation mechanism of the ACT-R cognitive architecture (Anderson & Lebiére, 1998). This mechanism has been used to account for a variety of priming and interference effects in semantic memory (e.g., Anderson & Lebiére, 1998). Applying it to another phenomenon helps validate the mechanism and links a variety of phenomena in a relationship of mutual constraint (Newell, 1990). Below we describe key aspects of the activation mechanism and additional assumptions in our model about the encoding of events in memory.

ACT-R is a production system with a long-term procedural memory containing productions and a long-term declarative memory (DM) containing *chunks*. Chunks can be *linked* to other chunks, allowing for structured declarative representations. ACT-R also implements a mechanism for focusing internal attention: The *goal chunk* is a privileged chunk, selected by productions, that in turn controls the selection of productions. In each cycle of operation, ACT-R selects one production that matches the goal chunk. The remaining conditions of this production are matched against DM. If DM contains chunks that match all the conditions, those chunks are retrieved from DM, bound to the production, and the production fired. The actions of a production can modify, remove, or replace the goal chunk.

ACT-R makes predictions based on the latency of matching and firing a production. Latency is determined in part by the activation of chunks retrieved during the match process (the higher the activation, the lower the latency to retrieve a chunk from DM). Activation is the sum of two terms: baseline activation, which belongs to a chunk independent of any other knowledge in the system, and which will not concern us here; and *source activation*, the product of *goal activation* and *associative strength*.

Goal activation captures the notion that knowledge in the focus of attention primes related knowledge. This is implemented as activation emanating from the goal chunk and spreading out through chunks to which it is linked. If the goal chunk is linked to chunk A and chunk A is linked to chunk B, chunk A conducts goal activation to chunk B.

Associative strength captures the notion that the retrieval of one memory element may predict the need to retrieve another. In terms of a task like cued recall, the idea is that the more targets are associated with a cue, the less the presence of the cue predicts the need to retrieve any one target (Anderson & Matessa, in press; Anderson, Reder, & Lebiére, 1996; Lovett, Reder, & Lebiére, 1997). The associative strength between chunk A and another chunk B is a function of how many chunks in total are linked to A. The more chunks are linked to A (the greater the "fan" out of A), the less the associative strength between A and any of them, including B.

A key additional assumption we make in our model is that of *pervasive episodic memory*, under which every task-

related event (i.e., every instruction and trial) is encoded distinctly in DM. This is in the spirit of Logan's (1988) instance theory, which posits a unique trace for every exposure to a stimulus. It also has face validity compared to the alternative: Were the model to reuse previously-allocated memory to encode a new event, then some old event would be forgotten literally without a trace. Without additional mechanisms that choose which elements to delete, forgetting by deletion implies indiscriminate reuse and hence complete episodic amnesia.

Pervasive episodic memory requires that the model keep track of which instruction and trial are current among the many in DM. The current instruction is distinguished by being linked to the goal chunk. This has the side benefit of making the current instruction immediately available for other uses. In particular, the current instruction is available to link to each new stimulus as the stimulus is encoded. Linking the instruction to the stimulus makes it faster for the classification process to retrieve the stimulus for processing after the stimulus has been encoded,[4] because activation spreads from the goal chunk through the instruction chunk to the stimulus chunk.

As the run progresses and more stimulus chunks are linked to the current instruction, associative strength between instruction and all stimuli decreases. This reduces the source activation flowing through the instruction chunk to each new stimulus chunk, accounting for the apparent trial-by-trial slowdown. It also accounts for the *release* of slowing when a new instruction is processed. The new instruction replaces its predecessor in the goal chunk, and because initially it has no associated stimuli, its strength of association with stimuli early in a run will be high.

Figure 2 plots our simulated data against the empirical data from Experiment 1. Between I+1 and I+2 both slopes are negative, indicating recovery from interrupt cost. The crossing lines reflect a work-in-progress account of interrupt and switch costs (Altmann, Gray, Lipps & Trickett, submitted). Between I+2 and I+7, the simulated and empirical data have the same upward slope, meaning that the model captures within-run slowing. The gap between the curves in this interval reflects our focus on a qualitative fit achieved with a minimum of parameter optimization. This qualitative fit is sufficient to make a zero-parameter prediction concerning within-run slowing, described next.

[4] We assume that retrieval of the stimulus from DM after it is encoded is a necessary precursor to classifying it. ACT-R provides a mechanism, known as *parameter passing*, that would allow this to occur by bypassing DM, but this mechanism has no clear theoretical or empirical justification and may overpredict the reliability of certain memory operations. Thus parameter passing was not used in our model.

Experiment 2

The Experiment 1 model makes a prediction about the effect of maintaining an instruction in the focus of attention. The current instruction was linked to each new stimulus to ensure correct performance, resulting in instructional priming of each new stimulus. If the instruction had to be inferred from the stimulus, then the link between instruction and stimulus would have to be made anew for every trial. There would be no source of instructional priming, and hence no opportunity for the subsequent interference among trials that accounts for within-run slowing.

Experiment 2 allowed us to test this prediction. Stimuli consisted of individual characters appearing on the computer screen one at a time. If the stimulus was a digit (1, 2, 3, 4, 5, 6, 7, 8), the task was to judge it as odd or even. If the stimulus was a letter (G, K, M, R, A, E, I, U), the task was to judge it as consonant or vowel. Trials occurred in runs of digits alternating with runs of letters, with the length of a run ranging from one to four trials. For example, the sequence of trials 243GK6RAKM contains runs of three digits, two letters, one digit, and four letters. There was no instruction intervening between runs, meaning that the task could change immediately from one trial to the next.

Participants completed 60 blocks of 40 trials, of which 18 blocks were excluded from analysis.[5] The independent variable was *Position* within a run (1, 2, 3, 4), which differs from Position in Experiment 1 in that no interrupt separates different Positions. All trials in a block were contiguous, with task switches indicated by stimulus type rather than by instructions inserted between trials. Thus runs are defined by stimulus type alone, and by definition Position 1 is always the first trial after a task switch. The dependent measure was response time. Ten George Mason University undergraduates participated in the study for course credit.

The Experiment 1 model was changed to do the Experiment 2 task. The changes reflect the need to infer task from stimulus and the absence of distinct instruction events. On each trial, the new model encodes the stimulus first and then retrieves the appropriate task from memory. There is no separate instruction chunk to link to the stimulus as the stimulus is encoded, and therefore no source of priming when the stimulus is retrieved by the classification process. Quantitative parameters were the same for both models.[6]

[5] The first 8 blocks per participant were excluded from analysis to factor out learning effects, matching the 320 initial trials excluded per participant in Experiment 1 (16 blocks * 20 trials per block). Also excluded were 10 blocks interspersed per session in which runs of length five were included to inhibit learning of the maximum run length. In the retained blocks, trials falling more than three standard deviations from each participant's mean RT were again excluded as outliers.

[6] Both models used the default global parameter values supplied with ACT-R 4.0b3, except for latency factor (F=0.5 rather than the default F=1.0) and goal activation (W=0.5 rather than the default W=1.0).

Our predictions based on the model are shown in Figure 4. The key prediction is the absence of within-run slowing, indicated by equal simulated RT for Positions 2 through 4.

A second prediction is of switch cost, indicated by elevated RT on Position 1. This follows from the need to infer the task from the stimulus (e.g., the odd/even task would be inferred from a digit). In the model, each chunk representing a stimulus identity is linked to the appropriate task chunk. On a given trial, the model first encodes the stimulus by retrieving its identity from DM. The model then retrieves the appropriate task, to perform the task on the stimulus. On the following trial, the stimulus is encoded more quickly if it implies the same task. The goal chunk is still linked to that task from the previous trial, making the task a conduit for goal activation to spread to the stimulus identity. Thus what others characterize as switch cost (Rogers & Monsell, 1995; Allport et al., 1994) is in our view a benefit due to priming when consecutive tasks are the same.

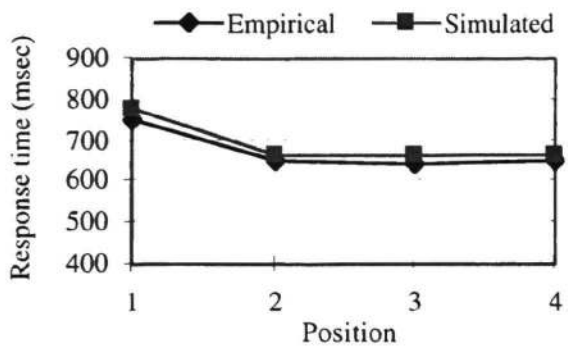

Figure 4: Empirical and simulated response times for trials in the 1st through 4th positions after a task switch.

Results

The empirical data are also shown in Figure 4.[7] The effect of Position was significant, $F(3, 16522) = 184.9$, $p < .001$.[8] Orthogonal contrasts performed between each Position and the mean of the subsequent Positions showed Position 1 slower than the mean of Positions 2 through 4, $t = 22.8$, $p < .00001$. This confirms the prediction of switch cost. The two remaining contrasts were not significant, $-1 < t < 1$, confirming the predicted absence of within-run slowing.

[7] Empirical means for Positions 1 through 4 were 754, 647, 641, and 649 msec.

[8] Degrees of freedom were calculated as follows. Participant was again treated as an independent variable. For each participant there were 672, 504, 336, and 168 observations for Positions 1 to 4, respectively, in the 42 blocks included in the analysis. Among these were 238 outlying observations. Total df were computed from Participant, Position, and observations per Participant per level of Position, minus outliers: 10 * (672 + 504 + 336 + 168) - 238 1 = 16561. Treatment df were allocated to Participant (9), Position (3), and Participant x Position (27), leaving 16561 - 39 = 16522 df for the error term.

General Discussion

Evidence from Experiment 1 suggests that processing an instruction presented during an interrupt induces an improvement in response time soon after task resumption. This improvement attenuates gradually but recurs soon after the following instruction. This pattern of effects, which we refer to as within-run slowing, seems to be independent of the instruction's meaning.

We account for within-run slowing by assuming that task-related events are encoded pervasively in memory, such that every instruction and stimulus encountered during the session leaves a distinct trace. In the context of ACT-R's activation mechanism, these distinct traces or chunks explain within-run slowing as an effect of priming and interference. The current instruction primes retrieval of the stimulus in preparation for processing the current trial. The marginal priming effect decreases as the instruction becomes less strongly associated with any particular stimulus, a kind of interference that is released as soon as the next instruction is fully encoded. Our model predicts the absence of within-run slowing when performance does not require an instruction to be retained in the focus of attention. This prediction was confirmed in Experiment 2, supporting our model in general and our assumption of pervasive episodic memory in particular.

We conclude that events may be stored in memory in large numbers at a fine grain, even in tasks where this is not logically necessary for successful performance. This suggests that episodic memory of this kind may influence other cognitive tasks as well. Converging evidence comes from a model of memory for attention events developed from observations of real-world problem solving (Altmann, Larkin, & John, 1995; Altmann & John, in press). The domain, data, and underlying cognitive architecture for that model were quite different from those for the current model, but the implied grain-size at which information is stored (roughly two traces per second) is quite similar.

The present research contributes to the debate over the distinction between episodic and semantic memory. Tulving (1983) has maintained that "priming effects are mediated by, and reflect the operations of, a system other than episodic memory." Contrary to this view, we have shown that priming flows from encoded events: Within-run slowing occurs even if the interrupt instruction says simply to continue the same task, implying that the episodic and not the semantic representation of the instruction causes the effect. In addition, we have shown that priming and interference effects in episodic and semantic memory can be accounted for by the same underlying mechanisms. The emergence of such functional symmetries bears out the promise of studying phenomena in context (Newell, 1973) and of using cognitive architectures to unify phenomena with existing theory (Newell, 1990).

With respect to human factors psychology, our data show that brief interruptions have more complex effects on

performance than previous studies document. In addition to immediate interrupt and switch costs (Gopher, et al., 1996; Rogers & Monsell, 1995; Allport, et al., 1994), there are delayed effects that include an interrupt *benefit* under certain circumstances. Whether task environments can be designed to exploit the apparent influence of episodic memory on performance is a question to address in future studies.

Acknowledgments

This work was supported by post-doctoral fellowship to the first author from the Krasnow Institute for Advanced Study at George Mason University, and by a grant from the Air Force Office of Scientific Research (F49620-97-1-0353) to the second author. We wish to thank Audrey W. Lipps, Susan K. Schnipke, and Susan B. Trickett for help with data collection and analysis, and Deborah A. Boehm-Davis, Irvin R. Katz, Audrey W. Lipps, Sheryl L. Miller, Carol L. Raye, J. Gregory Trafton, Susan B. Trickett, and anonymous reviewers for comments on this paper.

References

Allport, A., Styles, E. A., & Hsieh, S. (1994). Shifting intentional set: Exploring the dynamic control of tasks. In C. Umilta & M. Moscovitch (Eds.), *Attention and Performance IV* (pp. 421-452). Cambridge, MA: MIT Press.

Altmann, E. M., Gray, W. D., Lipps, A. L., & Trickett, S. B. (1998). *Individual differences in interrupt processing.* Manuscript submitted for publication.

Altmann, E. M. & John, B. E. (in press). Episodic indexing: A model of memory for attention events. *Cognitive Science.*

Altmann E. M., Larkin, J. H., & John, B. E. (1995). Display navigation by an expert programmer: A preliminary model of memory. In *CHI 95 Conference Proceedings* (pp. 3-10). New York: ACM Press.

Anderson, J. R. & Lebiére, C. (1998). *The Atomic Components of Thought.* Hillsdale, NJ: Erlbaum.

Anderson, J. R. & Matessa, M. P. (in press). A production system theory of serial memory. *Psychological Review.*

Anderson, J. R., Reder, L. M., & Lebiére, C. (1996). Working memory: Activation limitations on retrieval. *Cognitive Psychology,* 30, 221-256.

Gopher, D., Greenshpan, Y., & Armony, L. (1996). Switching attention between tasks: Exploration of the components of executive control and their development with training. In *Proceedings of the Human Factors and Ergonomics Society 40th Annual Meeting* (pp. 1060-1064). Philadelphia: HFES.

Howell, D. C. (1997). *Statistical Methods for Psychology.* New York: Duxbury Press.

Howes, A. (1993). Recognition-based problem solving. *Proceedings of the 15th Annual Meeting of the Cognitive Science Society* (pp. 551-556). Hillsdale, NJ: Erlbaum.

Jeffries, R., Turner, A. A., Polson, P. G., & Atwood, M. E. (1981). The processes involved in designing software. In J. R. Anderson (Ed.), *Cognitive Skills and Their Acquisition* (pp. 255-283). Hillsdale, NJ: Erlbaum.

Kintsch, W. (1998). *Comprehension: A paradigm for cognition.* New York: Cambridge University Press.

Logan, G. D. (1988). Toward an instance theory of automatization. *Psychological Review,* 95, 492-527.

Lovett, M. C., Reder, L. M., & Lebiére, C. (1997). Modeling individual differences in a digit working memory task. In *Proceedings of the Nineteenth Annual Conference of the Cognitive Science Society* (pp. 460-465). Hillsdale, NJ: Erlbaum.

Newell, A. (1973). You can't play 20 questions with nature and win: Projective comments on the papers of this symposium. In W. G. Chase (Ed.), *Visual Information Processing* (pp. 283-308). New York: Academic Press.

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

Rieman, J., Young, R. M., & Howes, A. (1996). A dual-space model of iteratively deepening exploratory learning. *International Journal of Human-Computer Studies,* 44, 743-775.

Rogers, R. D., & Monsell, S. (1995). Costs of a predictable switch between simple cognitive tasks. *Journal of Experimental Psychology: General,* 124(2), 207-231.

Tulving, E. (1983). *Elements of Episodic Memory.* New York: Oxford University Press.

Tulving, E. & Schacter, D. L. (1990). Priming and human memory systems, *Science,* 247, 301-306.