Title
The temporal dynamics of strategy execution in cognitive skill learning

Author
Bajic, Daniel Andrew

Publication Date
2009

UNIVERSITY OF CALIFORNIA, SAN DIEGO

The Temporal Dynamics of Strategy Execution in Cognitive Skill Learning

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy

in

Psychology

by

Daniel Andrew Bajic

Committee in charge:
Professor Timothy Rickard, Chair
Professor Stephan Anagnostaras
Professor Jeffrey Elman
Professor Victor Ferreira
Professor Jaime Pineda

2009

The Dissertation of Daniel Andrew Bajic is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

____

____

____

____

____
Chair

University of California, San Diego

2009

EPIGRAPH

“There is no essential difference between practice and learning
except that the practice experiment takes longer.”

Robert S. Woodworth

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

Chapter 2, in full, is a reprint of the material as it appears in Bajic, D., & Rickard, T. C. (2009). The temporal dynamics of strategy execution in cognitive skill learning. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 35*(1), 113-121. Copyright 2009 by the American Psychological Association. Reproduced with permission. The official citation that should be used in referencing this material is listed above. The use of APA information does not imply endorsement by APA. The dissertation author was the primary investigator and author of this paper.

Chapter 3, in part, has been submitted for publication of the material as it may appear in Bajic, D., & Rickard, T. C. (2009). The temporal dynamics of strategy execution in cognitive skill learning: The case of simple perceptual-motor algorithms. The dissertation author was the primary investigator and author of this paper.

This dissertation was made possible by the help and generosity of many individuals, and I would like to offer my deepest thanks to all of the following:

The chair of my dissertation committee, Dr. Timothy Rickard. My time in the Rickard Lab has been informative, productive, and fun, and I'm grateful to have been a member.

Dr. Stephan Anagnostaras, Dr. Jeffrey Elman, Dr. Victor Ferreira, and Dr. Jaime Pineda, for their time, their help, and their advice.

The research assistants who collected the data for this dissertation: Elena Eiden, Jazmin Morales, Laura Muzzatti, Laura Osborne, Wang Pan, Kristin West, Lisa Graves, Jennifer Um, Shivali Agarwal, Samir Patel, Jon Strunk, and Caitlin Oldenkamp. It's been a pleasure working with all of them.

All of the many individuals who participated as research subjects. Truly the unsung (or insufficiently sung) heroes of science.

My family, for more reasons than I could list.

And finally, I'd like to offer my gratitude to all the people that I remember when thinking about my time in grad school: the professors, the grads and undergrads, the staff, and the various unknown individuals that I encountered while walking across campus—all of whom made their own unique contributions to the spirit of the campus.

VITA

1999-2001	Research Assistant, University of California, San Diego
2001	Bachelor of Science, University of California, San Diego
2001-2009	Teaching Assistant, University of California, San Diego
2004	Master of Arts, University of California, San Diego
2009	Doctor of Philosophy, University of California, San Diego

PUBLICATIONS

Bajic, D., & Rickard, T. C. (2009). The temporal dynamics of strategy execution in cognitive skill learning. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 35*(1), 113-121.

Rickard, T. C., & Bajic, D. (2006). Cued recall from image and sentence memory: A shift from episodic to identical elements representation. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 32*(4), 734-748.

Rickard, T. C., & Bajic, D. (2005). In search of cross-talk facilitation in a dual-cued recall task. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 31*(4), 750-767.

Rickard, T. C., & Bajic, D. (2004). Memory retrieval given two independent cues: Cue selection or parallel access? *Cognitive Psychology, 48*(3), 243-294.

Rickard, T. C., & Bajic, D. (2003). Automatic mediation or absence of mediation? Commentary on Crutcher and Ericsson (2000). *Journal of Experimental Psychology: Learning, Memory, & Cognition, 29*(6), 1381-1386.

ABSTRACT OF THE DISSERTATION

The Temporal Dynamics of Strategy Execution in Cognitive Skill Learning

by

Daniel Andrew Bajic

Doctor of Philosophy in Psychology

University of California, San Diego, 2009

Professor Timothy Rickard, Chair

The transition from algorithmic to memory-based performance is a core component of cognitive skill learning. E.g., an arithmetic problem such as 4 x 3 may initially be solved with a repeated addition algorithm (4 + 4 + 4), but with practice the answer will be recalled directly from long-term memory (LTM). There has been debate about the temporal dynamics of strategy execution, with some models assuming a race (i.e., independent, capacity unconstrained parallel processing) between algorithm and retrieval, and others assuming a choice mechanism. This work introduces an original paradigm that permits (for the first time) the objective identification of strategy use on every trial, as well as the latency of each component step of an algorithm. I also introduce a technique for appropriately aggregating data across different learning curves. Results are uniquely consistent with a strategy choice mechanism involving a competition between the retrieval strategy and the 1st step of the algorithm. Some previously undiscovered skill-acquisition phenomena (such as increasing latency for algorithm initiation on trials immediately preceding the first correct direct retrieval for each item) are identified and discussed. Examination of partial-algorithm trials (in which the algorithm is initiated, but abandoned prior to completion in favor of direct retrieval) indicates that for algorithms consisting of multiple retrievals from LTM, the bottleneck extends beyond the 1st step of the algorithm, whereas for simple perceptual-motor algorithms, some parallel performance on later steps is possible. I introduce a theoretical framework that can accommodate the results found for different classes of algorithms. Results highlight the importance of studying partial-algorithm trials (something that has not been possible in previous skill-acquisition paradigms), and also the importance of considering the issue of efficiency in strategy scheduling as a factor that may affect performance over the course of practice.

Chapter 1

Skill Acquisition: An Introduction

“Practice makes perfect” is a phenomenon that is very well known, but not very well understood. We are all familiar with the basic principle of using repeated performance of a task over time as a means of improving performance. Present a child with a math problem, such as “What’s 4 x 3?”, and the child may initially find the answer in a slow and laborious manner. But present the same problem repeatedly over time, and the child will eventually begin answering the question with speed and seeming effortlessness (e.g., Siegler, 1988). What are the mechanisms underlying this change in the speed of performance? For a wide range of cognitive tasks, the major source—if not the sole source—of this speed-up is the shift from initial use of a slow, multistep algorithm to a faster and subjectively less effortful memory look-up (direct retrieval) of the answer (Logan, 1988; Rickard, 1997, 2004). A child presented with the arithmetic problem “4 x 3” may initially find the answer through an algorithm of repeated addition (4 + 4 + 4), but over time will learn to simply recall the answer from long-term memory (LTM).

For a task such as the above arithmetic problem, the algorithm itself consists of a series of retrievals from LTM. However, there are many tasks in which a direct-retrieval strategy competes against a perceptual-motor algorithm. For example, to find an unfamiliar room in an office building, a person may initially need to utilize a visuo-motor search algorithm; but a person who frequently returns to the same room will likely learn to directly retrieve the location from memory. To perform some editing function in a word processor, one may initially need to move the mouse pointer to a dropdown menu in search of the appropriate command; eventually, though, one may simply recall a particular shortcut key combination for accessing that command.

The transition from algorithm-based to direct-retrieval-based strategies is, in short, an ubiquitous phenomenon (Delaney, Reder, Staszewski, & Ritter, 1998; Hertzog, Touron, & Hines, 2007; Jenkins & Hoyer, 2000; Logan, 1988, 1992; Onyper, Hoyer, & Cerella, 2006; Palmeri, 1997; Rawson, 2004; Reder & Ritter, 1992; Rickard, 1997, 1999, 2004; Rickard & Bajic, 2003, 2006; Rogers, Hertzog, & Fisk, 2000; Schunn, Reder, Nhouyvanisvong, Richards, & Stroffolino, 1997; Touron, Hoyer, & Cerella, 2001), and the extent to which we understand this transition thus has important implications for our fundamental understanding of learning, of skilled performance, and of the underlying cognitive architecture. Yet there is ongoing debate regarding the nature of the transition, and of the ways in which competing strategies will or will not affect one another. In more concrete terms, if a child begins to solve a multiplication problem through the use of a repeated addition algorithm, would the execution of this strategy in any way interfere with the concurrent recall of the answer for that problem? If interference were found for tasks such as this, would this interference also extend to cases that involve simpler perceptual-motor algorithms? For example, would moving a mouse pointer to a dropdown menu interfere with the recall of a shortcut key combination? The experiments described in the following chapters have helped provide some new insight into these fundamental questions. Before they are described further, though, it is important to ground them within a historical perspective, and explain how the skill acquisition literature arrived at the questions that are currently being asked.

Here, a good starting point is a philosophical question: when we get better at any given activity, are we expanding our abilities, or are we better compensating for our limitations?

Skill Acquisition: Are We Becoming Richer, or Living Within Our Means?

While the latter view—skill as cognitive frugality—may seem to be a counterintuitive perspective, it has been an extremely influential viewpoint throughout much of the modern skill acquisition literature, and it is worth considering further, particularly in regard to what may be termed the capacity-based models of skill acquisition (e.g., Shiffrin & Schneider, 1977). The starting point for these models is the basic notion that we possess some limited cognitive resources, or "capacity limitations…that cannot be removed by training or practice" (Shiffrin & Dumais, 1981); such as, perhaps, the total amount of attention we can direct among different activities at any given moment. However, if a financial metaphor may be used, although we are living on a limited budget, we can learn to buy the same or equivalent goods at a lower cost. One early example that seemed to lend credence to this perspective came from comparisons of expert and novice chess players (see De Groot, 1965; Chase & Simon, 1973). It was found that (as might be expected) expert players had a superior ability to hold complex patterns of chess pieces in mind. However, it was also found that experts and novices did *not* differ in terms of the number of pieces of information they could hold in mind at any given time.

Rather, the experts simply learned to recognize familiar patterns of pieces, and could thus represent each familiar pattern as a single "chunk" of information, rather than as a collection of multiple distinct pieces of information (see De Groot, 1965; Chase & Simon, 1973; Miller, 1956). When dealing with unfamiliar patterns of chess pieces, the expert players were no better than the novices.

In relation to issues of short-term memory and the chunking of information (Miller, 1956), a capacity-based perspective clearly has some merit. However, the capacity limits that have received the greatest focus within the modern skill-acquisition literature have been the limits of attention, and research on how to circumvent these limits has focused on *automaticity,* which can roughly be defined as the ability to perform some practiced action without thinking about it. (As will be seen, though, no definition of automaticity will be entirely satisfying to everyone.)

It is easy to think of examples that appear to support the capacity-based perspective of skill acquisition. When first learning a new activity, such as driving (e.g., Brown & Poulton, 1961), or typing (e.g., Salthouse, 1986), or playing a musical instrument (e.g., Shiffrin & Dumais, 1981), people initially need to give their full attention to the mechanics of performance, but over the course of time and practice, more and more aspects of performance seem to occur with little or no attentional effort, freeing the skilled individual to direct attention to other concurrent activities, such as speaking on a cell phone while driving. This led some researchers to make an explicit dichotomy between automatic vs. controlled processing (e.g., Shiffrin & Dumais, 1981), or automatic vs. attentional processing (e.g., Logan, 1980;

McDaniel & Einstein, 2000). However, problems with this perspective soon emerged. (In regard to the perils of using cell phones while driving, see Redelmeier & Tibshirani, 1997.)

First, contrary to what many had assumed, it appears that no skilled activities can be completely independent of attention. Even with the most highly trained and seemingly uncontrollable automatic activities, such as the unintended word-reading that gives rise to the Stroop effect (and that also helped define many researchers' notions of what constituted a genuinely automatic activity; see Logan, 1988), manipulations of attention can still affect performance (e.g., Besner, 2001). Thus a distinction such as automatic vs. attentional is a false dichotomy.

In a further problem for the capacity-based perspective, researchers could not agree on how best to define the central concept of automaticity. For example, is an automatic activity one that can occur without thought, but be highly controlled (such as walking), or is it an activity that can be triggered without or against one's intentions (like a bad habit). Would both characteristics be necessary? Would either be sufficient? Multiple researchers attempted to develop a more specific definition of automaticity, but could not agree on the number of characteristics that defined automaticity (e.g., five according to Hasher & Zacks, 1979; twelve according to Schneider, Dumais, & Shiffrin, 1984).

Amid this backdrop, a more recent trend in the skill-acquisition literature (e.g., Logan, 1988; Nosofsky & Palmeri, 1997; Rickard, 1997) has shifted away from the previous focus on attentional savings, and has instead put an increased focus on LTM gains. For example, in an influential paper, Logan (1988) defined automaticity not in terms of how little a given skill impinges upon limited cognitive resources, but rather in terms of how fully it is based on direct retrieval from LTM. Attentional processes thus affect skill-acquisition to the extent that these processes affect LTM.

A core concept within these models is a phenomenon that I introduced earlier: strategy transitions. Specifically, the transition from algorithmic performance to direct memory retrieval. However, from the perspective of some earlier models of skill acquisition, what may subjectively appear to be direct retrieval from memory might in fact be automatized algorithmic performance. That is, we are still using the algorithm to reach the answer, but performance of the algorithm is now taking place outside of conscious awareness, resulting in a subjective experience that may be indistinguishable from direct retrieval. How could we distinguish this from actual direct retrieval? Before describing the strategy-transition models further, then, it is important to first validate two core assumptions of these models: namely, that direct retrieval is retrieval, and that it is direct.

Unconscious algorithm use could potentially take two basic forms: unconscious processing of information (which would not be tied to specific stimuli), or unconscious representations of information (which could be tied to specific stimuli). In regard to the former possibility, some learning models from the early part of the modern skill-acquisition literature (e.g., Kolers, 1975; LaBerge & Samuels, 1974; Logan, 1978) placed a heavy, if not exclusive, emphasis on processes rather than representations. One representative example would be the LaBerge and Samuels (1974) model of reading. From the perspective of this model, learning to read involves the development of processes for combining textual information into larger units: features into letters, letters into words, words into sentences into meaning. With practice, these processes become automatized, and fall out of conscious awareness, but every newly presented piece of textual information is still processed upward from the most basic units of information, and thus transfer of reading skills to new and unfamiliar texts would be expected to be quite good. However, this class of models (Logan, 1988, referred to them as process-based models, but that is not a uniformly used term within the literature) ran into a very basic problem: learning tends to be quite specific in nature. For example, when receiving training on unfamiliar math problems (e.g., either young children receiving practice on simple multiplication problems, such as 6 x 7, or college-age individuals receiving training on multiplication problems such as 17 x 23), learning may not even transfer to operand order reversals (Reder & Ritter, 1992; Siegler, 1986). (E.g., for a child first learning multiplication, training on 6 x 7 does not transfer to 7 x 6.) When dealing with more familiar math problems (e.g., college students training on simple multiplication problems), observed improvements in response time (RT) with practice will transfer to operand order reversals, but not to related problems involving a change of operand or operator (Rickard & Bourne, 1996; Rickard, Healy, & Bourne, 1994). That is, training on 6 x 7 = __ will result in faster performance for 7 x 6 = __, but will not result in faster performance for 7 x __ = 42, or for

42 ÷ 6 = __, or for other related manipulations. Equivalent results have been found in a study that used words and sentences rather than arithmetic (Rickard & Bajic, 2006). That is, if subjects are taught sentences such as “snow falls gently”, then repeated practice recalling the word “gently” when presented with “snow falls ____” will result in faster recall of that specific word when presented with that specific sentence fragment. However, this practice does not result in faster recall of the word “falls” if subjects are later tested on “snow ____ gently” (Rickard & Bajic, 2006).

In regard to unconscious representation-based algorithms, a good hypothetical example involves something that people frequently fail to see as algorithms: namely, mnemonic devices. Suppose that a person uses a mnemonic device to recall some information, such as using an image of a bowl of corn flakes in order to remember the name Cornthwaite.[1] Initially, accessing the name Cornthwaite will require use of the algorithm—the mediating image of corn flakes. After time and practice, it will seem that the name Cornthwaite is recalled directly to memory, with no need to access the image of cereal. However, as Adams and McIntyre (1967) pointed out (see also Bellezza & Poplawsky, 1974; Bellezza, Poplawsky, & Aronovsky, 1977; Crutcher & Ericsson, 2000), even if a mnemonic device ceases to come to conscious awareness, this does not necessarily imply that a transition to direct, unmediated retrieval has taken place. Instead, it is possible that the original mnemonic pathway continues to be used, but at an automatic level outside of awareness. Crutcher and Ericsson (2000), who argued in favor of this viewpoint,

[1] This example is taken from an anecdote related by the columnist Jack Smith. As Smith (1994) noted, however, the woman who attempted to use this mnemonic device later referred to the actor, Robert Cornthwaite, as “Mr. Kellogg”.

referred to this unconscious use of old mnemonic representations as *covert mediation*.

How can we distinguish genuine direct retrieval (which completely bypasses the old mediating representation) from covert mediation? One technique introduced by Crutcher and Ericsson (2000) involved an interference manipulation. To relate the logic of this approach to the earlier example, if one learns to access the name Cornthwaite via an unconsciously accessed representation of corn flakes, then associating the image of corn flakes with a different name should make it more difficult to access the name Cornthwaite. Crutcher and Ericsson (2000) found evidence of such interference in a task that involved vocabulary-learning via a mnemonic device (the keyword method; see Raugh & Atkinson, 1975). However, Rickard and Bajic (2003) argued that conventional mnemonic devices could potentially add confounding variables to such an analysis. Instead, they recreated the basic interference paradigm using a simple two-link paired associate chain. A subject learned to associate a set of color-word stimuli with a set of spoken letter responses (e.g., see GREEN, and say "H"), and then learned to associate the set of letters with a set of spoken number responses (e.g., see H, and say "5"). Later, on each trial of an extensive training phase, the subject was presented with a color-word stimulus, and spoke the number associated with the corresponding letter mediator (e.g., see GREEN, think H, and say "5"). With sufficient training, of course, the numbers could be retrieved from memory without conscious use of the letter mediators. At that point, the interference manipulation was introduced: a subset of the letter mediators were associated with new responses (e.g., see H, and say "3"). Contrary to the covert mediation hypothesis, the interference manipulation did not increase the difficulty of speaking the appropriate number for each color-word stimulus. Thus, consistent with our subjective experience, direct retrieval is genuinely retrieval, and genuinely direct.

Although, as noted, there is strong evidence that skill acquisition for many classes of tasks involves a transition to a direct-retrieval strategy, there is not a general consensus regarding the exact mechanisms of this transition. To put these controversies within the proper context, it is important to first discuss another topic that has long held a central position within the skill acquisition literature: the time course of learning.

Practice Curves and the Power Law

Across a number of studies dealing with what appeared to be quite different sets of skills—e.g., rolling cigars (Crossman, 1959), tracing images seen in a mirror (Snoddy, 1926), proving geometric theorems (Neves & Anderson, 1981)—careful analysis of learning-rate data revealed a surprisingly common pattern: when practice curves were plotted (with the number of repetitions of some task on the x-axis, and the time needed to complete the task plotted on the y-axis), the speed-up in performance was initially rapid but then leveled off, along a curve that could generally be well fitted by a power function. Put another way, practice curves are generally straight lines when plotted in log-log coordinates. This log-log linearity of practice curves was first noted by Snoddy (1926), and later noted by De Jong (1957).

Crossman (1959), who developed and formalized the idea further, suggested that it could be called “De Jong’s Law”. However, this commonality across different types of skill-learning scenarios did not truly gain traction until it was rediscovered in a highly-influential meta-analysis by Newell and Rosenbloom (1981). Newell and Rosenbloom suggested two possible names for the phenomenon: (a) the *log-log linear learning law*, and (b) the less alliterative name by which it is now more commonly known: the *power law of practice*.

The tendency of speed-up with practice to approximate a power function has since been identified in an even wider range of skill domains, such as mental rotation (Kail, 1986), lexical decisions (Logan, 1988), recall of facts (Pirolli & Anderson, 1985), and social judgments (Smith & Lerner, 1986). However, there has been disagreement regarding exactly what the power law signifies in terms of the processes and representations that underlie skilled performance. Pirolli and Anderson (1985) argued that smooth, power function learning curves were indicative of performance improvements based on quantitative, rather than qualitative, changes. Logan (1988), however, demonstrated via stochastic modeling that smooth, power function speed-up can occur when there is a gradual transition between qualitatively different performance strategies.

Although there was disagreement regarding the power law’s implications, there was consensus regarding its importance. As Logan (1988) observed, power-function speed-up “is treated as a law, a benchmark prediction that theories of skill acquisition must make to be serious contenders…If they cannot account for the power law, they can be rejected immediately." However, as also noted by Logan (1992), virtually any model of skill acquisition can be modified to predict power function speed-up; particularly if (as is often true) the shape of the learning curve is a free parameter.

It was in this particular climate—great respect for the power law, but diminishing respect for the limited-capacity models of skill acquisition—that Logan (1988) introduced his *instance theory* of skill acquisition. From the perspective of instance theory, each encounter with a particular problem is encoded in memory as a distinct representation, or instance. When the problem is encountered again, a parallel, capacity unlimited race occurs between the algorithm and all previously encoded instances, with the winner of this race determining the response. As more instances accrue with practice, it becomes probabilistically more likely that at least one instance on each trial will have a finishing time that is faster than that of the algorithm, and thus the transition to memory-retrieval based performance occurs (Logan, 1988). Further, Logan demonstrated that a race process such as that embodied by his model could predict power function speed-up over the course of practice. Indeed, Logan (1992) argued that one advantage of instance theory, relative to other models of skill acquisition, was the fact that power function speed-up naturally emerged from its core assumptions. (For debate regarding the accuracy of this assertion, see Colonius, 1995, and Logan, 1995.)

Instance theory was followed by the *exemplar-based random walk* (EBRW) of Nosofsky and Palmeri (1997; see also Palmeri, 1997), which synthesized Logan's

(1988) model of skill acquisition with Nosofsky's (1986) *generalized context model* of categorization. For present purposes, the most important detail is that the EBRW model shares instance theory's core assumption that an unlimited-capacity parallel race takes place between the algorithmic strategy and the direct-retrieval strategy. Over the course of practice, this results in power function speed-up.

But there was a problem in the skill acquisition literature: the models were obeying the power law, but the data were not. That is, practice curves could *generally* be fairly well fit by power functions, but there were particular patterns of deviation from log-log linearity that were frequently appearing in data sets. For some tasks, power function fits would tend to underestimate RTs in the earlier part of practice, and overestimate RTs in the later parts of practice (Rickard, 1997). This observation helped set the stage for Rickard's (1997) *component power laws* (CMPL) model of skill acquisition. The CMPL model assumes that two independent retrievals from LTM can not occur simultaneously. Thus, for algorithms that consist of multiple retrievals from LTM[2]—such as the serial addition algorithm for multiplication problems—the algorithm-based strategy and the direct-retrieval-based strategy can not be executed simultaneously (Rickard, 1997; Rickard & Bajic, 2004).

Based on their core assumptions (see Logan, 1988, 1992), the race models predicted that power function speed-up would be observed not just in performance data averaged over items and subjects, but even in the practice curves observed for individual items. From the perspective of a strategy-selection model such as

[2] The CMPL model, as currently specified, does not make any predictions regarding algorithms with no LTM retrieval component, such as simple perceptual-motor algorithms.

CMPL,[3] however, the practice curves for individual items could more closely resemble a step function. That is, RTs for an individual item could show the following pattern: slow RTs in the earlier stages of practice, when performance is based solely on the algorithm; an abrupt drop in RT on the first trial in which direct-retrieval occurs; fast RTs thereafter, as performance is based solely on the direct-retrieval strategy. Since the exact point of transition to direct retrieval would occur essentially independently for each item (see Rickard, 2004), average RT data over items and subjects could result in a practice curve that approximates a power function, albeit with the deviations from log-log linearity that are often observed in practice data: RTs slower than the power function prediction in the earlier stages of practice (when use of the algorithmic strategy dominates for most or all items), and RTs faster than the power function prediction in the later stages of practice (when the direct-retrieval strategy dominates). Analysis on item-level data revealed precisely this pattern for some tasks (Rickard, 2004).

The once sacrosanct power law of practice has since come under fire even from researchers who do not share the core assumption of the CMPL model. In a meta-analysis of item-level RT data conducted by Heathcote, Brown, and Mewhort (2000), it was found that item-level practice curves could be better fit with an

[3] There are other models of skill acquisition (e.g., Siegler, 1988; Lemaire & Siegler, 1995) that share the CMPL model's assumption of a non-parallel strategy-selection process. However, these other models do not focus on the same types of issues that CMPL, instance theory, and the EBRW model routinely deal with, such as slight changes in RT for tasks that are already being performed at near-asymptotic accuracy.

exponential function rather than a power function.[4] When multiple exponential functions are averaged together (as would occur in typical aggregate data analysis), a power function can result as an artifact of the averaging (Myung, Kim, & Pitt, 2000; Estes, 1956; see also Stratton, Liu, Hong, Mayer-Kress, & Newell, 2007).

As Palmeri (1999) observed, "the power law may not be as lawful as was once thought." This initially created problems for the race models of skill acquisition, which had log-log linearity so closely bound to their core assumptions. However, Palmeri (1999) demonstrated that race models can account for deviations from log-log linearity, provided that some simplifying assumptions from earlier specifications of the race models are dropped (see Logan, 1988, 1992), and provided also that there are modified assumptions regarding the parameters of the distribution of retrieval RTs (Palmeri, 1999; see also Rickard, 2004).

Thus, some of the most basic issues of skill acquisition—such as whether or not two strategies can be executed at the same time—remain a source of ongoing controversy.

Moving Forward

The difficulty of fully resolving the above issues within the context of the current literature stems largely from three factors: two of them related to experimental design, and one related to data analysis.

[4] Heathcote et al. (2000) found even better fits using a hybrid power and exponential function, which they termed the APEX function. However, the APEX function requires an extra parameter relative to the exponential, so they recommended the exponential function as a simpler alternative.

Problem 1. The first of these issues cuts directly to a fundamental question: on any given trial, how do we know which strategy (or strategies) a subject is using? Until now, no one has developed a method for objectively identifying strategy use on each trial of a study.

Logan (1988) lamented that there were no established methods for identifying strategy use; but he suggested that one possible means of doing so would be through transfer tests. That is, after some amount of practice, new items would be introduced, and performance for these items would be compared with performance for the old, practiced items. For subjects who have transitioned to direct retrieval, this would be a comparison of fast direct retrieval performance vs. slower algorithm performance. For subjects who have not transitioned, it would be algorithm vs. algorithm, and little difference, if any, would be expected. However, even if the reasoning behind such an approach were valid, it would still provide only a rough, after-the-fact indication that some transition to direct retrieval had occurred. It would not pinpoint the exact moment that such a transition had occurred for each item, and it would not provide information about the relative time-course of these transitions across different items.

In Siegler (1988), elementary school children performed multiplication-based arithmetic problems, with optional use of scratch paper. The performance of these children was videotaped, and two raters later classified which strategy each child had used on each problem. However, as would be expected, inter-rater reliability was not perfect. Further, under the classification scheme used by Siegler, children were classified as having used retrieval for any problem during which no overt behavior (such as the use of scratch paper) was observed. As Siegler himself noted, such an approach introduces a risk that some algorithm-based solutions (e.g., a trial in which a child performs serial addition mentally, with no assistance from scratch paper) will be misclassified as retrieval trials (Siegler, 1988).

Haider and Frensch (2002) attempted to identify strategy transitions by monitoring for drops in RT among subjects performing alphabet arithmetic tasks.[5] After RTs had fallen below a pre-set level, subjects would receive a transfer test (i.e., testing with new items). However, their chosen RT criteria were (as they admitted) apparently somewhat arbitrary. Further, using this approach, they concluded that the transition to faster strategies essentially occurred at the same time for all items. However, this result has not been replicated in more tightly controlled studies of strategy shifts (e.g., Delaney et al., 1998; Rickard, 2004), which have found that the transition to the more-efficient strategy of direct-retrieval essentially occurs independently for all items. (Also, see Rickard, 2004, for a critique of their interpretation of the transfer test data.)

Delaney et al. (1998) presented a reanalysis of data from Reder and Ritter (1992) and Schunn et al. (1997). On each trial of these studies, the subject was presented with an arithmetic problem, and had 850 ms in which to indicate (via a

[5] In a typical alphabet arithmetic task (e.g., Compton & Logan, 1991; Logan & Klapp, 1991; Rickard, 2004), the subject would be presented with a problem such as “H + 3”, with the correct answer being the letter, “K”, that is 3 positions down the alphabet from H. Note that the task utilized by Haider and Frensch (2003) differed slightly from a typical alphabet arithmetic task, but only in terms of visual presentation; the underlying logic of the problems was the same.

button press) whether he or she wished to solve the problem by direct retrieval from memory (in which case, the subject would have approximately 1 s in which to do so), or through calculation (with a limit of approximately 18 s). Trials in which the subject answered the problem incorrectly were discarded, but trials in which the subject went over the time limit for answering the problem were retained. For each trial, the subject was classified as having used either direct retrieval or the algorithm, based on his or her choice during the initial 850 ms after stimulus onset. Overall, RTs for trials identified as retrieval trials were faster than those trials identified as algorithm trials, which suggests there is some validity to this approach. However, among subjects that had already successfully completed direct memory retrievals for some items, reversions to the use of the algorithm for those items appeared to occur more frequently than it did in studies that did not impose the 850 ms snap-judgment requirement (e.g., Rickard, 2004), thus suggesting possible classification errors associated with the requirements of the task. Further, an approach such as this is not ideal for studying the initial transition to direct retrieval. That is, indicating that subjects should only select the retrieval option when they believe they will be able to complete the retrieval quickly may be appropriate in the later stages of practice, during which retrieval can most likely be consistently completed more rapidly than the algorithm. However, we can not be certain that direct retrieval would be faster *the first time* it is attempted in a more naturalistic setting. Potentially, the first attempt at direct retrieval could be slower than some or all of the steps of the algorithm. An additional issue, which applies to all the strategy-identification methods discussed so far, is that they do not provide any means of identifying possible *partial-algorithm trials*—that is, trials in which a subject might begin the algorithm, but recall the answer directly before the algorithm has run to completion. (By way of analogy, a person using a word-processing application might begin moving the mouse pointer toward the dropdown menus, but then suddenly remember the shortcut key combination for the command that he or she wishes to use.)

A more common method of assaying strategy use is through retrospective reports, or strategy probes. That is, after each trial (e.g., the protocol group in Rickard & Bajic, 2003), or after each of a representative subset of trials (e.g., Rickard, 2004), or (less ideally) at the end of an experimental session (e.g., Logan & Klapp, 1991), a subject would be asked to explicitly identify which strategy (or strategies) he or she had used in the preceding trial (or trials). This would be done either by verbal report (e.g., Rickard, 1997; Rickard & Bajic, 2003; Romero, Rickard, & Bourne, 2006), or by a key-press response to a presented set of candidate strategies (e.g., Compton & Logan, 1991; Rickard, 2004). If a paradigm utilizing retrospective responses is properly designed (e.g., if subjects are not asked leading questions, and are not asked to speculate about cognitive processes of which they are not likely to have any conscious awareness), this approach can provide meaningful, veridical data (see Ericsson & Simon, 1993). Even with properly designed studies, though, this approach brings some risk of reactivity; that is, the act of repeatedly asking subjects about their strategy use may affect their choices regarding possible strategies. One way to check for this possibility is through between-group RT comparisons between subjects who were or were not instructed to provide strategy reports. However, studies that have used this approach have found inconsistent results (see discussion in Rickard, 2004). More research would be needed to clarify the source of these inconsistent results. In particular, there needs to be greater clarification regarding what would constitute an appropriate filler task for subjects who do not provide strategy reports. That is, a task that roughly matches the duration and (relatively minor) difficulty of a strategy report, but does not itself introduce any problems of reactivity (see Rickard, 2004).

Rickard (2004) combined retrospective reports with item-level RT analyses. Subjects performed an alphabet arithmetic task, and provided retrospective strategy reports via button-press on a subset of trials. These reports were later compared against RT data at the level of individual trials and items for each subject, and were found to be highly correlated with RT, such that reports of algorithm use were associated with slower RTs, and reports of direct retrieval were associated with faster RTs. For about 1.2% of trials, though, a strategy report was associated with a RT that was more characteristic of a different strategy, possibly indicating that the subject had misclassified his or her strategy.

Of all the strategy-identification methods described thus far, retrospective reports appear to be the most reliable overall. However, as noted, they bring a risk of reactivity—the inquiry might alter the results. Also, such data may only be reliable at a broad, aggregate level. Strategy-identification data gained through this method may be correct *most* of the time, but if we select a specific trial for a specific subject, we can not be certain which strategy was used for that specific trial—either because it may be a trial for which no retrospective data was collected (in a design for which such data is only collected for a subset of trials), or because the subject may have accidentally (or deliberately) misidentified which strategy he or she had used for that trial.

Conceivably, retrospective reports could be used to investigate partial-algorithm trials, but so far the efforts to do so have not been promising. In Compton and Logan (1991), subjects made strategy reports by a key-press selection from a set of candidate answers. In addition to being able to report direct retrieval or the algorithm, some conditions allowed subjects to choose from a set of more subtle possibilities, such as starting one strategy and then switching to the other. However, only the basic direct retrieval and algorithm reports appear to have been reliable; the percentage of reports in other response categories fluctuated depending on what subjects were asked (Compton & Logan, 1991). This result is not surprising, since it is known that retrospective reports become unreliable when subjects are asked to speculate about subtle cognitive processes (Ericsson & Simon, 1993).

Thus, the effort to find better strategy-identification methodologies is an ongoing topic of research, and any improvements in reliability and validity are a useful contribution to the literature. In Chapter 2, I introduce an original paradigm that permits the collection of objective data regarding strategy use in every trial, with no need to ask subjects for retrospective reports (e.g., Romero et al., 2006), and no need to make indirect inferences about strategy use (e.g., Siegler, 1988). In addition, the paradigm that I introduce is the first that permits the systematic investigation of partial-algorithm trials.

Problem 2. The second ongoing problem within the skill literature is the fact that, in current experimental designs, RT data is only collected for the final answer. Such designs provide no means of indexing the latency of each individual step of a given algorithm. As such, significant and consistent latency changes that uniquely affect a single algorithm step may be concealed within the overall RT data. Consider again the example of a person who can either engage in a visuo-motor search for a room in a building, or simply remember the location. When seeking the room the first time, this individual might walk down a hallway while searching for the room. The second time, though, this individual might attempt to recall the location, fail in this attempt, and then hurry down the hallway during his search, in an effort to recover lost time. In such a case as this, there may be little change in the overall time required to locate the room, yet a change in strategic processing has clearly occurred—something that would only be revealed by data indicating how long this individual paused prior to making his first step down the hallway. Thus, a lack of latency data for individual steps of the algorithm hinders identification of the exact ways in which strategy use changes over time. In Chapters 2 and 3, I introduce a new approach that allows the latency of each individual algorithm step to be measured.

Problem 3. The third issue, as noted earlier, relates to data analysis. Consider a hypothetical experiment that involves multiple problems (items) for which a transition from algorithm to direct retrieval is possible; and suppose that these items are presented repeatedly over multiple blocks of practice. Because the shift to direct retrieval is known to occur roughly independently for each item (Rickard, 2004), it is vanishingly unlikely that any subject would simultaneously transition to direct-retrieval for all items. Thus, a given block may include some items for which the subject still uses the algorithmic approach (with a relatively slow overall RT), some items for which the subject is performing her first accurate direct retrievals for the corresponding answers (faster overall RTs; a qualitative improvement relative to the algorithm), and some items for which the subject has transitioned to direct retrieval several blocks earlier (even faster RTs, due to practice; a quantitative improvement relative to earlier retrieval trials). In an analysis involving aggregate data, how can we prevent the mixing of items at different stages of transition, and different levels of practice, from obscuring effects that may only exist for items belonging to one specific state of transition, or one specific level of practice? In Chapter 2, I introduce a data-analysis approach that makes it possible to appropriately and effectively aggregate data, even when the learning curves for individual items are not naturally synchronized.

In addition to finding solutions for the methodological and data analytic issues outlined above, an additional goal of the present work is to explore and compare two general classes of tasks that exhibit the shift to retrieval: namely, tasks with algorithms that require the retrieval of information from LTM, and tasks with algorithms that do not require LTM retrieval—i.e., those with simple perceptual-motor algorithms. The latter class of tasks is very common in everyday life (as evidenced by the amount of time individuals spend visually searching for objects, such as cars and keys, whose locations they could instead have remembered directly); however, this class of tasks has been underserved within the current skill acquisition literature. Studies of skill acquisition in perceptual-motor tasks (e.g., Snoddy, 1926; Stratton et al., 2007) tend to focus specifically on how people get better at perceptual-motor performance, not on how they could potentially skip the perceptual-motor performance entirely through the use of a LTM-retrieval strategy. For researchers interested in strategies based on LTM retrieval (e.g., Delaney et al., 1998; Jenkins & Hoyer, 2000; Logan, 1988, 1992; Onyper et al., 2006; Palmeri, 1997; Rawson, 2004; Reder & Ritter, 1992; Rickard, 1997, 2004, 1999; Rickard & Bajic, 2003, 2006; Rogers et al., 2000; Schunn et al., 1997), the algorithm typically consists of a different set of LTM retrievals. Considerably less attention has been given to tasks in which use of a perceptual-motor strategy transitions to use of a direct LTM-retrieval strategy (cf. Touron et al., 2001). There has not (until now) been a study that directly compares performance between two tasks with algorithms designed to be identical in all ways *except* in regard to whether LTM retrieval was needed.

In Experiments 1 and 2 of Chapter 3, I explore the dynamics of performance for tasks with simple perceptual-motor algorithms, including (for the first time) a direct, controlled comparison of performance for tasks with LTM-retrieval-based algorithms versus tasks with perceptual-motor algorithms.

Before going further, it would be worthwhile to note a philosophical perspective guiding much of this work. In all of the following experiments, the retrieval-based algorithms and the perceptual-motor algorithms have been designed to be as simple as possible—involving such basic components as counting and tapping. If one were to instead use algorithms based on more difficult operations, and one were to find that people can not execute two complex strategies at the same time, it would still be an open question as to whether they could have executed two simple strategies simultaneously. That is, the failure at simultaneous performance could indicate some structural limit in human cognition, or it could represent some capacity limit related to insufficiently practiced tasks, or it could be an artifact of some confounding variable introduced by (or obscured by) the complexity of the task. But if we instead use very simple algorithms, and we still find that people are unable to execute two very simple strategies simultaneously, then it would strongly suggest that they would have at least as much difficulty—if not much more difficulty—in simultaneously executing two complex strategies. For this reason, by using strategies based on very simple (and seemingly automatic) algorithms (counting, tapping, clicking) it can help us to identify the most basic, fundamental limits of human skilled performance—the bare minimum of what individuals can or can not accomplish with repeated practice of a task. It is from this perspective that I've chosen to use algorithms based on very simple activities—highly-familiar, highly-trained, essentially automatic—to explore the temporal dynamics of strategy execution in cognitive skill learning.

References

Adams, J. A., & McIntyre, J. S. (1967). Natural language mediation and all-or-none learning. *Canadian Journal of Psychology*, *21*(5), 436-449.

Bellezza, F. S., & Poplawsky, A. J. (1974). The function of one-word mediators in the recall of word pairs. *Memory & Cognition*, *2*(3), 447-452.

Bellezza, F. S., Poplawsky, A. J., & Aronovsky, L. A. (1977). The functional role of one-word mediators. *Bulletin of the Psychonomic Society*, *10*(6), 460-462.

Besner, D. (2001). The myth of ballistic processing: Evidence from Stroop's paradigm. *Psychonomic Bulletin & Review*, *8*(2), 324-330.

Brown, I. D., & Poulton, E. C. (1961). Measuring the spare "mental capacity" of car drivers by a subsidiary task. *Ergonomics*, *4*, 35-40.

Chase, W. G., & Simon, H. A. (1973). Perception in chess. *Cognitive Psychology*, *4*(1), 55-81.

Colonius, H. (1995). The instance theory of automaticity: Why the Weibull? *Psychological Review*, *102*(4), 744-750.

Compton, B. J., & Logan, G. D. (1991). The transition from algorithm to retrieval in memory-based theories of automaticity. *Memory & Cognition*, *19*(2), 151-158.

Crossman, E. R. F. W. (1959). A theory of the acquisition of speed-skill. *Ergonomics*, *2*, 153-166.

Crutcher, R. J., & Ericsson, K. A. (2000). The role of mediators in memory retrieval as a function of practice: Controlled mediation to direct access. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *26*(5), 1297-1317.

De Groot, A. D. (1965). *Thought and choice in chess*. Amsterdam University Press.

De Jong, J. R. (1957). The effects of increasing skill on cycle time and its consequences for time standards. *Ergonomics*, *1*, 51-60.

Delaney, P. F., Reder, L. M., Staszewski, J. J., & Ritter, F. E. (1998). The strategy-specific nature of improvement: The power law applies by strategy within task. *Psychological Science*, *9*(1), 1-7.

Ericsson, K. A., & Simon, H. A. (1993). *Protocol analysis: Verbal reports as data*. Cambridge, MA, US: The MIT Press.

Estes, W. K. (1956). The problem of inference from curves based on group data. *Psychological Bulletin*, *53*(2), 134-140.

Haider, H., & Frensch, P. A. (2002). Why aggregated learning follows the power law of practice when individual learning does not: Comment on Rickard (1997, 1999), Delaney et al. (1998), and Palmieri (1999). *Journal of Experimental Psychology: Learning*, (28), 392-406.

Hasher, L., & Zacks, R. T. (1979). Automatic and effortful processes in memory. *Journal of Experimental Psychology: General*, *108*(3), 356-388.

Heathcote, A., Brown, S., & Mewhort, D. J. K. (2000). The power law repealed: The case for an exponential law of practice. *Psychonomic Bulletin & Review*, *7*(2), 185-207.

Hertzog, C., Touron, D. R., & Hines, J. C. (2007). Does a time-monitoring deficit influence older adults' delayed retrieval shift during skill acquisition? *Psychology and Aging*, *22*(3), 607-624.

Jenkins, L., & Hoyer, W. J. (2000). Instance-based automaticity and aging: Acquisition, reacquisition, and long-term retention. *Psychology and Aging*, *15*(3), 551-565.

Kail, R. (1986). The impact of extended practice on rate of mental rotation. *Journal of Experimental Child Psychology*, *42*(3), 378-391.

Kolers, P. A. (1975). Memorial consequences of automatized encoding. *Journal of Experimental Psychology: Human Learning and Memory*, *1*(6), 689-701.

LaBerge, D., & Samuels, S. J. (1974). Toward a theory of automatic information processing in reading. *Cognitive Psychology*, *6*(2), 293-323.

Lemaire, P., & Siegler, R. S. (1995). Four aspects of strategic change: Contributions to children's learning of multiplication. *Journal of Experimental Psychology: General*, *124*(1), 83-97.

Logan, G. D. (1988). Toward an instance theory of automatization. *Psychological Review*, *95*(4), 492-527.

Logan, G. D. (1992). Shapes of reaction-time distributions and shapes of learning curves: A test of the instance theory of automaticity. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *18*(5), 883-914.

Logan, G. D. (1980). Attention and automaticity in Stroop and priming tasks: Theory and data. *Cognitive Psychology*, *12*(4), 523-553.

Logan, G. D. (1978). Attention in character-classification tasks: Evidence for the automaticity of component stages. *Journal of Experimental Psychology: General*, *107*(1), 32-63.

Logan, G. D. (1995). The Weibull distribution, the power law, and the instance theory of automaticity. *Psychological Review*, *102*(4), 751-756.

Logan, G. D., & Klapp, S. T. (1991). Automatizing alphabet arithmetic: I. Is extended practice necessary to produce automaticity? *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *17*(2), 179-195.

McDaniel, M. A., & Einstein, G. O. (2000). Strategic and automatic processes in prospective memory retrieval: a multiprocess framework. *Applied Cognitive Psychology*, *14*(7), S127-S144.

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.

Myung, I. J., Kim, C., & Pitt, M. A. (2000). Toward an explanation of the power law artifact: Insights from response surface analysis. *Memory & Cognition*, *28*(5), 832-840.

Neves, D., & Anderson, J. (1981). Knowledge compilation: Mechanisms for the automatization of cognitive skills. In *Cognitive skills and their acquisition* (pp. 57-84). Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Nosofsky, R. M. (1986). Attention, similarity, and the identification-categorization relationship. *Journal of Experimental Psychology: General*, *115*(1), 39-57.

Nosofsky, R. M., & Palmeri, T. J. (1997). An exemplar-based random walk model of speeded classification. *Psychological Review. Vol 104(2)*, 266-300.

Onyper, S. V., Hoyer, W. J., & Cerella, J. (2006). Determinants of retrieval solutions during cognitive skill training: Source confusions. *Memory & Cognition*, *34*(3), 538-549.

Palmeri, T. J. (1997). Exemplar similarity and the development of automaticity. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *23*(2), 324-354.

Palmeri, T. J. (1999). Theories of automaticity and the power law of practice. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *25*(2), 543-551.

Pirolli, P. L., & Anderson, J. R. (1985). The role of practice in fact retrieval. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *11*(1), 136-153.

Raugh, M. R., & Atkinson, R. C. (1975). A mnemonic method for learning a second-language vocabulary. *Journal of Educational Psychology*, *67*(1), 1-16.

Rawson, K. A. (2004). Exploring automaticity in text processing: Syntactic ambiguity as a test case. *Cognitive Psychology*, *49*(4), 333-369.

Redelmeier, D. A., & Tibshirani, R. J. (1997). Association between cellular-telephone calls and motor vehicle collisions. *New England Journal of Medicine*, *336*(7), 453-458.

Reder, L. M., & Ritter, F. E. (1992). What determines initial feeling of knowing? Familiarity with question terms, not with the answer. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *18*(3), 435-451.

Rickard, T. C. (2004). Strategy execution in cognitive skill learning: An item-level test of candidate models. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *30*(1), 65-82.

Rickard, T. C. (1997). Bending the power law: A CMPL theory of strategy shifts and the automatization of cognitive skills. *Journal of Experimental Psychology: General*, *126*(3), 288-311.

Rickard, T. C. (1999). A CMPL alternative account of practice effects in numerosity judgment tasks. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *25*(2), 532-542.

Rickard, T. C., & Bajic, D. (2003). Automatic mediation or absence of mediation? Commentary on Crutcher and Ericsson (2000). *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *29*(6), 1381-1386.

Rickard, T. C., & Bajic, D. (2006). Cued recall from image and sentence memory: A shift from episodic to identical elements representation. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *32*(4), 734-748.

Rickard, T. C., & Bajic, D. (2004). Memory retrieval given two independent cues: Cue selection or parallel access? *Cognitive Psychology*, *48*(3), 243-294.

Rickard, T. C., & Bourne, L. E. (1996). Some tests of an identical elements model of basic arithmetic skills. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *22*(5), 1281-1295.

Rickard, T. C., Healy, A. F., & Bourne, L. E. (1994). On the cognitive structure of basic arithmetic skills: Operation, order, and symbol transfer effects. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *20*(5), 1139-1153.

Rogers, W. A., Hertzog, C., & Fisk, A. D. (2000). An individual differences analysis of ability and strategy influences: Age-related differences in associative learning. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *26*(2), 359-394.

Romero, S. G., Rickard, T. C., & Bourne, L. E. (2006). Verification of multiplication facts: An investigation using retrospective protocols. *American Journal of Psychology*, *119*(1), 87-120.

Salthouse, T. A. (1986). Perceptual, cognitive, and motoric aspects of transcription typing. *Psychological Bulletin*, *99*(3), 303-319.

Schneider, W., Dumais, S. T., & Shiffrin, R. M. (1984). Automatic and control processing and attention. In *Varieties of attention* (pp. 1-27). New York: Academic Press.

Schunn, C. D., Reder, L. M., Nhouyvanisvong, A., Richards, D. R., & Stroffolino, P. J. (1997). To calculate or not to calculate: A source activation confusion model of problem familiarity's role in strategy selection. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *23*(1), 3-29.

Shiffrin, R. M., & Dumais, S. T. (1981). Development of automatism. In *Cognitive skills and their acquisition* (pp. 111-140). Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Shiffrin, R. M., & Schneider, W. (1977). Controlled and automatic human information processing: II. Perceptual learning, automatic attending and a general theory. *Psychological Review*, *84*(2), 127-190.

Siegler, R. S. (1988). Strategy choice procedures and the development of multiplication skill. *Journal of Experimental Psychology: General*, *117*(3), 258-275.

Siegler, R. S. (1986). *Children's thinking*. Englewood Cliffs, NJ: Prentice-Hall.

Smith, E. R., & Lerner, M. (1986). Development of automatism of social judgments. *Journal of Personality and Social Psychology*, *50*(2), 246-259.

Smith, J. (1994, February 28). In memory of all those name games. *Los Angeles Times*.

Snoddy, G. S. (1926). Learning and stability: a psychophysiological analysis of a case of motor learning with clinical applications. *Journal of Applied Psychology*, *10*(1), 1-36.

Stratton, S. M., Liu, Y., Hong, S. L., Mayer-Kress, G., & Newell, K. M. (2007). Snoddy (1926) revisited: Time scales of motor learning. *Journal of Motor Behavior*, *39*(6), 503-515.

Touron, D. R., Hoyer, W. J., & Cerella, J. (2001). Cognitive skill acquisition and transfer in younger and older adults. *Psychology and Aging*, *16*(4), 555-563.

Chapter 2

The Temporal Dynamics of Strategy Execution in Cognitive Skill Learning

The Temporal Dynamics of Strategy Execution in Cognitive Skill Learning

Daniel Bajic and Timothy C. Rickard
University of California, San Diego

The transition from algorithmic to memory-based performance is a core component of cognitive skill learning. There has been debate about the temporal dynamics of strategy execution, with some models assuming a race (i.e., independent, capacity unconstrained parallel processing) between algorithm and retrieval, and others assuming a choice mechanism. The authors investigated this issue using a new approach that allows the latency of each algorithm step to be measured, in turn providing new insight into (a) whether there is slowing of 1 or more algorithm steps on trials immediately preceding the 1st retrieval trial for an item, as might be expected if there is a competitive strategy execution process of some type other than a race, and (b) whether there is partial algorithm completion on retrieval trials, as would be expected if the 2 strategies are executed in parallel. Results are uniquely consistent with a strategy choice mechanism involving a competition between the retrieval strategy and the 1st step of the algorithm.

Keywords: skill, learning, cued recall, strategy execution, strategy choice

An important component of cognitive skill learning—indeed, arguably the signature learning event for many tasks—is the shift from initial use of a slow, multistep *algorithm* to a faster and subjectively less effortful memory look-up of the answer (*direct retrieval*). The classic example is arithmetic learning. In doing single-digit multiplication, children may initially perform a repeated addition algorithm, but with sufficient practice they will transition to direct retrieval (e.g., Siegler, 1988). Multiple laboratory studies have confirmed the ubiquity of this shift for arithmetic and arithmetic-like tasks (Delaney, Reder, Staszewski, & Ritter, 1998; Jenkins & Hoyer, 2000; Logan, 1988, 1992; Onyper, Hoyer, & Cerella, 2006; Palmeri, 1997; Reder & Ritter, 1992; Rickard, 1997, 1999, 2004; Rogers, Hertzog, & Fisk, 2000; Schunn, Reder, Nhouyvanisvong, Richards, & Stroffolino, 1997; Touron, Hoyer, & Cerella, 2001, 2004).

Similar shifts from algorithmic (defined broadly) to retrieval-based performance are believed to occur in a wide variety of nonarithmetic domains, including recall from episodic memory (Rickard & Bajic, 2006), the shift from mnemonically mediated to unmediated memory retrieval (Kole & Healy, 2007; Rickard & Bajic, 2003), lexical decision (Logan, 1988), word reading (e.g., Tao & Healy, 2002), and text comprehension (Rawson, 2004). Similar shifts may occur under item repetition conditions for visuospatial tasks such as mental rotation (Kail, 1986). A reasonable argument can be made, in fact, that any efficiently executed cued recall is a consequence of this shift.

Recent efforts to characterize the mechanism underlying this shift make diametrically opposing claims about the dynamics of strategy execution on each trial. One class of models, exemplified by the instance theory of automaticity (Logan, 1988) and its theoretically allied successor, the exemplar-based random walk (EBRW) model (Nosofsky & Palmeri, 1997; Palmeri, 1997), assumes a straightforward race between the two strategies: Both strategies are attempted on each trial, and the finishing time for each strategy is unaffected by competition with the other strategy. The direct retrieval process that is initiated at the start of the trial is assumed to continue throughout algorithm execution even for complex algorithms involving multiple steps.

In this class of models, instance representations are assumed to support direct retrieval. With each repetition of a given item, a new memory instance is encoded. These instances (each with its own distribution of finishing times) race with one another and with the algorithm during each trial. As more instances accrue with practice, there is an increased probability that retrieval of at least one of the instances will beat the algorithm. The EBRW model elaborates on the instance theory by assuming that the instance retrieval feeds into a random walk retrieval process that races with the algorithm.

A second class of models assumes that only one strategy can be executed at any given moment (Rickard, 1997, 2004; Schunn et al., 1997; Siegler, 1988; for a more general theoretical framework that is consistent with this assumption, see Byrne & Anderson, 2001). Siegler's (1988) distribution of associations model assumes that retrieval is always attempted first, with the algorithm serving as a back-up strategy. The Schunn et al. (1997) source activation confusion (SAC) model, which builds upon Reder's (1987, 1988) work on question answering, focuses on the feeling of knowing phenomenon as a mechanism of strategy choice and on factors that affect feeling of knowing. Rickard's (1997) component power laws (CMPL) model provides perhaps the most natural framework, for the current purposes, within which to draw predictive comparisons between choice and race models, and it is thus the focus of the following development. We consider the other choice models further in the Discussion section.

Daniel Bajic and Timothy C. Rickard, Department of Psychology, University of California, San Diego.

We would like to thank Elena Eiden, Jazmin Morales, Laura Muzzatti, Laura Osborne, and Wang Pan for their assistance with data collection.

Correspondence concerning this article should be addressed to Timothy C. Rickard, Department of Psychology, University of California, San Diego, 9500 Gilman Drive, La Jolla, CA 92093. E-mail: trickard@ucsd.edu

The CMPL model treats the algorithm (e.g., the repeated addition algorithm for multiplication) as a sequence of memory retrieval steps (henceforth referred to as algorithm steps) that tap the same memory retrieval system that is used to execute the direct retrieval strategy—a memory system that is in turn assumed to be the same as that used in explicit cued recall. The model assumes a memory retrieval bottleneck such that only one retrieval can be completed at a time (for supporting evidence in the case of cued recall, see Nino & Rickard, 2003). Thus, for any algorithm that involves one or more retrievals from long-term memory, the two strategies cannot be executed in parallel.

Strategy choice in the model is based on a brief parallel competition between two interpretations of the stimulus at the beginning of each trial (i.e., between the "problem-level" nodes in the Rickard, 1997, simulation model). In one interpretation, the stimulus is treated as a cue for executing the algorithm first step. In the other, it is treated as a cue for executing the direct retrieval strategy. The stimulus interpretation that first reaches an activation threshold is selected and the memory retrieval(s) for the corresponding strategy are executed. The other strategy is aborted and undergoes no further processing on that trial. The retrieval strategy becomes gradually more competitive over trials (through a strengthening process rather than instance accrual) until it eventually wins the competition with the algorithm.

To a close approximation, the CMPL model instantiates a race to determine which stimulus interpretation will be selected for further processing.[1] That is, choice processing adds no latency component to the total time to finish the trials, regardless of the strategy that is selected. An alternative and perhaps more viable version of the model would incorporate a time-consuming choice process, such that the competing but unsuccessful strategy slows the selection time of the winning strategy in proportion to its competitiveness (i.e., its "strength" relative to the winning strategy). These two hypothetical cases constitute what are referred to as *zero latency* and *positive latency* choice processes, respectively.

Two additional strategy execution dynamics that have not been considered formally in simulation models to date also merit consideration. The first of these is a modified version of the EBRW or instance models in which there is capacity-limited parallel strategy execution rather than a race (for related work see Navon & Miller, 2002; Tombu & Jolicoeur, 2003). As noted earlier for the race models, the retrieval process that is initiated at the start of the trial would be assumed to continue throughout algorithm execution. This model would predict that processing of one or both strategies will be slowed when the two strategies are in competition.

The second possibility, consistent with both parallel and choice models, is that subjects might execute the algorithm as a check on retrieval accuracy on one or more trials before gaining sufficient confidence to rely exclusively on retrieval for that item. In the case of choice models, strategy execution on such trials would be sequential—retrieval followed by the algorithm.

An Overview of the Empirical Evidence Bearing on the Models

As developed to date, race models assume that the retrieval strategy comes to dominate the algorithm gradually over multiple trials as the most quickly retrieved instance becomes probabilistically more likely to beat the algorithm. These models were originally designed to account for (among other things) smooth, power function response time (RT) speedup while making the novel predictions of power function reduction in the standard deviation (*SD*) and matched learning rate parameters for the RT and *SD* functions.

Rickard (1997, 1999, 2004) showed that, at least for some tasks, those predictions do not hold. Instead, patterns are consistent with what were novel predictions of the CMPL model: First, speedup in mean RT does not follow a power function; rather, there are separate power functions with different parameter values that govern speedup for the algorithm and retrieval trials.[2] Second, *SD* does not decrease as a power function; instead, separate power functions with different parameter values govern reductions in *SD* for the algorithm and retrieval trials. Third, due to the strategy mixture effect over items that is implied by the CMPL model, overall *SD* can, under some circumstances, reach its maximum value at roughly the point during training wherein about 50% of the trials involve memory retrieval; and fourth, RT speedup curves for individual items for each subject can exhibit an abrupt (step-function) RT reduction at the point of the strategy shift (Rickard, 2004; for related work see Haider & Frensch, 2002). The CMPL model predicts this step-function RT reduction provided that retrieval is executed more quickly than the algorithm at the point during training wherein the strategy shift occurs. (As a simplifying assumption, it has been assumed in CMPL model fits to date that once the shift to retrieval occurs for an item, the retrieval strategy is used for every subsequent trial. This assumption is not critical to the CMPL model, and it is not a requirement of choice models generally.) Step-function RT reduction is not expected in the averaged data because the strategy shift is not expected to occur on the same trial for every item.

The first three results outlined above might be explainable by the race models through modification of assumptions about how parameters differ over items (Palmeri, 1999). Rickard (2004), however, argued that the fourth result above is not consistent with any race model that assumes a gradual and probabilistic replacement of algorithm by retrieval over many trials, because gradual strategy replacement predicts a smooth, continuous speedup in expected RT value even at the item level.

Race models are nevertheless still viable, for two reasons. First, step-function RT drops have to date been demonstrated for only one task (alphabet arithmetic; Rickard, 2004). Second, even the step-function RT drops can be explained by a race model if the gradual strategy replacement assumption in those models is dropped. Suppose that for the first $n - 1$ trials for a given item, the algorithm is

[1] In the Rickard (1997) simulation, selection of the algorithm stimulus interpretation is slowed by an inhibitory connection from the retrieval path as the retrieval strategy becomes more competitive. Due to the properties of the activation functions in that model, however, completion time for the algorithm first step is unaffected by the competition from the retrieval strategy and vice versa, constituting a race dynamic.

[2] The CMPL model also predicts power law speedup for each strategy at the item level. Race models make the same prediction for retrieval speedup. Heathcote, Brown, and Mewhort (2000) showed that the exponential function fits slightly better than the power function to item-level data, calling these predictions into question. However, the CMPL prediction of item-level power function speedup is not central to either the CMPL architecture or to choice models generally, and this issue is not pertinent to the focus of this article.

executed, and that no memory instance (or only a very weak memory instance) is encoded. On trial *n*, the algorithm is again executed and a strong memory instance that can support fast direct retrieval on subsequent trials is finally encoded. From trial $n + 1$ onward, direct retrieval will win the race with the slower algorithm, potentially resulting in a step-function RT drop at the strategy shift point.

Indexing the Latency of Each Algorithm Step

In the current experiment, we attempted to gain more theoretical leverage by using a task that allows indexing of not only the RT for each trial (defined as the latency between stimulus onset and vocal response execution) but also the latency for completion of each step of the algorithm. On each trial of the experiment, subjects saw a two-digit number and were instructed to count forward from that number, pressing the space bar in synchrony with each count, until the computer informed them to stop and to speak the number to which they had counted. For each stimulus number (e.g., 21) the same number of counts was always required (say, 11), and the same response was always to be spoken (e.g., 32). Each stimulus was presented multiple times over training blocks. If subjects remembered the answer at any point during a trial, they could end the trial prior to completing the algorithm by speaking that answer. Each keypress recorded the approximate latency of each counting step, and a microphone voice key recorded the RT.

This task design allows us to explore two previously unaddressed questions about the temporal dynamics of strategy execution: First, on the last few algorithm trials preceding the first correct retrieval trial for an item, is there evidence of progressively slower execution times for one or more algorithm steps, as might be expected if the retrieval strategy becomes more competitive over trials and if there is a latency-consuming strategy competition (i.e., limited capacity parallel processing or positive latency choice)? And second, on retrieval trials (i.e., trials on which the answer is spoken prior to completion of the algorithm), is there evidence that some fraction of the algorithm steps are completed, as would be expected if the two strategies race?

Method

Subjects

Forty-one University of California at San Diego undergraduate students participated for course credit.

Materials, Apparatus, and Procedures

Subjects were tested individually on IBM-compatible personal computers, with each subject seated approximately 50 cm from the computer screen and approximately 3 cm from a microphone. The computer keyboard was positioned directly behind the microphone, such that the subject could comfortably place one hand over the space bar; the experimenter was seated to the right of the subject, with access to the keyboard's number pad. The experiment was created with E-Prime software (Psychology Software Tools, Pittsburgh, PA) and the accompanying voice-key apparatus (Model 200A).

The experiment consisted of a warm-up phase and a training phase. Prior to each phase, instructions were presented on the screen and were also read aloud by the experimenter. Within each trial of each phase, a stimulus would be presented visually, an algorithmic solution (if used) would consist of silent counting accompanied by concurrent keypresses, and the final response provided by the subject would be generated vocally. The Appendix lists all visual stimulus and vocal response items used in the training phase. The warm-up phase utilized the same values, each raised by 10 (e.g., 30 and 44 in the warm-up phase vs. 20 and 34 in the training phase, etc.). In the description below, a *block* is defined as one randomly ordered presentation for each of the 10 possible stimulus-response items, with each item therefore having a mean repetition lag of 10 trials across blocks.

The warm-up phase consisted of a single block. At the start of each trial, the screen went blank for 500 ms, a fixation field (consisting of three plusses) was presented at the center of the screen for 500 ms, the screen again went blank for 500 ms, and then a two-digit number—the trial stimulus—was presented at the center of the screen. Subjects were instructed to count silently upward from the presented number, pressing the space bar once with each count, until the word *STOP* was presented on the screen. At that point, the subject was to speak his or her answer—the number that he or she had counted up to—into the microphone. The stimulus remained on the screen during the keypresses and was replaced by the word *STOP* after the number of keypresses for that trial was equal to the value of the correct response minus the numerical stimulus, thus ranging from a minimum of 9 keypresses to a maximum of 15.

After the subject provided a vocal response, the experimenter entered the subject's response and recorded whether the voice key tripped properly. If the subject was in error (as might occur if a rapidly counting subject overshot the final answer), the correct response was presented for 5 s. Otherwise, the word *Correct!* was presented for 800 ms. Immediately following feedback, the next trial began.

The training phase of the study was identical to the warm-up phase, with the following exceptions. Multiple blocks were presented, and subjects were informed that the same set of starting numbers (stimuli) would be presented repeatedly throughout the phase, with each starting number always having the same final number. Subjects were informed that they therefore had two methods that could be used to find the correct answer for each trial: (a) counting upward from the starting number, tapping the space bar once with each count until the word *STOP* appeared on the screen, and (b) remembering the answer associated with the starting number for that trial and speaking the answer into the microphone without doing all of the keypresses. To promote parallel strategy execution, if such parallelism is possible, the instructions stated (falsely) that "Many subjects report good results when they attempt to use both strategies at the same time." Subjects were told that they could speak the answer into the microphone at any time during each trial. They were instructed that they should try to finish this part of the experiment as quickly as possible while still being accurate.

In this phase, each trial stimulus was removed from the screen either when the subject spoke an answer or when the subject had entered a sufficient number of keypresses to bring the word *STOP* onto the screen—whichever came first. Subjects were permitted a brief pause between each block and continued to receive new blocks until 45 min from the start of this phase, after which the experiment concluded and the subject was debriefed.

Results

Prior to analysis, data from 3 subjects were discarded due to unusually low accuracy (<80%) and data from 1 subject were discarded due to unusually frequent voice-key errors. All analyses reported below are for the training phase data of the remaining 37 subjects.

Voice-key errors, which occurred in 6% of trials, were removed prior to analysis. Mean accuracy on the first training block was 90.5%, increasing to 95.6% by the 23rd block, the furthest block that all 37 subjects completed.

The mean of the subject-level mean correct RTs (latency from stimulus presentation to vocal response) is plotted as a function of training block in Figure 1A, and the mean of the subject-level *SD*s is plotted in Figure 1B. Best fitting three-parameter power functions are also included in the figures for reference. The observed pattern of deviation from power function improvement in both cases is consistent with that observed in prior studies of tasks that exhibit the shift from algorithm to retrieval (Rickard, 1997, 1999). Of particular note, the pattern of increasing *SD*s over the first few blocks has been observed previously for both numerosity judgment (Rickard, 1999) and alphabet arithmetic (Wagenmakers & Brown, 2007), the two tasks most studied to date.

The proportion of trials in which subjects retrieved the answer (defined as those trials in which the subject spoke the answer before completing all algorithm steps) is shown as a function of training block in Figure 2. The strategy shift was about 80% complete by the 23rd block, roughly in accord with findings of studies in which strategy probes were used (e.g., Rickard, 1997). Both Figure 2 and the peak of the *SD* curve in Figure 1B indicate that the shift to retrieval had occurred for 50% of the items by about Block 9.

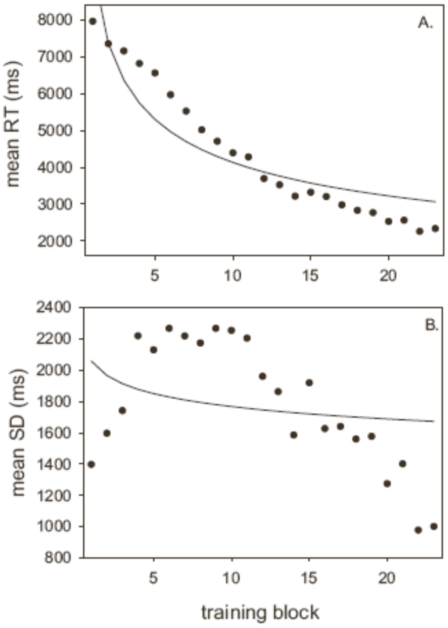

Figure 1. Mean response time (A) and mean standard deviation (B) as functions of training block for the first 23 blocks (to which all subjects contributed) along with the best fitting three-parameter power functions. RT = response time.

There were 57 items over 9 subjects (15% of items) that exhibited no shift to retrieval. Mean latencies for these items decreased from about 8,000 ms on the first block to about 5,000 ms on the 23rd block. This speedup was well fit by a power function—a result that is consistent with the CMPL model, according to which separate power functions govern speedup for each strategy. Further analysis showed stepwise speedup of about 200 ms for Algorithm Steps 2 and onward over the course of the first 13 training blocks, with no further speedup thereafter. In contrast, over the course of training there was several hundred milliseconds of slowing in latency to execute the algorithm first step. These results presage the results for shift items discussed below and are consistent with a positive latency choice competition between retrieval and the algorithm first step. For these no-shift items, it appears that the retrieval strategy did not become sufficiently competitive to win against the algorithm strategy before the end of training.

We also evaluated item-level RT plots, following the visual and statistical categorization scheme used by Rickard (2004). The results are shown in Table 1, along with results from the alphabet arithmetic task (Rickard, 2004) for comparison. About 6% of items exhibited no speedup, defined as a *p* value greater than .20 for the slope in a linear regression. Another 2.7% exhibited a step-function RT improvement (i.e., a visually prominent, abrupt, and sustained drop in RT) between the first and second block, indicating an immediate shift to retrieval (see 1st block items in Table 1), 41.9% exhibited step-function RT drops after Training Block 2 (Type 1 cluster items), 33.2% exhibited step-function RT drops after Block 2 with occasional slow outlier RTs late in training (Type 2 cluster items), and 16.2% exhibited smooth speedup (i.e., no pronounced step-function RT decrease).

Among the 22 items exhibiting no speedup, 15 never exhibited a shift to retrieval (i.e., for those items the algorithm was run to completion on every trial). The remaining 7 items showed various unusual patterns that masked speedup in the linear regression, most often a reversion back to use of the algorithm toward the end of training, yielding a U-shaped learning curve. Among the 60 items exhibiting smooth speedup, 40 never exhibited a shift to retrieval. The remaining 20 smooth speedup items (5.4% overall) are candidate cases of parallel strategy execution in which retrieval gradually becomes more competitive with the algorithm over multiple trials. It should be noted, however, following Rickard (2004), that obvious and usually dramatic deviations from smooth speedup were necessary before an item was classified as a Type 1 or Type 2 cluster item. For most smooth speedup items, there were hints of discontinuities similar to those for items classified as Type 1 or Type 2 cluster items.

Although the Type 1 and Type 2 cluster items rule out parallel strategy execution in which there is a gradual shift from algorithm to retrieval over trials for most items, they do not rule out the special case of parallel processing involving the abrupt increase in retrieval competitiveness that was described earlier. The finer grained analyses that are afforded by the current task design and that are discussed below provide a strong test of that version of the parallel model versus the choice model.

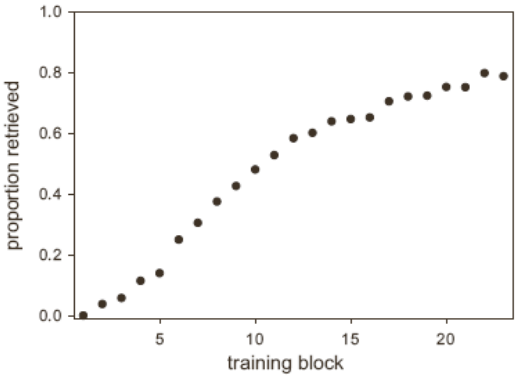

Figure 2. The proportion of trials on which the direct retrieval strategy was selected, as a function of training block.

Algorithm Step Latencies on Trials Preceding the First Correct Retrieval

Prior to conducting this analysis, we reset the training block variable for each item for each subject such that zero corresponded to the first correct retrieval block for that item, with blocks preceding the first correct retrieval taking negative values. For each subject, the mean latency (over items) for each algorithm step (correct trials only) was then computed for block values of −5 through −1 (i.e., for the last five algorithm blocks preceding each item's first correct retrieval block). These block means were then averaged over subjects and plotted in Figure 3A. Shown are results for the Algorithm Steps 1, 2, 3, and 4, along with the mean of Steps 5–9, among which there were no differences. Most items required more than nine algorithm steps, but data from those steps showed patterns like those for Steps 5–9 and so are not plotted.

The algorithm first step is substantially slower than subsequent steps, presumably reflecting subjects' need to orient to the presented stimulus and to initiate the counting algorithm. Also, for the algorithm first step, there was a pronounced (and not previously noted in the literature) 839 ms increase in latency from Block −5 through Block −1, confirmed by a within-subjects analysis of variance (ANOVA), $F(4, 130) = 10.51$, $p < .0001$. This algorithm first step slowing (which we term the *pause effect*) was not observed for Algorithm Steps 2–9, which are shown on a zoomed scale in Figure 3B. Instead, ANOVAs (identical to that described above) that were performed separately for each step indicated significant speedup over training blocks for Steps 3, 4, and the mean of Steps 5–9 ($p < .003$ in all cases). The U-shaped pattern for Step 2 did not reach significance ($p > .05$). In post hoc analyses, with the removal of the slower 2% of the data as outliers, the right section of the U-shaped curve for Step 2 was eliminated, yielding speedup analogous to that in Steps 3–9. The same outlier removal did not affect the shape of the functions in Figure 3 for any of the other algorithm steps. The speedup for Steps 2–9 appears to reflect algorithm learning over the course of the training blocks and is consistent with the algorithm speedup that was observed for the no-shift items. It is also possible that the speedup for Steps 2–9 on blocks approaching the first correct retrieval reflects attempts by subjects to make up for the time lost due to the Step 1 pause effect.

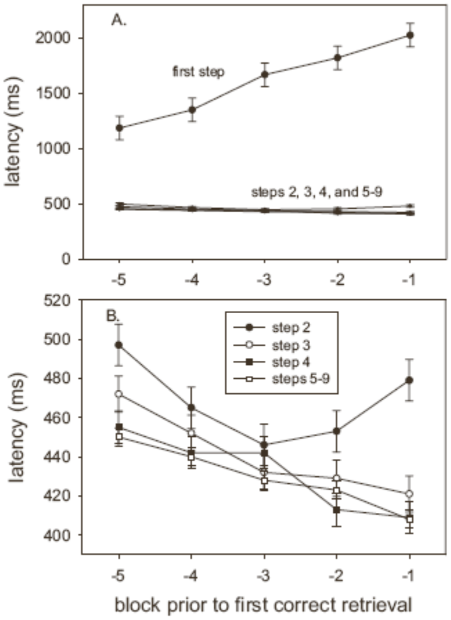

Figure 3. A: Mean algorithm step latencies on the five blocks prior to the first correct retrieval block for Algorithm Steps 1, 2, 3, and 4 and the mean of 5–9. B: Zoom of mean algorithm step latencies on the five blocks prior to the first correct retrieval block for Algorithm Steps 2, 3, and 4 and the mean of 5–9. Error bars represent standard errors of the mean.

Table 1
Visual and Statistical Categorization of the Item-Level Data

Learning curve category	Frequency	Percentage[a]	Percentage in Rickard (2004)	Items exhibiting shift to retrieval[b] *n*	%
No speedup	22	5.95	9	7	32
1st block	10	2.70	2.4	10	100
Type 1	155	41.89	48.3	154	99
Type 2	123	33.24	20.8	122	99
Smooth	60	16.22	29.1	20	33

[a] Out of all items. [b] Out of all items in each category.

To explore whether the pattern described above extended beyond 5 blocks, we performed a supplementary analysis in which we included the first 15 blocks preceding the first correct retrieval block. Because many subjects completed the shift to retrieval in fewer than 15 blocks for many or all items, the number of items qualifying for this analysis was reduced by 75% relative to the 5-block case. Results are shown in Figures 4A and 4B. The general pattern is similar to that in Figure 3, with the slowing again being significant for Step 1, $F(14, 207) = 6.31$, $p < .0001$; and the speedup being

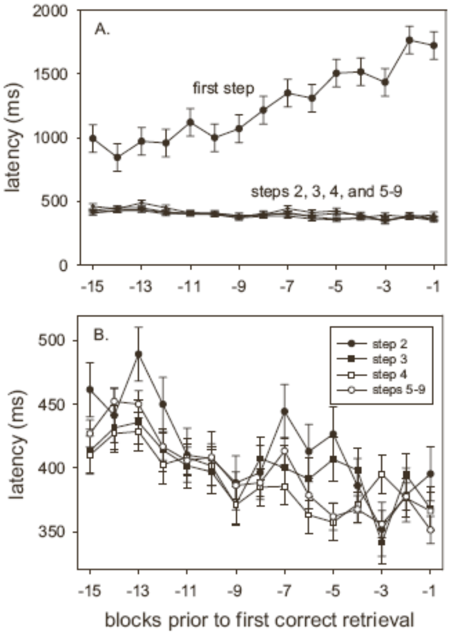

Figure 4. A: Mean algorithm step latencies on the 15 blocks prior to the first correct retrieval block for Algorithm Steps 1, 2, 3, and 4 and the mean of 5–9. B: Zoom of mean algorithm step latencies on the 15 blocks prior to the first correct retrieval block for Algorithm Steps 2, 3, and 4 and the mean of 5–9. Error bars represent standard errors of the mean.

significant for Steps 2, 4, and the mean of Steps 5–9 ($p < .05$ in all cases). In these averaged data, the algorithm first step slowing began about nine blocks prior to the first correct retrieval trial.

RTs (latency between stimulus onset and the vocal response) for items in the analyses described above (not shown in the figures) exhibited slowing from Block −5 to Block −1 that was analogous to, though of smaller magnitude than, that observed for Step 1 of the algorithm. That slowing was not statistically significant, however, because the Step 1 slowing was partially compensated for by the speedup that occurred over practice blocks for subsequent steps. Given this result, researchers should exercise caution when interpreting nonsignificant changes in overall algorithm RT over the course of training (e.g., Rickard, 2004). Note that algorithm first step slowing may not have been detected to date for other tasks because the first step latency has not been measured independently before, and because block numbers have not previously been synchronized to the first correct retrieval before averaging the data.

Accuracy on blocks approaching the first correct retrieval block is shown for the last 5 blocks in Figure 5A and for the last 15 blocks in Figure 5B. In both cases linear regression indicated significant negative slopes ($p < .05$ for the 5-block case and $p < .0001$ for the 15-block case). Further analyses showed that algorithm accuracy was constant in both cases at around .92. The increasing error rates were therefore entirely driven by increases in the rate of incorrect retrieval attempts.

It is worth noting that for 40% of the shift items, the shift occurred after the 13th training block, beyond which no algorithm speedup was observed for the no-shift items. Thus, we can reasonably infer that about 40% of the strategy shifts were occurring under conditions of constant, roughly asymptotic algorithm execution. Shift patterns for these 40% of items were qualitatively the same as for the other shift items. It appears, then, that the algorithm speedup with training for Steps 2 and onward was not a factor in the observed shift dynamics. Note also that none of these results depended on the number of algorithm steps, which varied from 9 to 15 over items.

Partial Algorithm Step Completion on Retrieval Trials

A bar graph of the frequency with which 0, 1, 2, 3, and so forth algorithm steps were completed during Blocks 0 to 4 (as defined in the preceding section, where zero corresponds to the first correct retrieval block for each item) is shown in Figure 6, along with the expected frequencies according to a race model. (Error trials and trials on which subjects reverted back to use of the complete algorithm were excluded.) These trials were used because they tended to have the slowest retrieval latencies, and hence they would be expected to exhibit the most algorithm step completion according to a parallel model. Results did not depend critically on this choice.

We derived the expected number of algorithm steps according to a race model in the following way. First, latencies for each algorithm step during the first block of the training phase were averaged over items for each subject (correct trials only). Prior anal-

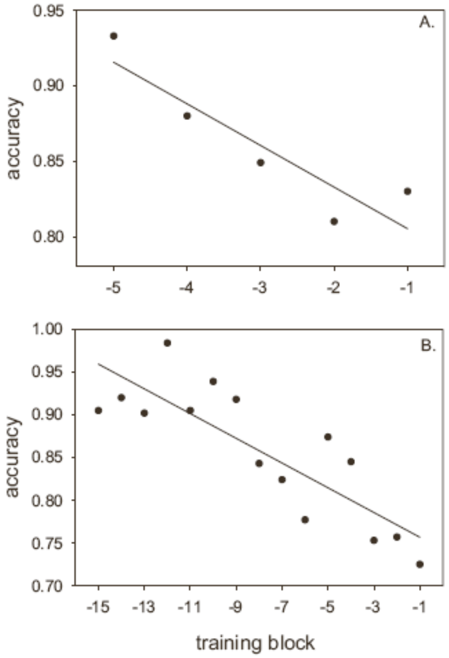

Figure 5. Accuracy on the last 5 (A) and last 15 (B) blocks prior to the first correct retrieval block.

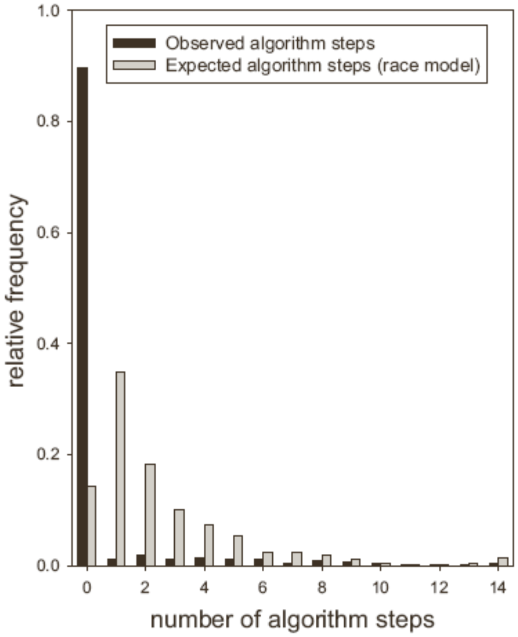

Figure 6. Relative frequency bar chart of expected (according to a race model) and observed completion frequencies for each algorithm step on the first five correct retrieval trials.

yses indicated no significant effect of items on these latencies, motivating the averaging. The first training block was used because no algorithm step slowing due to retrieval competition would be present. Next, for each retrieval trial under consideration, we estimated the expected number of completed algorithm steps under a race assumption by determining the number of first training block algorithm steps that the subjects would have been expected to complete on that trial—that is, by adding step latencies and recording the largest step number with a cumulative latency less than the retrieval latency for that trial. Note that because the algorithm exhibited speedup with practice, use of the latencies on the first training block in this calculation will tend to yield fewer predicted algorithm steps than would actually be expected by a race at the point of the strategy shift. Given the outcome described below, that bias does not complicate interpretation.

Figure 6 shows that partial algorithm step completion was far less frequent than expected by a race, $\chi^2(14) = 5{,}360$, $p < .0001$. Indeed, on 89.6% of those retrieval trials no algorithm steps at all were completed, a result that is in agreement with a strategy choice involving a competition between the retrieval strategy and the first step of the algorithm. For the remaining 121 trials, one or more algorithm steps were completed prior to answer retrieval (we refer to these as *partial algorithm* trials). These trials are candidates for parallel strategy execution.

A race model predicts that the speed of algorithm step execution on the partial algorithm trials will not be influenced by the race with retrieval. That prediction can be tested through analysis of algorithm step latencies on the partial algorithm trials shown in Figure 6 relative to the step latencies on the first training block, on which retrieval was not possible. For each subject, the means of the first, second, third, and fourth algorithm step latencies for the first training block were subtracted from the mean of these step latencies on partial algorithm trials. These difference scores were then subjected to matched t tests. (There were substantially fewer trials available for analysis of the later steps due to attrition. The trends, however, matched those of Steps 2–4.) Contrary to the race prediction, for Step 1 there was a highly significant 703.3 ms slowing on partial algorithm trials, $t(15) = 5.45$, $p < .0001$. For Steps 2–4, however, there were nonsignificantly faster completion times for the partial algorithm trials (difference scores of −93.8, −79.8, and −50.6, respectively; $p > .05$ in all cases).

Discussion

Implications for Skill Theories

None of the theories as developed to date can fully account for performance on the current task. The race theories cannot account for the results, even for the relatively infrequent partial algorithm trials, because of the substantial algorithm first step slowing that occurred even on those trials. Parallel models as a more general class, including limited capacity models, fare little better. Although a limited capacity model can explain the algorithm first step slowing on trials approaching the first retrieval trial (and on partial algorithm trials) when that result is considered in isolation, it cannot in any straightforward way explain why that slowing was not also observed on subsequent algorithm steps for those trials.

The results instead indicate a strategy choice process that involves a competition between the algorithm first step and the memory retrieval strategy. At that level of analysis the CMPL model fares well. However, the CMPL simulation model as developed to date (Rickard, 1997) assumes a zero-latency choice process, an assumption that is not, on its face at least, consistent with the data. Given the magnitude of the observed slowing (839 ms for Block −1 relative to Block −5 in Figure 3A), the simplest way to modify the CMPL model to account for the results would be to assume that the direct retrieval strategy wins the initial competition on some of those trials, and is executed, but that subjects are in some cases not sufficiently confident in the retrieved answer. They may then hold the retrieved answer in working memory and run the algorithm as a check. If the answers generated by the two strategies match, they may then tag that item as supporting correct retrieval and then rely solely on retrieval on subsequent trials if they recall the tag. This hypothesis may also explain the algorithm first step slowing that was observed on the partial algorithm trials. During algorithm execution on those trials, subjects may have decided to speak the previously retrieved answer prior to finishing the algorithm. This might occur, for example, because the act of counting narrows the range of candidate answers, potentially leading subjects to have more confidence in their initially retrieved answer. For both of these types of trials, this account characterizes the algorithm first step slowing as being the result of a *postchoice* strategy. The zero-latency choice process is thus, in principle, consistent with this account (i.e., the initial choice to retrieve might not be slowed by the competition with the algorithm). It should also be noted that the initial attempt at retrieval might simply fail to yield an answer. Within a framework such as the CMPL model, the retrieval strategy might win the competition, but the association to the answer may not be strong enough to bring answer activation above a response threshold. In this case, subjects would shift to the algorithm as a back-up strategy, in a manner analogous to that hypothesized in Siegler's (1988) distribution of associations model.

Alternatively, or in addition to the postchoice, sequential strategy execution hypothesized above, it is possible that the first step slowing reflects a *prechoice* (and presumably preconscious) competition that increases the time for the algorithm first step to be selected. This possibility corresponds, by our earlier definition, to a positive latency choice process, and it does not currently have an implementation in the CMPL model. Given the observed first step slowing of more than 800 ms (see Figure 3A), we speculate that the postchoice, sequential strategy account is most likely correct, at least as the major component of the slowing.

The Schunn et al. (1997) choice model has a number of features in common with CMPL. Both models assume a strategy choice that occurs prior to initiating either retrieval or the algorithm. Reder and colleagues (e.g., Schunn et al., 1997) have focused on feeling of knowing, which is modeled by the activation of stimulus representations within a semantic network, as a mechanism of strategy choice. The CMPL model embodies a similar network implementation of choice while also staking claims about the specific nature of the bottleneck that requires a strategy choice and about the manner in which the algorithm competes with retrieval. The two models appear to be compatible, and indeed they may offer prospects for synthesis. In the Rickard (1997) simulation model, activation of the problem-level node that corresponds to the retrieval strategy could serve as the basis for subjective feeling of knowing. The SAC model incorporates a mechanism for interference among items with overlapping operands that could be integrated with CMPL.

Generalization to Other Tasks

Clearly, strategy execution is not parallel in the current task, but to what range of tasks does that conclusion extend? Although more research is needed to address this question, a reasonably strong prediction can be made by considering the simplicity and low subjective cognitive load of our counting–tapping algorithm. We submit that similar results would be obtained for any of the broad class of algorithms that require a series of long-term memory retrieval steps. For example, given the current results it seems unlikely that a repeated addition algorithm for single-digit multiplication—which is much more subjectively taxing for children than is simple counting for adults—would run in parallel with retrieval.

Our task is atypical among those explored in the literature to date in that it required a simple keypress response in coordination with each algorithm step. It is not unique, however, in its requirement that a motor event take place in coordination with each algorithm step. The dot-counting task (e.g., Palmeri, 1997), which requires an orienting eye movement with each count, shares that property. For a number of reasons, it is unlikely in our view that simple algorithm-related motor events are the primary reason why retrieval and algorithm strategies were not executed in parallel in the current task. First, the counting component of the algorithm is a form of serial memory retrieval with (presumably) subvocal manifestation of each count, and it is on its face more likely to interfere with verbally based direct retrieval of the answer than is simple repetitive keypressing. Buttressing this claim is evidence that simple tapping alone has a negligible effect on other cognitive processing (for discussion see Pashler, 1994). Second, as noted above, the overall attentional demands of the algorithm used here are small compared to most other algorithms that have been explored in the literature to date or that occur in natural settings (e.g., arithmetic algorithms). Third, the item-level learning curve categorization for the current experiment is highly similar to that observed by Rickard (2004), who used an algorithm that involved no motor component (see Table 1).

We cannot strictly rule out the possibility that subjects used desynchronized algorithm tapping and counting; for example, tapping between counts. However, because desynchronizing is likely more time consuming than synchronizing, and because there was no task performance benefit to desynchronizing, it is reasonable to assume that subjects synchronized. If subjects did desynchronize, with counting preceding tapping, then it is possible that they executed retrieval in parallel through completion of the first count but aborted retrieval on the first tap. This possibility implies, however, that the tapping is the primary factor preventing parallel strategy execution, a possibility that appears unlikely in light of prior data on finger tapping that we noted above. Note also that neither the race models of skill nor any other current model of attention and performance that we know of would predict that simple, repetitive keypresses themselves would be sufficient to preclude parallel direct retrieval from long-term memory. The race models of skill as developed to date treat all algorithms homogeneously, and they assume that retrieval can take place in parallel throughout the execution of any algorithm.

It is an open question whether the same results will be obtained for the class of algorithms that do not require a sequence of long-term memory retrieval steps. Examples include practice on visual search and mental rotation with repeated presentation of the same items. More generally, any algorithm that involves only the execution of rules held in working memory is a member of this class. Additional work to explore strategy execution dynamics in such task domains is needed.

References

Byrne, M. D., & Anderson, J. R. (2001). Serial modules in parallel: The psychological refractory period and perfect time-sharing. *Psychological Review, 108*(4), 847–869.

Delaney, P. F., Reder, L. M., Staszewski, J. J., & Ritter, F. E. (1998). The strategy-specific nature of improvement: The power law applies by strategy within task. *Psychological Science, 9,* 1–7.

Haider, H., & Frensch, P. A. (2002). Why aggregated learning follows the power law of practice when individual learning does not: Comments on Rickard (1997, 1999), Delaney et al. (1998), and Palmeri (1999). *Journal of Experimental Psychology: Learning, Memory, and Cognition, 28,* 392–406.

Heathcote, A., Brown, S., & Mewhort, D. J. K. (2000). The power law repealed: The case for an exponential law of practice. *Psychonomic Bulletin and Review, 7,* 185–207.

Jenkins, L., & Hoyer, W. J. (2000). Instance-based automaticity and aging: Acquisition, reacquisition, and long-term retention. *Psychology and Aging, 15,* 551–565.

Kail, R. (1986). The impact of extended practice on rate of mental rotation. *Journal of Experimental Child Psychology, 42*(3), 378–391.

Kole, J. A., & Healy, A. F. (2007, November). *Examining the retrieval process in mediated learning using priming effects.* Poster presented at the 48th annual meeting of the Psychonomic Society, Long Beach, CA.

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

Logan, G. D. (1992). Shapes of reaction-time distributions and shapes of learning curves: A test of the instance theory of automaticity. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 18*(5), 883–914.

Navon, D., & Miller, J. (2002). Queuing or sharing? A critical evaluation of the single-bottleneck notion. *Cognitive Psychology, 44,* 193–251.

Nino, R. S., & Rickard, T. C. (2003). Practice effects on two memory retrievals from a single cue. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 29*(3), 373–388.

Nosofsky, R. M., & Palmeri, T. J. (1997). An exemplar-based random walk model of speeded classification. *Psychological Review, 104,* 266–300.

Onyper, S. V., Hoyer, W. J., & Cerella, J. (2006). Determinants of retrieval solutions during cognitive skill training: Source confusions. *Memory & Cognition, 34,* 538–549.

Palmeri, T. J. (1997). Exemplar similarity and the development of automaticity. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 23*(2), 324–354.

Palmeri, T. J. (1999). Theories of automaticity and the power law of practice. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 25*(2), 543–551.

Pashler, H. (1994). Dual-task interference in simple tasks: Data and theory. *Psychological Bulletin, 116*(2), 220–244.

Rawson, K. A. (2004). Exploring automaticity in text processing: Syntactic ambiguity as a test case. *Cognitive Psychology, 49,* 333–369.

Reder, L. M. (1987). Strategy selection in question answering. *Cognitive Psychology, 19,* 90–138.

Reder, L. M. (1988). Strategic control of retrieval strategies. In G. H. Bower (Ed.), *The psychology of learning and motivation* (Vol. 22, pp. 227–259). San Diego, CA: Academic Press.

Reder, L. M., & Ritter, F. E. (1992). What determines initial feeling of knowing? Familiarity with question terms, not with the answer. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 18*(3), 435–451.

Rickard, T. C. (1997). Bending the power law: A CMPL theory of strategy shifts and the automatization of cognitive skills. *Journal of Experimental Psychology: General, 126,* 288–311.

Rickard, T. C. (1999). A CMPL alternative account of practice effects in numerosity judgment tasks. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 25*(2), 532–542.

Rickard, T. C. (2004). Strategy execution in cognitive skill learning: An item-level test of candidate models. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 30*(1), 65–82.

Rickard, T. C., & Bajic, D. (2003). Automatic mediation or absence of mediation? Commentary on Crutcher and Ericsson (2000). *Journal of Experimental Psychology: Learning, Memory, and Cognition, 29*(6), 1381–1386.

Rickard, T. C., & Bajic, D. (2006). Cued recall from image and sentence memory: A shift from episodic to identical elements representation. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 32*(4), 734–748.

Rogers, W. A., Hertzog, C., & Fisk, A. D. (2000). An individual differences analysis of ability and strategy influences: Age-related differences in associative learning. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 26*(2), 359–394.

Schunn, C. D., Reder, L. M., Nhouyvanisvong, A., Richards, D. R., & Stroffolino, P. J. (1997). To calculate or not to calculate: A source activation confusion model of problem familiarity's role in strategy selection. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 23*(1), 3–29.

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

Tao, L., & Healy, A. F. (2002). The unitization effect in reading Chinese and English text. *Scientific Studies of Reading, 6,* 167–197.

Tombu, M., & Jolicoeur, P. (2003). A central capacity sharing model of dual-task performance. *Journal of Experimental Psychology: Human Perception and Performance, 29,* 3–18.

Touron, D. R., Hoyer, W. J., & Cerella, J. (2001). Cognitive skill acquisition and transfer in younger and older adults. *Psychology & Aging, 16,* 555–563.

Touron, D. R., Hoyer, W. J., & Cerella, J. (2004). Cognitive skill learning: Age-related differences in strategy shifts and speed of component operations. *Psychology & Aging, 19,* 565–580.

Wagenmakers, E., & Brown, S. (2007). On the linear relation between the mean and the standard deviation of a response time distribution. *Psychological Review, 114,* 830–841.

Appendix

Phase 2 Stimulus–Response Pairings

Stimuli	Responses
20	34
21	36
22	32
23	35
24	33
25	37
26	41
27	40
28	39
29	38

Received March 27, 2008
Revision received July 24, 2008
Accepted July 25, 2008 ■

Chapter 2, in full, is a reprint of the material as it appears in Bajic, D., & Rickard, T. C. (2009). The temporal dynamics of strategy execution in cognitive skill learning. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 35*(1), 113-121. Copyright 2009 by the American Psychological Association. Reproduced with permission. The official citation that should be used in referencing this material is listed above. The use of APA information does not imply endorsement by APA. The dissertation author was the primary investigator and author of this paper.

Chapter 3

The Case of Simple Perceptual-Motor Algorithms

Abstract

Prior work has shown that algorithmic and direct memory retrieval strategies are not executed in parallel in cognitive skill learning tasks in which the algorithm involves a series of long-term memory retrievals (as is the case, for example, in arithmetic). This effect has been hypothesized to reflect a bottleneck in memory retrieval processes that forces a strategy choice during an early stage of processing. Here we investigate simple perceptual-motor algorithms that involve no memory retrieval steps. The data indicate a surprising amount of interference between algorithmic and memory retrieval strategies even in this case, eliminating simple versions of parallel strategy execution models. We infer that an early-stage strategy execution bottleneck precludes parallel strategy initiation even in the case of perceptual-motor algorithms. We advance a theoretical framework that can accommodate results both for perceptual-motor algorithms and for algorithms that involve memory retrieval steps.

A common phenomenon in cognitive skill learning is a shift with practice from reliance on use of a multistep algorithm to direct memory retrieval. The classic example is arithmetic learning. In doing single-digit multiplication, children may initially perform a repeated addition algorithm, but with sufficient practice will transition to direct retrieval (Siegler, 1988). Multiple laboratory studies have confirmed the occurrence of this shift over a variety of arithmetic and non-arithmetic tasks (Delaney, Reder, Staszewski, & Ritter, 1998; Hertzog, Touron, & Hines, 2007; Jenkins & Hoyer, 2000; Logan, 1988, 1992; Onyper, Hoyer, & Cerella, 2006; Palmeri, 1997; Rawson, 2004; Reder & Ritter, 1992; Rickard, 1997, 1999, 2004; Rickard & Bajic, 2003, 2006; Rogers, Hertzog, & Fisk, 2000; Schunn, Reder, Nhouyvanisvong, Richards, & Stroffolino, 1997; Touron, Hoyer, & Cerella, 2001).

There has been debate about the temporal dynamics of strategy execution on each trial, with one group of models assuming parallel (simultaneous) strategy execution (Logan, 1988; Palmeri, 1997), and a second class of models assuming a strategy choice process such that only one strategy is executed at a time (Rickard, 1997, 2004; Schunn et al., 1997; Siegler, 1988). The most recent evidence (Bajic & Rickard, 2009) clearly favors the strategy choice account, at least for the case of algorithms that involve memory retrieval steps. Bajic and Rickard used a task design that allowed indexing of not only the response time (RT) for each trial (defined as the latency between stimulus onset and response execution) but also the latency for completion of each step of the algorithm. On each trial of the experiment, subjects saw a two-digit number and were instructed to count forward from that number, pressing the space bar in synchrony with each count, until the computer informed them to stop and to speak the number to which they had counted. For each stimulus number (e.g., 21) the same number of counts was always required (say, 11), and the same response was always to be spoken (e.g., 32). Each stimulus was presented multiple times over training blocks. If subjects remembered the answer at any point during a trial, they could end the trial prior to completing the algorithm by speaking that answer (these trials will be referred to as *retrieval trials* even if some algorithm steps were completed). Each key press recorded the approximate latency of each counting step, and a microphone voice-key recorded the RT.

This task design allowed Bajic and Rickard (2009) to explore two previously unaddressed questions about the temporal dynamics of strategy execution: (1) on the last few algorithm trials preceding the first correct retrieval trial for an item, is there evidence of progressively slower execution times for one or more algorithm steps, as might be expected if the retrieval strategy becomes more competitive over trials and if there is a latency-consuming strategy competition? And (2) on retrieval trials, is there evidence that some fraction of the algorithm steps are completed, as would be expected if the two strategies can be executed in parallel?

With respect to both questions above, the data supported a strategy choice process in which there is a competition between the first step of the algorithm (i.e., the first tap-count event) and the retrieval strategy. On the last few algorithm trials preceding the first correct retrieval trial, there was a marked slowing of execution of the first algorithm step (but not for subsequent steps), reaching a peak value of over

800 ms on the trial immediately preceding the first correct retrieval trial. Bajic and Rickard (2009) termed this selective first step slowing the *pause effect* (because on observation it often produced a palpable hesitation at the beginning of the trial) and we will refer to the increasing magnitude of this effect approaching the first correct retrieval as the *pause effect slope*. The algorithm was apparently blocked temporarily on some of those trials while subjects attempted to retrieve the answer. Answer retrieval either failed, or subjects were insufficiently confident to speak the answer based on retrieval, and thus proceeded to execute the algorithm as a backup strategy. The positive slope of the pause effect suggests that the proportion of trials on which retrieval was initially attempted increased steadily on trials approaching the first correct retrieval.

On the first few correct retrieval trials for each item (when retrieval was slowest), one or more algorithm steps were completed on only about 10% of trials (we will refer to these as *partial-algorithm* trials), whereas a parallel race account (i.e., independent, capacity unconstrained parallel strategy execution; Logan, 1988; Palmeri, 1997) predicted that one or more algorithm steps should have been observed on about 86% of trials in that portion of the study. For the few partial-algorithm trials that were observed, there was a roughly 700 ms *partial-algorithm pause effect* (i.e., a 700 ms delay in execution of the algorithm first step only) just as there was for algorithm trials approaching the first correct retrieval, ruling out simple race or limited capacity parallel retrieval models even for those trials.

Bajic and Rickard (2009) interpreted their results in the context of Rickard's (1997) component power laws (CMPL) model. That model assumes a bottleneck in cued-recall from long-term memory, such that only one response can be retrieved at a time (see also Nino & Rickard, 2003). The entire retrieval, up to the point at which the answer becomes available for responding, must be completed before another retrieval can be initiated. Thus, for any algorithm that involves a series of memory recall events (even a series of retrievals as simple as counting), the algorithm and retrieval strategies can not be executed in parallel (for more detail on the CMPL model, see Rickard, 1997, 1999, 2004; Rickard & Bajic, 2004; Bajic & Rickard, 2009).

The CMPL model leaves open the possibility that algorithm and retrieval strategies can be executed in parallel for algorithms that do not involve a series of long-term memory retrievals. An example laboratory task is noun-pair look-up (Hertzog et al., 2007; Touron et al., 2001). In a typical version of that task, a set of noun pairs is presented at the top of the screen. On each trial, a noun pair cue is presented in the center of the screen, and the subject must indicate, by pressing one of two keys, whether the cue pair corresponds to one of the pairs on the top of the screen. Subjects must either remember the paired nouns (direct retrieval) or visually search the noun-pair table (the algorithm) to find it. In this task, execution of the search algorithm does not require retrieval from long-term memory. There are also numerous cases in everyday cognition in which people can either retrieve an answer from memory or execute a visual or visuo-motor search algorithm. Consider, as examples, remembering a keyboard shortcut to perform a word processing operation versus searching a dropdown menu, or remembering a person's name or phone number versus looking it up on a contact list.

The goal of the current experiments was to explore the temporal dynamics of strategy execution when the algorithm does not involve memory retrievals. We compared simple key-tapping and mouse-clicking algorithms (that involved no memory retrieval processes) to matched versions of those algorithms that did involve retrieval (i.e., counting) using the basic design and methods employed by Bajic and Rickard (2009).

Experiment 1

This experiment is nearly identical to the experiment described in Bajic and Rickard (2009), the major exception being that subjects were not required to count as they executed the key tapping algorithm. Subjects simply began tapping when the stimulus was presented and stopped tapping either when the response was provided by the computer or when they remembered the response.

Method

Subjects

31 University of California at San Diego undergraduate students participated for course credit.

Materials, Design, and Procedures

Subjects were tested individually on IBM-compatible personal computers, with each subject seated approximately 50 cm from the computer screen, and approximately 3 cm from a microphone. The computer keyboard was positioned directly behind the microphone, such that the subject could comfortably place one hand over the spacebar key. The experimenter was seated to the right of the subject, with access to the keyboard's number pad. The experiment was created with E-Prime software (Psychology Software Tools, Pittsburgh PA) and the accompanying voice key apparatus (model 200A).

The experiment was designed to be as similar as possible to the experiment conducted by Bajic & Rickard (2009), while using an algorithm that did not involve any memory retrieval steps. To achieve this, the count-tap algorithm from that study was replaced by a simple repeated tapping algorithm. The experiment consisted of a warm-up phase and a training phase. Prior to each phase, instructions were presented on the screen, and were also read aloud by the experimenter. Within each trial of each phase, a two-digit number stimulus was presented visually and the subject had to speak the answer (another two-digit number) either by use of the tapping algorithm or through memory retrieval. The Appendix lists all visual stimulus and vocal response items used in the training phase. The warm-up phase utilized the same values, each raised by 10 (e.g., "30" and "44" in the warm-up phase versus "20" and "34" in the training phase, etc.). The Appendix also lists the Algorithm Steps, which were the number of key taps required to complete the algorithm for each item. Although the visual stimulus and vocal response pairings were the same across all subjects, the Algorithm Steps were randomly assigned to items for each subject, with the only requirement being that the number of algorithm steps for a particular item could not be equal to the value of the correct response minus the numerical stimulus (that is, the correct answer could not be found by counting taps). In the description below, a *block* is defined as one randomly ordered presentation for each of the 10 possible stimulus-response items, each item therefore having a mean repetition lag of 10 trials across blocks.

The warm-up phase consisted of a single block. At the start of each trial, the screen went blank for 500 ms, a fixation field (consisting of three plusses) was presented at the center of the screen for 500 ms, the screen again went blank for 500 ms, and then a two-digit number—the trial stimulus—was presented at the center of the screen. Subjects were instructed to begin rapidly tapping the spacebar key when the number appeared. After the number of taps equaled the number of algorithm steps for that item, the stimulus disappeared, and the word "STOP" was presented on the screen, along with the answer for that trial presented just below. The subject would then speak the answer into the microphone.

After the subject provided a vocal response, the experimenter entered the subject's response, and recorded whether the voice key tripped properly. Immediately after each incorrect trial, the correct response was presented for 5 s. Otherwise, the word "Correct!" was presented for 800 ms. Immediately following feedback, the next trial began.

The training phase of the study was identical to the warm-up phase, with the following exceptions. Multiple blocks were presented, and subjects were informed that the same set of starting numbers (stimuli) would be presented repeatedly throughout the phase, with each starting number always having the same final number. Subjects were informed that they therefore had two methods that could be used to find and speak the correct answer for each trial: (1) tapping the spacebar until the word "STOP" and the answer for that trial appeared on the screen (the algorithm), and (2) remembering the answer associated with the starting number for that trial and speaking the answer into the microphone without doing all of the key presses. To promote parallel strategy execution, if such parallelism is possible, the instructions stated (falsely) that, "Many subjects report good results when they attempt to use both strategies at the same time." Subjects were told that they could speak the answer into the microphone at any time during each trial. They were instructed that they should try to finish this part of the experiment as quickly as possible, while still being accurate.

Each trial stimulus was removed from the screen either when the subject spoke an answer, or when the subject had entered a sufficient number of key-presses to bring the word "STOP" onto the screen—whichever came first. Subjects were permitted a brief pause between each block, and continued to receive new blocks until 45 minutes from the start of this phase, after which the experiment concluded and the subject was debriefed.

Results

For all experiments, only the training phase data were analyzed and subjects were excluded from analysis if they gave inaccurate responses on more than 20% of trials, or if they made voice key errors on more than 10% of trials.

For Experiment 1, all subjects had sufficiently high accuracy for inclusion, and four subjects were rejected due to frequent voice key errors. Individual trials were excluded from analysis in all experiments if voice key errors occurred (approximately 3% of trials in Experiment 1), or if there was a latency of less than 180 ms from stimulus onset to the first algorithm step, or less than 300 ms from stimulus onset to the vocal response (less than 1% of trials across all experiments).[6] (Note that in supplementary analyses with excluded data reintroduced, it was found that the above exclusion rules did not alter the central results or any conclusions discussed below.)

Mean accuracy was initially near-perfect (over 99% on the first training block), fell to its lowest value (approximately 91%) on Block 10, then rose again to approximately 98% by the 27th block, the furthest block that all 27 subjects completed.

The mean of the subject-level mean correct RTs (latency from stimulus presentation to vocal response) is plotted as a function of training block in Figure 3.1A, and the mean of the subject-level SDs is plotted in Figure 3.1C. For reference, the corresponding data from the matched version of this experiment that involves counting and tapping (Bajic & Rickard, 2009) are shown in Figures 3.1B and 3.1D. Best-fitting three-parameter power functions for the RT data are also included for reference. Both the deviation from power function decreases in RT and the inverted U-shaped SD values are consistent with results of prior studies. These

[6] See Luce (1991) and Rickard (2004).

effects have to date only been observed for tasks known to exhibit a shift to retrieval, and reflect a strategy mixture over trials for each subject during the strategy shift portion of training (Rickard, 1999). Note the substantially faster latencies of the tap-only task in Figure 3.1A versus the tap-count task in Figure 3.1B. This latency difference is expected because the relatively time-consuming counting operation is absent for the tap-only group.

The proportion of correctly answered trials in which subjects retrieved the answer (defined as those trials in which the subject spoke the answer before completing all algorithm steps) is shown as a function of training block in Figure 3.1E. The strategy shift was nearly 100% complete by the 27th block. Both Figure 3.1E and the peak of the SD curve in Figure 3.1C indicate that the shift to retrieval had occurred for approximately 50% of the items by about Block 10 (see Bajic and Rickard, 2009, and Rickard, 1999, for discussion of the relationship between peak SD and proportion of trials on which retrieval occurs). Similar results were observed by Bajic and Rickard (2009; see Figure 3.1F), although there appears to have been a slightly slower rate of shift to retrieval in that experiment.

Algorithm step latencies on trials preceding the first correct retrieval

Here and in all subsequent analyses of algorithm step latency effects on trials preceding the first correct retrieval, the following procedure was used. First, prior to conducting this analysis, the training block variable for each item for each subject was reset, such that zero corresponded to the first correct retrieval block for that item, with blocks preceding the first correct retrieval taking negative values. Below, when referring to block numbers synchronized in this manner, we use the abbreviated term *sync-blocks*. For each subject, the mean latency (over items) for each algorithm step was then computed for sync-block values of -5 through -1 (i.e., for the last five algorithm blocks preceding each item's first correct retrieval block). Items with 5 or fewer blocks prior to the first correct retrieval were excluded from this analysis. These sync-block means were then averaged over subjects and plotted in Figure 3.2A. Shown are results for Algorithm Steps 1, 2, 3, and 4, along with the mean of Steps 5-9. Most items required more than nine algorithm steps, but data from those steps showed patterns like those for Steps 2-9 and so are not plotted.

The algorithm first step is substantially slower than subsequent steps, presumably reflecting the need to orient to the presented stimulus and to initiate the tapping algorithm. Also for the algorithm first step, there was a pronounced 340 ms increase in latency from Sync-block -5 through Sync-block -1 (the pause effect slope), confirmed by a within-subjects analysis of variance (ANOVA), F(4, 104) = 9.63, $p < .0001$. Note, however, that this effect is less than half the size of the pause effect slope for the tap-count algorithm in Bajic and Rickard (2009; shown in Figure 3.2B for reference). There was no significant effect of sync-block for Steps 2, 3, 4, or 5-9 ($p > .05$ in all cases).

To test for group differences in the pause effect in this experiment versus those observed by Bajic and Rickard (2009), a mixed factors analysis of variance (ANOVA) with factors of experiment and sync-block (-5 through -1) was conducted on the subject-level mean latencies for Step 1. There were significant effects of sync-block, F(4, 232) = 17.25, p < .0001, and group, F(1, 58) = 17.84, p < .0001. There was also a significant interaction, F(4, 232) = 3.18, p < .02, as reflected in the different latency slopes in Figures 3.2A and 3.2B. The same analysis, when performed on Algorithm Steps 2 through 9, also yielded significant effects of group and the interaction in all cases. As expected, latencies for individual algorithm steps were faster overall for the tap-only algorithm. In addition, practice apparently somewhat improved the speed of algorithm step execution (for Steps 2 and onward) for the tap-count algorithm but not for the tap-only algorithm.

Partial-algorithm step completion on retrieval trials

A bar graph of the frequency with which 0, 1, 2, 3, or more algorithm steps were completed during Sync-blocks 0 to 4 (as defined in the preceding section, where zero corresponds to the first correct retrieval block for each item) is shown in Figure 3.2C, along with the expected frequencies according to a race model. These trials were used because they tended to have the slowest retrieval latencies, and hence would be expected to exhibit the most algorithm step completion according to a parallel model. (Results did not depend critically on this choice.)

Here and in all subsequent analyses, the expected number of algorithm steps according to a race model was derived as in Rickard and Bajic (2009). First, latencies for each algorithm step during the first block of the training phase were averaged over items for each subject. Prior analyses indicated no significant effect of items on these latencies, motivating the averaging. The first training block was used because no algorithm step slowing due to retrieval competition would be present. Next, for each retrieval trial under consideration, the expected number of completed algorithm steps under a race assumption was estimated by determining the number of first training block algorithm steps that the subjects would have been expected to complete on that trial. (That is, if a particular subject had a Block 1 mean latency of 1200 ms for the completion of four algorithm steps, and 1400 ms for the completion of five steps, then a race model would predict that this subject would complete four algorithm steps on a trial with a vocal RT of 1300 ms.)

As shown in Figure 3.2C, algorithm steps were completed less frequently than predicted by the race model. Statistical significance of this effect was tested by first computing the difference between the number of algorithm steps expected by the race model and the number of observed steps on each trial, taking the mean of these difference scores over trials for each subject, and then conducting a Wilcoxon Signed Rank Test on those mean difference scores, $T^+ = 378$, $p < .0001$. The failure of the race model to fit the data is driven largely by (1) a much larger than predicted percentage of trials with zero completed algorithm steps, and (2) very slow retrievals on about 21% of trials for which the race model predicted that all algorithm steps should have been completed (represented by the ">" column of Figures 3.2C and 3.2D).

Despite the strong statistical rejection of the race model, there were far more trials on which one or more steps were completed than were observed by Bajic and Rickard (2009) for the tap-count algorithm (shown in Figure 3.2D). Statistical significance of this difference in distribution shapes was confirmed using the χ^2 test of independence, $\chi^2(12) = 353.1$, $p < .0001$.[7] Note also that, with respect to the tap-only group, there was a tendency for the observed distribution of completed algorithm steps (beyond 0 steps) to mimic the distribution predicted by the race model (albeit with a left-shifted mode, as would be expected given the pause effect on partial-algorithm trials that is discussed below).

The race model predicts that the speed of algorithm step execution on the partial-algorithm trials will not be influenced by the race with retrieval. One approach to evaluating these race model latency predictions is through analysis of algorithm step latencies on the partial-algorithm trials shown in Figure 3.2C relative to the step latencies on the first training block (where retrieval was not possible). For each subject, the means of the latencies for Algorithm Steps 1 to 4 for the first training block were subtracted from the mean of these step latencies on partial-algorithm trials (due to data attrition, partial-algorithm latencies on subsequent steps were not analyzed). These difference scores (depicted in Figure 3.2E, with positive scores indicating slowing relative to Block 1, and negative scores indicating speed-up) were then subjected to matched t-tests. Contrary to the race prediction, for Step 1 there was a significant 745.59 ms pause effect, $t(24) = 3.64$, $p < 0.002$. This result roughly matches the more than 700 ms partial algorithm pause effect in Bajic and Rickard (2009; see Figure 3.2F).

These difference scores for Algorithm Steps 2 to 4 were: 67.88, 54.53, and 65.53, respectively. Despite being more than one order of magnitude smaller than

[7] Here and elsewhere, bins were combined as needed to meet χ^2 assumptions of expected frequency of no fewer than 5.

the slowing observed for Step 1, these results were still significant ($p < .05$ in all cases). This result is in contrast to Bajic and Rickard (2009), who observed nonsignificant speed-up for Steps 2-4 on partial-algorithm trials in the same type of analysis.

One potential problem with analyzing partial-algorithm latencies by way of comparison to step latencies on the first practice block is that the results may be biased if there is algorithm step speed-up or slowing over the first few practice blocks, which could occur independently of any competition effects with the retrieval strategy. An analysis of algorithm step latencies on the first five practice blocks confirmed that there was modest speedup in Steps 2-4 (about 30 to 60 ms from the first to the fifth block) for the Bajic and Rickard (2009) experiments, as well as for both groups in Experiment 2 of this paper. It appears that, in these experiments, subjects gained algorithm step execution skill over the first few blocks of practice. In Experiment 1 of this paper, in contrast, there was about 20 ms of slowing of Algorithm Steps 2-4 over the first five practice blocks. We suspect that subjects initiated the very simple and highly automated repeated tapping at a very fast rate that could not be improved upon with practice. Instead, there appears to have been a slight fatigue effect for that algorithm, or an adjustment to a slightly more comfortable tapping pace. Similar fatigue or adjustment effects might have also occurred for the algorithm steps in the other experiments but may have been more than compensated for by the larger magnitude learning effects.

These modest sources of bias in the tests of the race model for partial-algorithm step latencies are too small to materially influence the inferences that were drawn from the distribution of completed algorithm steps (Figure 3.2C) and from the partial-algorithm pause effect of more than 700 ms. However, these 20-60 ms biases are similar in magnitude to the partial-algorithm latency discrepancies (relative to the race prediction) for Steps 2-4 that were discussed above. As a supplemental approach that is less influenced by changes in algorithm step latency over the first few practice blocks, we analyzed latencies for Steps 2-4 for Sync-block -1 in comparison to Sync-block 0 (the first correct retrieval block), limited (for both sync-block values) to items that exhibited partial-algorithm step completion on Sync-block 0. As can be seen in Figures 3.2A and 3.2B, latencies for Algorithm Steps 2 and upward were relatively stable throughout the range from Sync-block -5 to -1. Also, the increasingly competitive retrieval strategy was not interfering with latencies during those sync-blocks, as evidenced by the lack of any significant slowing for Steps 2 and upward. Thus, the latencies on Sync-block -1 for Steps 2-4 provide a relatively bias-free estimate of what the expected latency should be for those steps on Sync-block 0 according to a race model, or in the case in which there is no ongoing retrieval attempt during partial-algorithm execution. Results of these analyses were similar for Steps 2-4, so data were averaged over these steps. For each subject who completed some algorithm steps at Sync-block 0, the mean latency of Steps 2-4 for all partial-algorithm items at Sync-block 0 were compared against the mean latencies of those steps for the same items at Sync-block -1. Matched t-tests revealed nonsignificant (p > .05) differences of less than 20 ms both for Experiment 1 and for the data from Bajic and Rickard (2009). It appears, then, that there was no (or at most minimal) algorithm step slowing for Steps 2-4 in these experiments, not only on algorithm trials approaching the first correct retrieval trial, but also for the partial-algorithm steps on the first correct retrieval trial. The significant effects reported earlier for Steps 2-4, using comparison to the latencies on the first practice block, are most likely due to algorithm fatigue effects in the present experiment, while the nonsignificant speed-up found in the same comparison for the Bajic and Rickard (2009) data likely represents modest algorithm learning over the course of the study.

Discussion

There was a striking degree of strategy interference even for this simple tapping task that was predicted by neither the CMPL model nor any other model in the literature, including models of the shift to retrieval in skill learning (Logan, 1988; Palmeri, 1997) and models of dual-task performance that emphasize parallel processing (Logan, 2002; Meyer, Glass, Mueller, Seymour, & Kieras, 2001). Nevertheless, two important differences between the current results and those of Bajic and Rickard (2009) were observed: (1) the algorithm pause effect in this experiment (Figure 3.2A), while still large and highly significant, is less than half the magnitude of that in Bajic and Rickard (compare Figures 3.2A and 3.2B), and (2) partial-algorithm step completion on retrieval trials is much more frequent in this experiment (compare Figures 3.2C and 3.2D).

Also of note is the bimodal distribution of observed partial-algorithm steps in Figure 3.2C, with one mode at zero and the other mode at 3 steps. Bimodality typically signifies a mixture of observations from two distinct populations. In the current case, it appears to reflect the adoption by subjects of one of two distinct processing strategies on each retrieval trial: one in which the algorithm is entirely abandoned in favor of retrieval from the outset of the trial (yielding the mode of zero steps), and another in which the algorithm is initiated at some point during the trial but then aborted before completion in favor of retrieval (yielding the mode of 3 steps). Theoretical implications of these results will be considered after describing the closely related results for Experiment 2.

Experiment 2

The lack of random assignment of subjects in the above comparison of Experiment 1 to the experiment of Bajic and Rickard (2009) may compromise statistical conclusions about between-groups differences. Experiment 2 resolves this problem by randomizing assignment of subjects to groups. It also employs a different motor task to explore the generalizability of the results. Instead of key tapping, the motor task in this experiment required subjects to use the computer mouse to click alternately on target regions on the left and right side of the computer screen.

Method

Subjects

36 University of California at San Diego undergraduate students participated for course credit.

Design and Procedure

The simple tapping algorithm from Experiment 1 was replaced with a visuo-spatial task. This experiment included two randomized between-subjects conditions. The first (*click-only*) was a perceptual-motor task analogous to the tap-only task from Experiment 1. Subjects used the computer's mouse to click over the location of a rectangle that alternated between positions on the left and right side of the computer screen. The second group (*click-count*) was analogous to the tap-count task used by Bajic and Rickard (2009): namely, to find the answer for each item, the subject was required to count upward from the presented number each time that he or she clicked the rectangle on a given trial. The rectangle alternated its position in response to each click just as for the click-only group.

As in Experiment 1, this experiment consisted of a 1-block learning phase and a 45-minute training phase, with similar instructions prior to each phase. The Appendix lists the numerical visual stimuli, number of algorithm steps, and correct vocal responses for all 10 stimulus-response items. In contrast to Experiment 1 (but analogous to the experiment in Bajic & Rickard, 2009), the number of algorithm steps for both groups was always equal to the value of the correct response minus the numerical stimulus. Subjects in the click-only group, however, were not aware of this consistency, nor of the requirement to count for the click-count group, and thus had no reason to do any counting during algorithm execution.

Prior to each trial, a small textbox containing the words "Click Here to Begin" would appear just below the center of the screen, and clicking this box would immediately initiate the trial. (This ensured that the mouse pointer was always at roughly the same position at the start of each trial). During each trial, the screen would be divided into two equal halves by a thin vertical line down the center. At the beginning of each trial, a 1.5x1.5 cm square containing the numerical stimulus for that trial would appear at the center of the screen, and a green rectangle (6.5x11 cm) would appear on either the left or right half of the screen (randomly determined), centered vertically on the screen, with one edge positioned 2.5 cm away from the vertical line that divided the screen. This rectangle will henceforth be referred to as the *target.* If the target was clicked with the mouse pointer, it would immediately move to its equivalent position on the opposite half of the screen. Each mouse-click of the target constituted one algorithm step.

For subjects in the *click-only* group, the complete algorithm simply involved the repeated clicking of the target. After the full number of algorithm steps for a given trial had been provided, the mouse pointer and the trial's numerical stimulus would vanish from the screen. Simultaneously, the target rectangle would change from green to black, with white text inside it presenting both the word “STOP”, and the appropriate numerical answer for that trial's vocal response.

For subjects in the *click-count* group, finding the answer algorithmically required that the subject count upward from the value of the numerical stimulus each time he or she clicked the target rectangle. That is, if the numerical stimulus for a given trial were 21, the subject would silently count up to 22 the first time that he or she clicked the target square, then count to 23 the second time, and so forth. When the full set of algorithm steps had been provided, the same changes described above for the *click-only* group would occur, except that the white text would only present the word “STOP”, and not the answer itself.

Results and Discussion

Following the rules described for Experiment 1, data from four subjects (two in each group) were excluded due to a high rate of voice key errors; and data from two subjects in the click-count group and one subject in the click-only group were excluded due to low overall accuracy. Among the remaining subjects, voice key errors occurred on approximately 3.85% of trials in the click-count group, and approximately 4.24% of trials in the click-only group.

For subjects in the click-count group, mean accuracy was initially 89.31% on the first training block, fell to its lowest value (85.58%) on Block 10, then rose to 93.41% by the 22nd block, the furthest block that all subjects completed. The mean RT, the mean SD, and the proportion of trials on which the answer was retrieved, followed patterns similar to those depicted in Figure 3.1, and thus will not be depicted visually. Instead we provide summary results: the mean of the subject-level mean correct RTs (latency from stimulus presentation to vocal response) was

7340.57 ms in Block 1, falling to 1876.06 by Block 22. The mean of the subject level SDs was initially 1429.04 in Block 1, rose to its highest value (2491.1) on Block 8, then fell to 641.16 by Block 22. The proportion of trials in which subjects retrieved the correct answer rose to approximately 50% by Block 10, and approximately 95% by Block 22.

For subjects in the click-only group, mean accuracy was initially perfect (100%) on the first training block, fell to its lowest value (90.23%) on Block 6, then rose again to 98.67% by the 26th block, the furthest block that all subjects completed. The mean of the subject-level mean correct RTs (latency from stimulus presentation to vocal response) was 5825.53 ms in Block 1, then fell to 2095.63 ms by Block 26. The mean of the subject level SDs was 940.32 in Block 1, rose to its highest value (1854.16) on Block 10, then fell to 985 by Block 26. The proportion of trials in which subjects retrieved the correct answer rose to approximately 50% by Block 9, and approximately 93% by Block 26.

Algorithm step latencies on trials preceding the first correct retrieval

For each experimental group, synchronized blocks were computed using the procedures described for Experiment 1. Mean algorithm step latencies from Sync-block -5 through -1 are shown in Figures 3.3A (click-only) and 3.3B (click-count). A mixed factors ANOVA on the algorithm first step data (identical to that performed earlier) revealed significant effects of sync-block, $F(4, 108) = 16.5$, $p < .0001$, group, $F(1, 27) = 30.3$, $p < .0001$, and their interaction, $F(4, 108) = 2.47$, $p < .05$. The simple effects of sync-block were also confirmed in separate ANOVAs performed for each group: $F(4, 52) = 9.32$, $p < .0001$, for the click-count group; and $F(4, 56) = 8.91$, $p < .0001$, for the click-only group. There were no significant effects involving sync-block when this analysis was performed on the subsequent algorithm steps.

Partial-algorithm step completion on retrieval trials

This analysis was conducted using the procedures described for Experiment 1, and the corresponding bar graphs are shown in Figures 3.3C (click-only) and 3.3D (click-count). The Wilcoxon tests allow rejection of the race model for both groups: click-only: $T^{+} = 102$, $p < .0007$; click-count: $T^{+} = 105$, $p < .0002$. A χ^2 test of independence, comparing the observed distribution of completed algorithm steps on retrieval trials for the two groups, was again highly significant, $\chi^2(9) = 205.26$, $p < .0001$, indicating more algorithm step completion on retrieval trials for the click-only group than for the click-count group. The bimodality in Figure 3.3C replicates that observed in Figure 3.2C, as does the tendency for the observed distribution of completed algorithm steps (beyond 0 steps) to mimic somewhat the distribution predicted by the race model (albeit again with a left-shifted mode, as would be expected given the pause effect on partial-algorithm trials that is discussed below).

As depicted in Figure 3.3F, for the click-count group, partial-algorithm step latency difference scores based on comparison to the first training block (with positive scores indicating slowing relative to Block 1, and negative scores indicating speed-up) for Algorithm Steps 1-4 were: 607.6, -8.43, 128.31, and 62.16, respectively. There was significant slowing for Step 1, $t(9)=2.35$, $p < .05$. No subsequent steps reached significance. For the click-only group (see Figure 3.3E), partial-algorithm difference scores for Algorithm Steps 1-4 were: 459.49, 66.42, 43.05, and 13.06, respectively. There was significant slowing for Step 1, $t(14) = 3.15$, $p < 0.008$. No subsequent steps reached significance.

In the alternative analyses of Steps 2-4 on partial-algorithm trials that compared latencies for Sync-blocks -1 and 0 (see results of Experiment 1), there were again no significant effects for either group ($p > .05$ in each case).

In nearly all respects, the results of this experiment closely match those for the comparison of tap-only versus tap-count discussed in Experiment 1. Relative to the tap-count group, there was a shallower (but still robust) algorithm pause effect slope and a large increase in the frequency of partial-algorithm trials. A partial-algorithm pause effect was observed in both the click-only and the click-count groups, and the bimodal distribution in completed algorithm steps that was observed in the tap-only group was also observed in the click-only group.

Theoretical Development

The results for the click-count group confirm and strengthen the conclusion of Bajic and Rickard (2009) that strategy execution reflects a choice between retrieval and the algorithm first step for the general case in which the algorithm steps require retrieval from long-term memory, even for very simple retrievals such as counting. There is little or no evidence of parallel strategy execution for that general case.

The results for the tap-only and click-only groups of Experiments 1 and 2 also eliminate any straightforward parallel model (including the race model and simple versions of limited capacity parallel models) for the currently studied perceptual-motor algorithms, and most likely for the entire class of perceptual-motor algorithms. A simple limited capacity model that assumes constant capacity demand from both strategies throughout all stages of their execution can potentially explain the pause effects if they are considered in isolation, but it cannot explain the finding of no slowing for Algorithm Steps 2 and beyond (on both algorithm trials and partial-algorithm trials), nor does it naturally predict the bimodal distribution of completed algorithm steps on retrieval trials that was observed in both experiments.

A pure one-at-a-time strategy execution account, like that of CMPL, also encounters difficulty accommodating the overall pattern of results. It does not straightforwardly accommodate the substantially shallower slope in the algorithm first step latencies approaching the first correct retrieval for the tap-only and click-only algorithms relative to their counting algorithm counterparts. It is also questionable whether a one-at-a-time strategy execution model can plausibly account for the frequent partial-algorithm retrieval trials in the tap-only and click-only groups, as it would imply that subjects switched, mid-algorithm, to speaking the previously retrieved response on about half of the retrieval trials represented in Figures 3.2C and 3.3C (as opposed to about 10% of trials in Bajic & Rickard, 2009, and in the click-count group of Experiment 2). In the Bajic and Rickard (2009) experiment (and in the click-count group of Experiment 2), the counting process yielded results that gradually approached the correct answer. Thus, the counting algorithm progressively narrowed the set of candidate answers. On some trials this property of the algorithm may have increased the subjects' confidence in a previously retrieved answer, leading them to speak that answer before completing the algorithm. In the current experiments, however, the tap-only and click-only algorithms did not serve to narrow the set of candidate answers, eliminating that explanation of why subjects might shift back to speaking a previously retrieved response after starting the algorithm. Note also that the algorithm is easier and faster when there is no counting, which would presumably decrease any motivation to switch back to a previously retrieved response after initiating the algorithm. Even if one allows for the large number of partial-algorithm trials, purely one-at-a-time strategy execution predicts neither the bimodal distribution of partial-algorithm steps, nor the tendency for the distribution of completed algorithm steps (beyond 0 steps) to mimic the distribution expected by the race model.

The considerations above lead us to conclude that there is at least some degree of parallel strategy processing for the case of perceptual-motor algorithms. As a working model, we propose that both the perceptual-motor algorithm strategy and the retrieval strategy can be understood (sufficiently for current purposes) as involving two discrete, sequentially executed stages, which we will term the high attention (HA) and the low attention (LA) stages. For both strategies, the hypothesized HA stage occurs immediately after stimulus perception and corresponds to the processes of (1) selecting a task set to execute (i.e., "initiate the algorithm" or "initiate retrieval"), and (2) executing any subsequent processing during which the selected strategy requires continued attentional focus on the task set and on the stimulus. For the simple perceptual-motor algorithms explored here, we propose that the HA stage is brief (perhaps on the scale of about 100-200 ms), and can be understood as the triggering event for initiating a perceptual-motor algorithm that is already primed in working memory (e.g., repeated key tapping). We propose that the HA stage for memory retrieval lasts at least several hundred ms, a period during which sustained attentional focus on both the stimulus and the retrieval task set are needed to drive the retrieval process until it can proceed to completion without that input.

We propose that a processing bottleneck at the HA stage necessitates that subjects complete that stage for only one strategy at a time, resulting in a strategy initiation choice at the outset of each trial. Performance on the count algorithms and other algorithms that involve memory retrieval steps can be understood within the HA-LA model by assuming that even the LA stage for retrieval is sufficiently demanding on attentional resources that initiation of another retrieval is precluded until both the HA and LA stages of the ongoing retrieval are completed. That is, it may only be possible to retrieve one response at a time from long-term memory, as argued by Nino and Rickard (2003) and by Rickard and Bajic (2004, 2005). As such, strategy execution for algorithms that involve memory retrieval steps is a purely one-at-a-time phenomenon, consistent with the general principles of the CMPL model and with the data for the tap-count and click-count groups.

For perceptual-motor algorithms, however, we propose that the LA stage for either strategy can run in parallel with either the HA or the LA stage of the other strategy. The processing stage sequences for four strategy scheduling types that are consistent with the HA-LA model are summarized in Figure 3.4. Note that Scheduling Types 1 and 2 (a and b) prioritize the algorithm, in that it is initiated first, whereas Scheduling Types 3 (a and b) and 4 prioritize retrieval.

Initially, before retrieval is competitive, only the algorithm stages are executed on each trial (Scheduling Type 1 in Figure 3.4). On trials for which both strategies are roughly equally competitive (i.e., around the point of the strategy shift for each item) there are potentially two distinct scheduling types. One possibility is that subjects first initiate the HA stage for the algorithm, followed by parallel execution of the LA stage for the algorithm and the HA and LA stages for retrieval. The case in which the algorithm generates the response first is depicted by Scheduling Type 2a in Figure 3.4, and the case in which retrieval generates the response first is depicted by Scheduling Type 2b. Note that this scheduling type is efficient, in that it would yield no algorithm pause effects while also allowing the retrieval strategy a chance to generate the response first (Type 2b) as it becomes more competitive, with the only penalty being a brief (perhaps 200 ms or less) delay in retrieval initiation as the algorithm HA stage is executed.

The second possible scheduling type when the strategies are roughly equally competitive is that subjects first initiate the HA stage for retrieval and then switch to parallel execution of the LA stage for retrieval and of the HA and LA stages of the algorithm. If retrieval fails to deliver an answer before the algorithm is completed, or if subjects are not confident in the retrieved answer, then the algorithm runs to completion and produces the executed response (Scheduling Type 3a in Figure 3.4). The alternative outcome is that retrieval is successful and the subject is sufficiently confident to execute the retrieved response (Scheduling Type 3b), but possibly only after some algorithm steps have been completed.

Finally, after subjects become highly confident in the retrieval strategy, they execute the HA and LA stages of the retrieval strategy only (Scheduling Type 4 in Figure 3.4). Subjects would be expected to choose Type 4 even over the efficient Type 2b at this point because execution of the algorithm in Type 2b would delay the onset of the HA stage for retrieval. Related reasoning indicates that subjects should come to prefer Scheduling Type 4 to Type 3b.

Now consider how the HA-LA model can accommodate the main patterns in the data for perceptual-motor algorithms. First, the shallower (but still substantial and highly significant) algorithm pause effect slope for the perceptual-motor case compared to the counting case can be accommodated by a mixture of Scheduling Type 2a trials (yielding zero slope) and Scheduling Type 3a trials (yielding a slope similar to that of the counting algorithms). Second, the much larger percentage of retrieval trials with partial-algorithm step completion for the perceptual-motor algorithms (relative to the counting algorithms) is consistent with Scheduling Types 2b and 3b, neither of which can occur under our framework for the counting algorithms. Third, the pause effect on partial-algorithm trials is explained by

Scheduling Type 3b. The bimodality in the distribution of completed algorithm steps on retrieval trials is explained by a mixture of Scheduling Type 4 (yielding the mode of 0 in Figures 3.2C and 3.3C) and Types 2b and 3b (which are consistent with the mode of 2 or 3 steps shown in those figures). The tendency of the distribution of completed algorithm steps on retrieval trials to mimic (beyond the case of zero steps) the race prediction is accounted for because once the HA stage for the prioritized strategy is complete, the tapping and clicking sequences are run in parallel (perhaps as a race) with retrieval. Finally, the left-shift of the larger mode in Figures 3.2C and 3.3C reflects Scheduling Type 3b, in which algorithm step execution is delayed until the HA stage of retrieval is completed.

Subject-level Analyses Combining Experiments 1 and 2

The use of more efficient (i.e., Types 2a and 2b) versus less efficient (i.e., Types 3a and 3b) strategy scheduling is likely to reflect, in large part at least, individual differences rather than differences over trials within each subject. In other studies of strategy use, most subjects do not use all available strategies (e.g., Lemaire & Siegler, 1995; Romero, Rickard, & Bourne, 2006; Siegler, 1988). It would also place less strategy selection load on subjects if they adopt a general (i.e., prior to trials) rather than trial-specific (i.e., during the trial) scheduling approach.

Efficient schedulers would eschew Scheduling Type 3 in favor of Type 2. As retrieval begins to produce the answer first (i.e., as the transition from Type 2a to 2b occurs), these subjects would have evidence that they can successfully shift from prioritizing the algorithm to prioritizing retrieval. Given that these subjects would already have had successful retrieval experience at that point (through scheduling Type 2b), they may be inclined to shift directly from Scheduling Type 2 to Type 4, with relatively few trials on which Scheduling Type 3b is used; that is, these hypothetical subjects may feel that there is little need to initiate the algorithm after the HA stage of retrieval as insurance in case retrieval were to fail. Highly efficient subjects, then, may adopt a strategy scheduling sequence transition of 1-2a-2b-4. In the idealized case, they would exhibit zero pause effect slope and no partial-algorithm pause effect. On the other hand, some subjects may not recognize the efficiency advantage of Scheduling Type 2. Instead, when retrieval becomes competitive, they simply pause the algorithm as they attempt to retrieve (i.e., they use Scheduling Type 3). After sufficient practice, retrieval begins to succeed and they undergo a transition from Type 3a to 3b. Eventually they become sufficiently confident in retrieval that they do not execute any algorithm steps (shifting to Scheduling Type 4). Low efficiency subjects, then, would tend to adopt the scheduling sequence transition of 1-3a-3b-4.

Given the hypothesis of individual differences in scheduling efficiency outlined above, the HA-LA model implies a positive correlation over subjects between the magnitude of the pause effect slope on algorithm trials and the magnitude of the pause effect on partial-algorithm trials. Efficient scheduling should yield small values for both measures, whereas inefficient scheduling should yield large values for both measures. The combined data from the tap-only and click-only groups is shown in Figure 3.5, along with the best fitting linear prediction.

Necessarily, this graph only includes subjects who exhibited partial-algorithm trials, and to reduce noise effects we further limited it to subjects who had at least 5 partial-algorithm trials. The slope of this fit is highly significant, t(30) = 8.51, p < .0001, whereas the intercept does not significantly differ from zero (p > .05). This result, interpreted within the context of the HA-LA model, indicates that some subjects use efficient scheduling and others inefficient scheduling.

The data for the highly efficient subjects shown in Figure 3.5 (lower left data points) are consistent with the HA-LA model, but are also consistent with a simple race model. For that subset of subjects, the pause effect data offer no evidence (or little evidence) of any strategy interference on either algorithm or partial-algorithm trials. The HA-LA and race accounts for that subset of subjects can be discriminated, however, by analysis of the distribution of completed algorithm steps on retrieval trials (i.e., the data depicted in Figures 3.2C and 3.3C). The race account is consistent with a unimodal distribution that has a shape matching that of the race prediction as derived earlier. The HA-LA model, on the other hand, predicts an additional mode at zero steps (because efficient subjects are likely to select Scheduling Type 4 over Type 2b after sufficient practice, as discussed earlier), just as was the case in Figures 3.2C and 3.3C. Additional analyses were conducted for the most efficient subjects shown in Figure 3.5, defined as those subjects with a pause effect slope value of less than 50, and a partial-algorithm pause effect less than the predicted value at that point (boundary conditions delimited by the dashed line in Figure 3.5). Twelve subjects met the criteria for inclusion in this analysis. For this subset of subjects, the distribution of completed algorithm steps in Sync-blocks 0-4 was again bimodal, with a major mode at 0 steps (approximately 17% of trials), and a minor mode at 4 steps (approximately 14% of trials). The Wilcoxon test on the subjects' mean difference scores (the mean of the expected minus the observed number of completed steps) was significant, $T^+ = 63$, $p < .005$.

As an alternative (but in our view implausible) account of the results shown in Figure 3.5, it is in principle possible that there are profound individual differences in cognitive architecture, such that some subjects can execute these strategies to a large extent in parallel whereas others can not. We know of no precedent, however, that suggests such differences in any of the related literature on dual-task performance or attention. Our account in terms of individual differences in efficiency of strategy scheduling is, on the other hand, highly plausible in light of numerous demonstrations in the literature of subject differences in strategy selection over a variety of tasks (Lemaire & Siegler, 1995; Romero et al., 2006; Siegler, 1988).

General Discussion

We can now draw several well-grounded conclusions about the temporal dynamics of strategy execution in cognitive skill learning. First, there is an apparently complete bottleneck in strategy execution when the algorithm involves memory retrieval, even for the simplest retrieval operations such as counting. For this class of tasks, the algorithm first step competes with retrieval and only one of those strategies is executed at a time until completed.

Second, most subjects exhibit substantial strategy interference even for simple perceptual-motor algorithms involving no memory retrieval. As noted earlier, both the race and the simplest case limited capacity versions of parallel strategy execution models can be eliminated for the tasks studied here. It is questionable whether such models could provide sufficient accounts of *any* task exhibiting a shift from algorithm to retrieval, regardless of the properties of the algorithm.

Third, strategy execution for the case of perceptual-motor algorithms is generally well accounted for by the new model proposed here, the HA-LA model (see the Discussion of Experiment 2 for a detailed description), which proposes a strategy processing bottleneck at an early, high-attention stage of processing. The HA-LA model is simple at its core, yet empirically powerful. It accommodates a number of different strategy scheduling types (see the Scheduling Types listed in Figure 3.4) that appear to be represented in the data. Our conclusion that these scheduling types are all used is of course based on inference rather than direct observation. Nevertheless, there does not appear to be an alternative model that can both explain the complex patterns in the data and incorporate a similar level of simplicity in its core assumptions.

Anecdotally, it is not uncommon for people to pause briefly and attempt to remember, say, where their keys are before searching, or to pause and try to remember a key combination for a computer command before initiating a search of the dropdown menus. Our results suggest that these palpable experiences reflect fundamental limitations in parallel processing of memory retrieval with simple perceptual-motor algorithms, combined with commonly occurring inefficiencies in strategy scheduling.

Our results leave open the question of why some subjects would adopt efficient strategy scheduling whereas others would adopt inefficient scheduling. One interesting possibility is that efficient subjects initiate the algorithm as soon as they detect stimulus onset but prior to stimulus identification. (In the case of the perceptual-motor algorithms, initiation of the algorithm does not depend on which stimulus is presented, whereas initiation of the memory retrieval strategy does.) For these subjects, stimulus identification can then take place during the HA stage of the algorithm (our HA-LA model does not preclude the possibility of perceptual stimulus processing operating in parallel with the HA stage of either strategy). By the time stimulus identification runs to completion, the HA stage of the algorithm is also partially completed, so the delay in initiation of the HA stage for retrieval (Scheduling Type 2) may be quite brief for these subjects.

Inefficient subjects, on the other hand, may wait for stimulus identification to occur before initiating either strategy. Once they identify the stimulus, its familiarity (after several trials) may trigger a feeling-of-knowing response that promotes prioritization of retrieval (i.e., Scheduling Type 3). This account is particularly plausible in light of the demonstration by Reder and Ritter (1992) that feeling of knowing is driven by familiarity with the stimulus rather than by familiarity with the answer. To the extent that the feeling of knowing is not well calibrated in this circumstance, the retrieval attempt would sometimes fail for these subjects (or subjects would not have sufficient confidence to respond based on the retrieved answer), giving rise to use of the algorithm as a back-up strategy and to the observed pause effects on algorithm and partial-algorithm trials. Further implications will be discussed in Chapter 4.

References

Bajic, D., & Rickard, T. C. (2009). The temporal dynamics of strategy execution in cognitive skill learning. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *35*(1), 113-121.

Delaney, P. F., Reder, L. M., Staszewski, J. J., & Ritter, F. E. (1998). The strategy-specific nature of improvement: The power law applies by strategy within task. *Psychological Science*, *9*(1), 1-7.

Hertzog, C., Touron, D. R., & Hines, J. C. (2007). Does a time-monitoring deficit influence older adults' delayed retrieval shift during skill acquisition? *Psychology and Aging*, *22*(3), 607-624.

Jenkins, L., & Hoyer, W. J. (2000). Instance-based automaticity and aging: Acquisition, reacquisition, and long-term retention. *Psychology and Aging*, *15*(3), 551-565.

Lemaire, P., & Siegler, R. S. (1995). Four aspects of strategic change: Contributions to children's learning of multiplication. *Journal of Experimental Psychology: General*, *124*(1), 83-97.

Logan, G. D. (1988). Toward an instance theory of automatization. *Psychological Review*, *95*(4), 492-527.

Logan, G. D. (1992). Shapes of reaction-time distributions and shapes of learning curves: A test of the instance theory of automaticity. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *18*(5), 883-914.

Logan, G. D. (2002). An instance theory of attention and memory. *Psychological Review*, *109*(2), 376-400.

Luce, R. D. (1991). *Response times: Their role in inferring elementary mental organization*. Oxford University Press, USA.

Meyer, D. E., Glass, J. M., Mueller, S. T., Seymour, T. L., & Kieras, D. E. (2001). Executive-process interactive control: A unified computational theory for answering 20 questions (and more) about cognitive ageing. *European Journal of Cognitive Psychology*, *13*(1-2), 123-164.

Nino, R. S., & Rickard, T. C. (2003). Practice effects on two memory retrievals from a single cue. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *29*(3), 373-388.

Onyper, S. V., Hoyer, W. J., & Cerella, J. (2006). Determinants of retrieval solutions during cognitive skill training: Source confusions. *Memory & Cognition*, *34*(3), 538-549.

Palmeri, T. J. (1997). Exemplar similarity and the development of automaticity. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *23*(2), 324-354.

Rawson, K. A. (2004). Exploring automaticity in text processing: Syntactic ambiguity as a test case. *Cognitive Psychology*, *49*(4), 333-369.

Reder, L. M., & Ritter, F. E. (1992). What determines initial feeling of knowing? Familiarity with question terms, not with the answer. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *18*(3), 435-451.

Rickard, T. C. (2004). Strategy execution in cognitive skill learning: An item-level test of candidate models. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *30*(1), 65-82.

Rickard, T. C. (1997). Bending the power law: A CMPL theory of strategy shifts and the automatization of cognitive skills. *Journal of Experimental Psychology: General*, *126*(3), 288-311.

Rickard, T. C. (1999). A CMPL alternative account of practice effects in numerosity judgment tasks. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *25*(2), 532-542.

Rickard, T. C., & Bajic, D. (2003). Automatic mediation or absence of mediation? Commentary on Crutcher and Ericsson (2000). *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *29*(6), 1381-1386.

Rickard, T. C., & Bajic, D. (2006). Cued recall from image and sentence memory: A shift from episodic to identical elements representation. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *32*(4), 734-748.

Rickard, T. C., & Bajic, D. (2004). Memory retrieval given two independent cues: Cue selection or parallel access? *Cognitive Psychology*, *48*(3), 243-294.

Rickard, T. C., & Bajic, D. (2005). In search of cross-talk facilitation in a dual-cued recall task. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *31*(4), 750-767.

Rogers, W. A., Hertzog, C., & Fisk, A. D. (2000). An individual differences analysis of ability and strategy influences: Age-related differences in associative learning. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *26*(2), 359-394.

Romero, S. G., Rickard, T. C., & Bourne, L. E. (2006). Verification of multiplication facts: An investigation using retrospective protocols. *American Journal of Psychology*, *119*(1), 87-120.

Schunn, C. D., Reder, L. M., Nhouyvanisvong, A., Richards, D. R., & Stroffolino, P. J. (1997). To calculate or not to calculate: A source activation confusion model of problem familiarity's role in strategy selection. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *23*(1), 3-29.

Siegler, R. S. (1988). Strategy choice procedures and the development of multiplication skill. *Journal of Experimental Psychology: General*, *117*(3), 258-275.

Touron, D. R., Hoyer, W. J., & Cerella, J. (2001). Cognitive skill acquisition and transfer in younger and older adults. *Psychology and Aging*, *16*(4), 555-563.

Appendix

Phase 2 Stimulus-Response Pairings

Stimuli	Responses	Algorithm Steps
20	34	14
21	36	15
22	32	10
23	35	12
24	33	9
25	37	12
26	41	15
27	40	13
28	39	11
29	38	9

Note: For Experiment 1, the Algorithm Steps listed above were randomly reassigned to the various stimulus-response pairings at the start of the experiment, following randomization constraints listed in the Method section of that experiment.

Author Note

We would like to thank Kristin West, Lisa Graves, Jennifer Um, Shivali Agarwal, Samir Patel, Jon Strunk, and Caitlin Oldenkamp for their assistance with data collection.

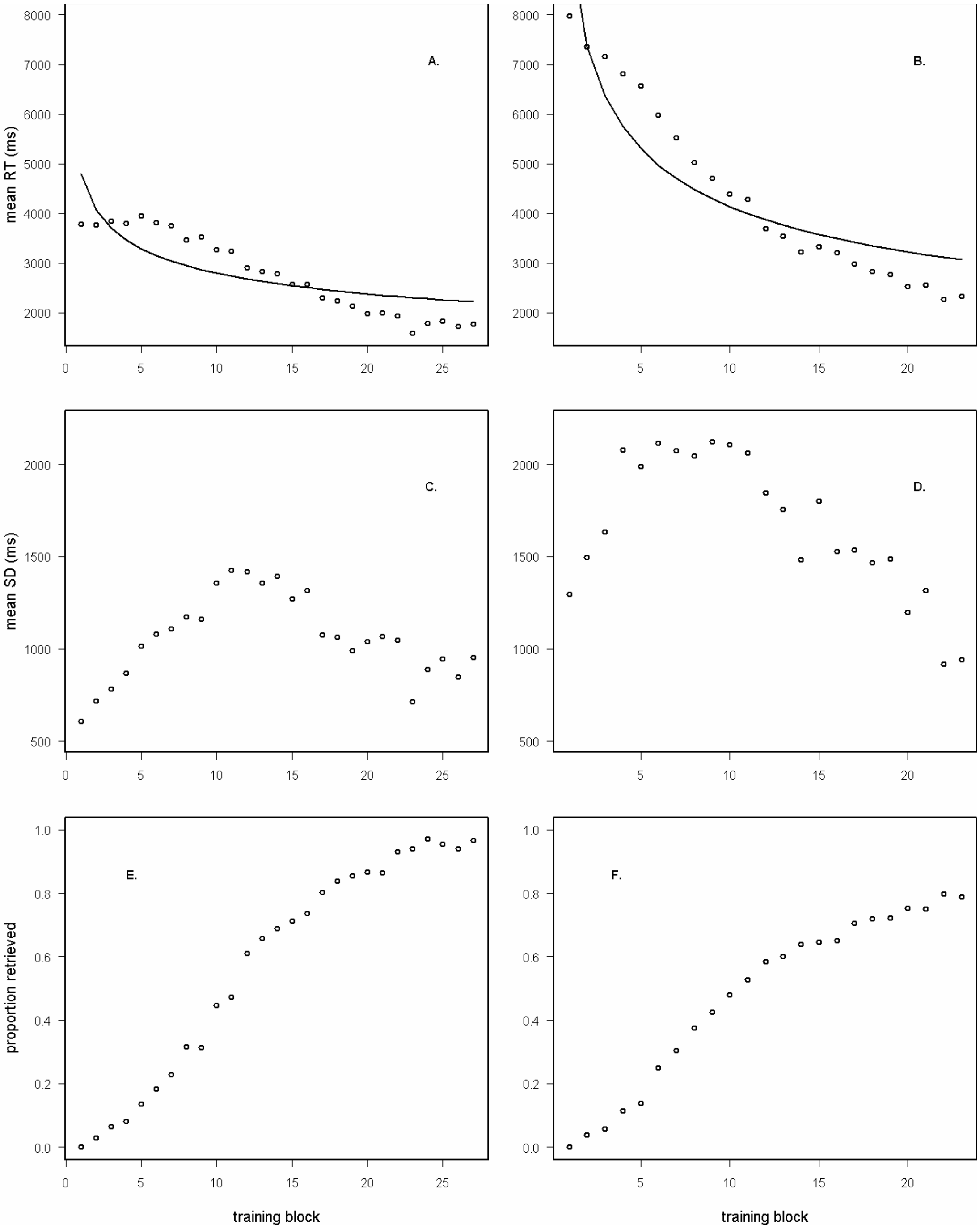

Figure 3.1: Mean response time with best fitting three-parameter power functions (panels A and B), mean standard deviation (C and D), and the proportion of trials on which the direct retrieval strategy was selected (E and F), as functions of training block, for Experiment 1 (panels A, C, and E, respectively), and the experiment in Bajic and Rickard (2009) (B, D, and F).

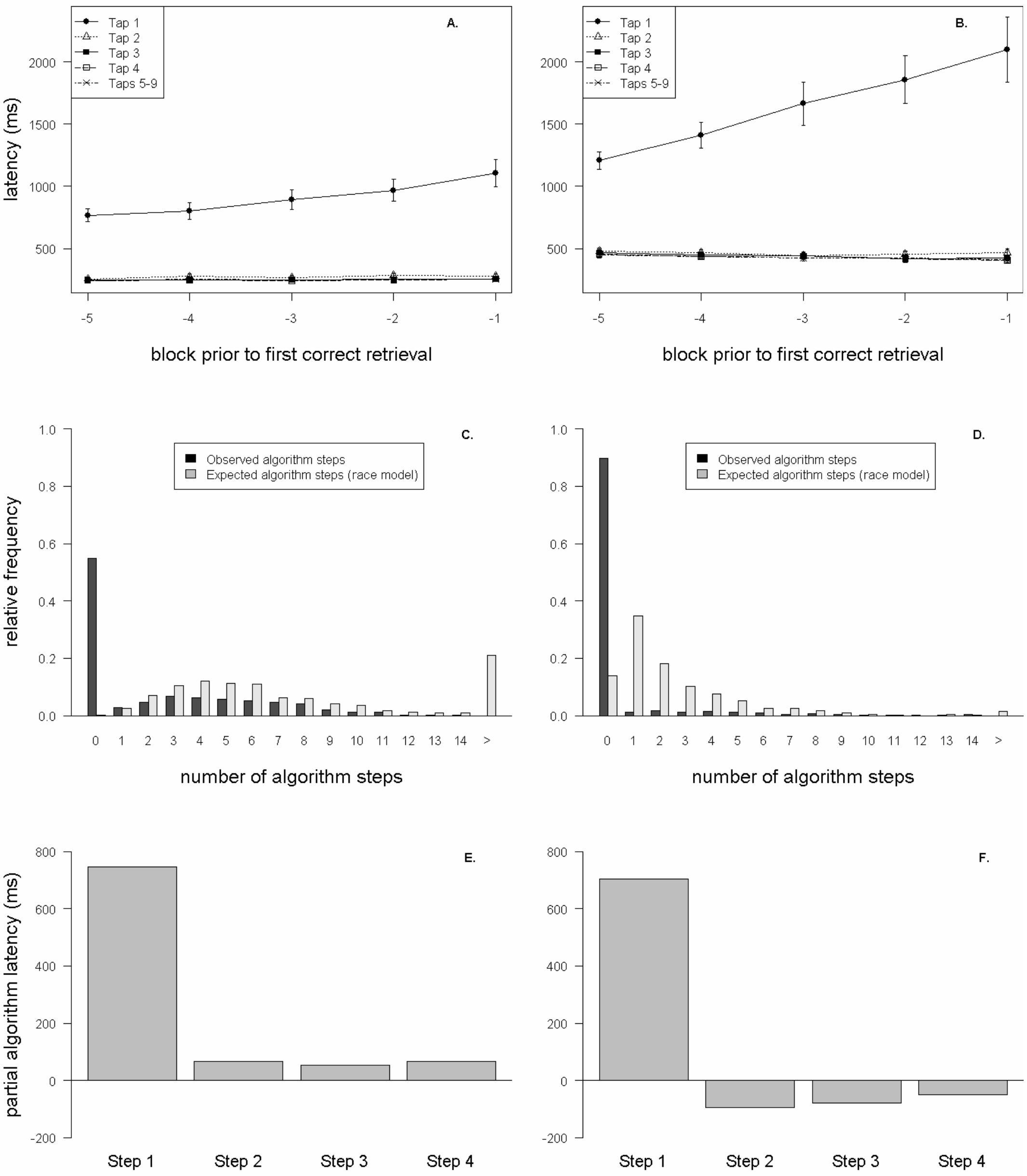

Figure 3.2: Mean algorithm step latencies on the five blocks prior to the first correct retrieval block for Algorithm Steps 1, 2, 3, 4, and the mean of 5 through 9 (panels A and B), relative frequency bar charts of expected (according to a race model) and observed completion frequencies for each algorithm step on the first five correct retrieval trials (C and D), and partial-algorithm latency difference scores (Sync-blocks 0-4 vs. Block 1) for Steps 1-4 (E and F), for Experiment 1 (A, C, and E, respectively) and the experiment in Bajic and Rickard (2009) (B, D, and F). For panels A and B, error bars represent the between subjects standard error, computed independently for each sync-block.

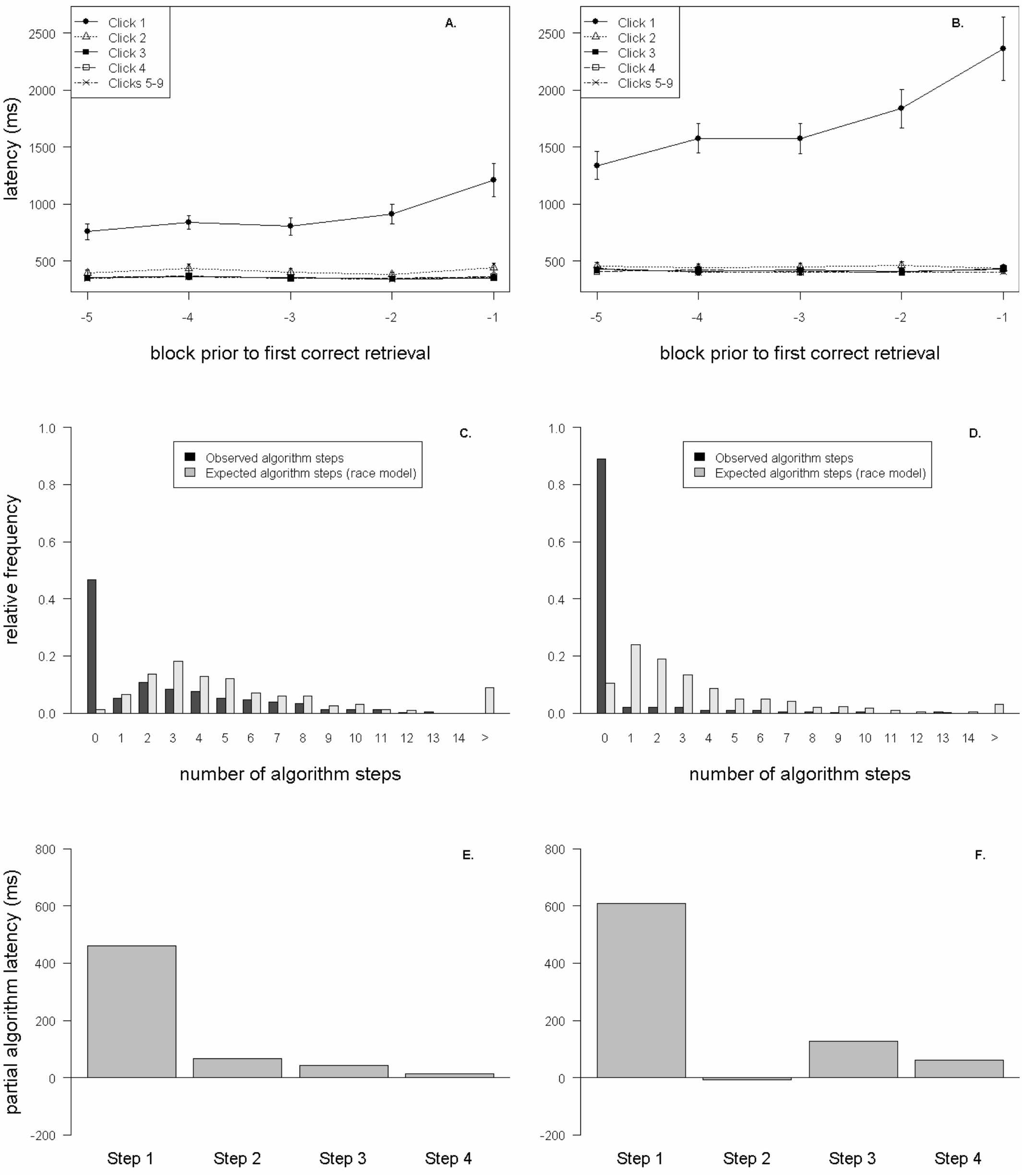

Figure 3.3: Mean algorithm step latencies on the five blocks prior to the first correct retrieval block for Algorithm Steps 1, 2, 3, 4, and the mean of 5 through 9 (panels A and B), relative frequency bar charts of expected (according to a race model) and observed completion frequencies for each algorithm step on the first five correct retrieval trials (C and D), and partial-algorithm latency difference scores (Sync-blocks 0-4 vs. Block 1) for Steps 1-4 (E and F), for the click-only group (A, C, and E, respectively) and the click-count group (B, D, and F) of Experiment 2. For panels A and B, error bars represent the between subjects standard error, computed independently for each sync-block.

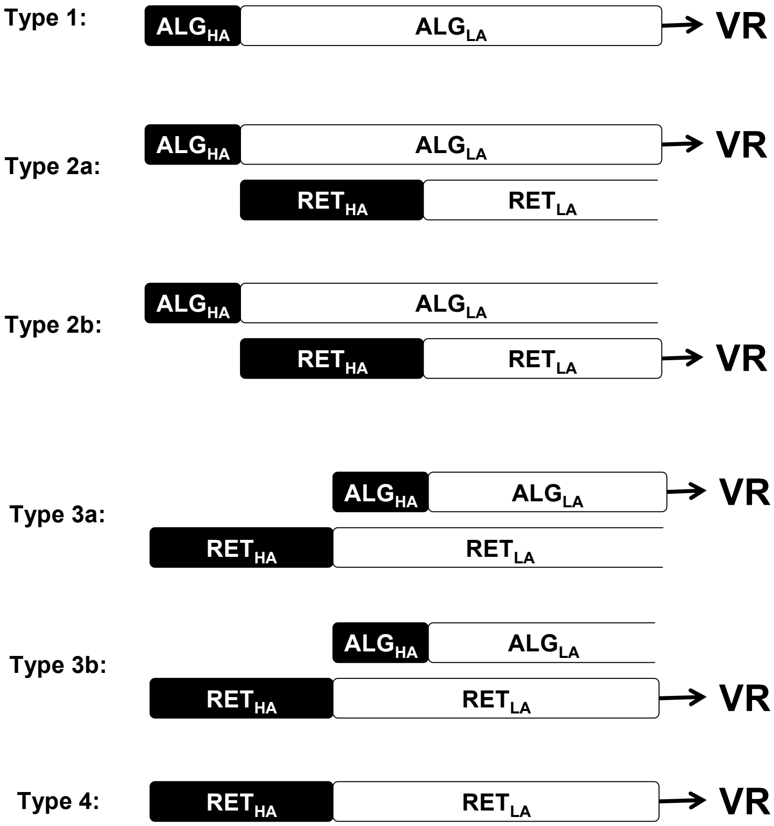

Figure 3.4: Hypothetical strategy scheduling types for competing perceptual-motor algorithm and retrieval strategies. RET = retrieval; ALG = algorithm; HA = high-attention; LA = low-attention; VR = vocal response. The right-side open-ended rectangles represent the strategy that does not drive the vocal response, either because it fails to generate a response on that trial or because the subject is not sufficiently confident to execute the response generated by that strategy.

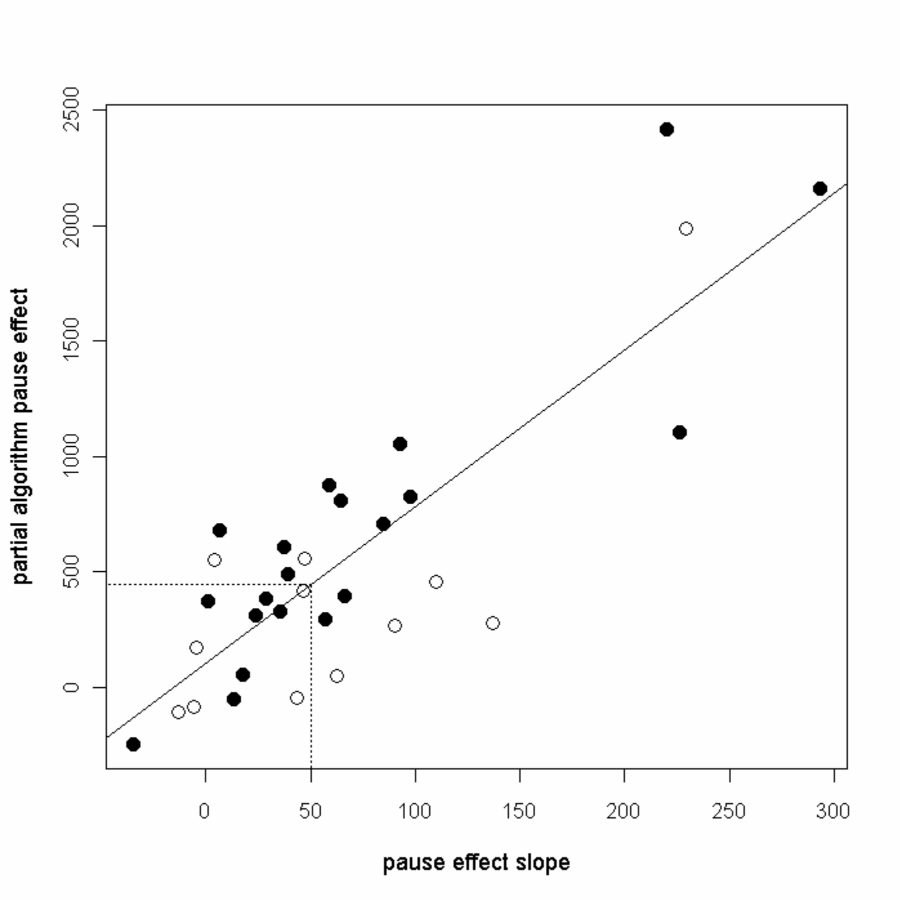

Figure 3.5: Scatterplot, with best linear fit, of partial-algorithm pause effect as a function of the pause effect slope, for the combined tap-only (black dots) and click-only (white dots) data. The 12 subjects in the lower left corner (within the dashed box) demonstrated the most efficient strategy scheduling.

Chapter 3, in part, has been submitted for publication of the material as it may appear in Bajic, D., & Rickard, T. C. (2009). Temporal dynamics of strategy execution in cognitive skill learning: The case of simple perceptual-motor algorithms. The dissertation author was the primary investigator and author of this paper.

Chapter 4

Conclusions

In Chapter 1, I described three problems in the skill acquisition literature: (1) Previous paradigms have been limited in their capacity to provide accurate data regarding strategy use on each trial, and have provided no means of collecting reliable data regarding partial algorithm trials. (2) Previous paradigms have provided no means of measuring the latency of each individual step of an algorithm. And (3) the problem of appropriately and effectively aggregating data when the learning curves for individual items are not naturally synchronized.

In the preceding chapters, I have introduced fruitful solutions to all of the above problems. The original paradigm that I introduced in the preceding chapters is the first that permits the collection of objective data regarding strategy use in every trial, with no need to ask subjects for retrospective reports (e.g., Compton & Logan, 1991), and no need to make indirect inferences about strategy use (e.g., Siegler, 1988). In addition, this paradigm is the first that permits the systematic investigation of partial-algorithm trials, and also the first that allows the latency of each individual algorithm step to be measured.

With accurate data on strategy usage collected for every trial, it is possible to synchronize the learning curves for all items at the point of each item's initial transition to direct-retrieval (i.e., sync-block analysis), thus providing a solution to the data analysis problem described above.

Sync-block analysis, combined with latency data for individual algorithm steps, revealed two previously undiscovered skill acquisition phenomena: (1) the pause effect slope for the first algorithm step on trials immediately preceding the first correct direct retrieval for each item, and (2) the partial-algorithm pause effect on trials following the first direct retrieval.

As noted in Chapter 1, an additional goal of the present work was to explore and compare two general classes of tasks that exhibit the shift to retrieval: namely, tasks with algorithms that require the retrieval of information from long-term memory (LTM), and tasks with algorithms that do not require LTM retrieval—i.e., those with simple perceptual-motor algorithms. Chapter 3 included the first direct, controlled comparison of performance for tasks with LTM-retrieval-based algorithms versus tasks with perceptual-motor algorithms, utilizing algorithms designed to be identical in all ways *except* in regard to whether LTM retrieval was needed. For each type of task, the results were uniquely consistent with a strategy choice mechanism involving a competition between the retrieval strategy and the first step of the algorithm. Regardless of whether a task involved a retrieval-based algorithm or a simple perceptual-motor algorithm, there was evidence of a pause effect slope preceding the transition to the direct-retrieval strategy, and a partial-algorithm pause effect following the transition.

The collection of data regarding partial-algorithm trials (impossible in previous paradigms) provided critical insights differentiating performance for tasks with each type of algorithm. In tasks with algorithms that consisted of multiple retrievals from LTM, partial-algorithm trials were very rare. In tasks with simple perceptual-motor algorithms, partial-algorithm trials were more common, although there was still a heavy bias toward abandoning the algorithmic strategy altogether once the direct-retrieval strategy became competitive. Neither result is consistent with models that assume parallel strategy execution (e.g., Logan, 1988; Nosofsky & Palmeri, 1997). The result in the case of tasks with retrieval-based algorithms is predicted by the CMPL model (see Rickard, 1997, 2004; Rickard & Bajic, 2004), which assumes that the direct-retrieval strategy can not be executed in parallel with the LTM-retrieval for any step of the algorithm. No previous models predicted the full set of results observed for the tasks with perceptual-motor algorithms: that is, the partial-algorithm data described above, as well as the pause effect slope and the partial-algorithm pause effect. However, as argued in Chapter 3, the full set of results observed for the tasks with simple perceptual-motor algorithms can be accounted for by a model that assumes an early-stage strategy execution bottleneck, with some parallel performance possible on later steps of the algorithm. In Chapter 3, I introduced such a model: the HA-LA model. Incorporating very few assumptions, the HA-LA model is capable of accounting for the full range of results observed for all tasks and algorithms described in the preceding chapters.

A novel contribution of the HA-LA model that goes beyond skill learning proper is the proposal of an initial phase of high-attention (HA) demand that exists not just for relatively effortful and time consuming cognitive processes such as LTM retrieval, but also (with at least a brief duration) for very simple processes, such as initiation of repeated tapping of the same key (Chapter 3, Experiment 1). At least for the case of strategy execution discussed here, this HA stage constitutes a processing bottleneck. An intriguing possibility is that a similar HA stage and bottleneck occur for any type of goal-directed cognitive process that is triggered by stimulus onset. Here, a potentially instructive connection can be made with the central bottleneck theory that has been developed in the dual-task literature (see Pashler, 1994; Welford, 1952). From the perspective of the dual-task literature, the direct-retrieval strategy from the preceding experiments can be thought of as a choice response time (RT) task. That is, there are multiple stimuli associated with multiple responses, making necessary a response-selection stage of cognitive processing on each trial. The tap-only algorithm, in contrast, could potentially be regarded as a simple RT task. That is, the onset of any stimulus is associated with a single response—tapping. From the perspective of some influential models of task processing (e.g., Donders, 1969), simple RT tasks do not involve a response-selection stage of processing.

The most common finding within the dual-task literature is that of a bottleneck in task processing that interferes with the ability to perform two independent tasks concurrently, even if there is no interference at the perceptual or motor stages of performance for the two tasks (Pashler, 1994). Rather, this bottleneck occurs in central stages of task processing, and is most widely believed to reflect a structural inability to simultaneously perform response-selection for two different tasks (Pashler, 1994). However, dual-task interference can be observed not just in concurrent choice RT tasks, but also in concurrent simple RT tasks (e.g., Telford, 1931; Welford, 1952), despite the fact that, as noted, simple RT tasks are widely assumed to lack a response-selection stage of processing. This has led some to speculate that the central bottleneck may not be limited to just the response-selection stages of task processing (see Pashler, 1994). Within the experiments of Chapter 3, to the extent that a perceptual-motor strategy such as the tap-only algorithm can be regarded as a simple RT task, then the finding of HA stage interference between the direct-retrieval and perceptual-motor strategies is consistent with the notion that central interference does not require two concurrent choice RT tasks. Viewing the dual-task literature from the perspective of the HA-LA model, it may be that the interference observed for concurrent simple RT tasks likewise reflects a bottleneck at the early, HA stage of processing. It is important to note, however, that standard dual-task designs possess some notable differences relative to the experiments described in the preceding chapters (e.g., two stimuli and two responses on each trial of a standard dual-task design, compared to one stimulus and one response—but two possible strategies—on each trial of the current studies), so we should be cautious in making cross-paradigm comparisons. However, the present studies suggest some interesting prospects for future synthesis.

An even stronger case for synthesis can be made in regard to the HA-LA model and the CMPL model, which could quite easily be integrated. As was noted in Chapter 1, the CMPL model, as currently specified, does not make any predictions regarding algorithms with no LTM retrieval component. However, the CMPL model could be extended to tasks with simple perceptual-motor algorithms, if we incorporate the HA-LA model's assumption of a bottleneck not just between competing LTM retrievals, but also between the early, HA stage of processing for the competing strategies.

As was noted earlier, the experiments in the preceding chapters were the first that have provided a means of systematically studying partial-algorithm trials. In the comparisons between tasks with retrieval-based algorithms and those with simple perceptual-motor algorithms, the collection of data on partial-algorithm trials provided critical insight, as one of the most distinctive differences between the two types of tasks was the greater frequency of partial-algorithm trials in the tasks with simple perceptual-motor algorithms. This highlights the fact that partial-algorithm trials are an issue that the skill acquisition literature can not afford to continue ignoring. For the tasks with simple perceptual-motor algorithms, there was an initial HA stage bottleneck, followed by parallel processing thereafter. Sloppy analysis of tasks such as these could make it appear either that there is no parallel processing (if the later algorithm steps are ignored), or that there is no initial bottleneck (if the slowing of the first step is ignored). By considering the full picture, though, we can discern possibilities for such tasks that are much more interesting: some parallel processing, with a slight delay at the start, along with the possibility of reducing the duration of this initial delay, if we schedule our strategies more efficiently.

These, then, are the temporal dynamics of strategy execution.

References

Compton, B. J., & Logan, G. D. (1991). The transition from algorithm to retrieval in memory-based theories of automaticity. *Memory & Cognition*, *19*(2), 151-158.

Donders, F. C. (1969). On the speed of mental processes. *Acta Psychologica*, *30*, 412-431.

Logan, G. D. (1988). Toward an instance theory of automatization. *Psychological Review*, *95*(4), 492-527.

Nosofsky, R. M., & Palmeri, T. J. (1997). An exemplar-based random walk model of speeded classification. *Psychological Review. Vol 104(2)*, 266-300.

Pashler, H. (1994). Dual-task interference in simple tasks: Data and theory. *Psychological Bulletin*, *116*(2), 220-244.

Rickard, T. C. (1997). Bending the power law: A CMPL theory of strategy shifts and the automatization of cognitive skills. *Journal of Experimental Psychology: General*, *126*(3), 288-311.

Rickard, T. C. (2004). Strategy execution in cognitive skill learning: An item-level test of candidate models. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *30*(1), 65-82.

Rickard, T. C., & Bajic, D. (2004). Memory retrieval given two independent cues: Cue selection or parallel access? *Cognitive Psychology*, *48*(3), 243-294.

Siegler, R. S. (1988). Strategy choice procedures and the development of multiplication skill. *Journal of Experimental Psychology: General*, *117*(3), 258-275.

Telford, C. W. (1931). The refractory phase of voluntary and associative responses. *Journal of Experimental Psychology*, *14*(1), 1-36.

Welford, A. T. (1952). The 'psychological refractory period' and the timing of high-speed performance—a review and a theory. *British Journal of Psychology*, *43*, 2-19.