Modeling Adaptivity in a Dynamic Task

Bradley J. Best, Christian D. Schunn, and Lynne M. Reder
(bbest@andrew.cmu.edu, schunn@cmu.edu, and reder@cmu.edu)
Carnegie Mellon University
Department of Psychology
Pittsburgh, PA 15213 USA

Abstract

Adaptivity is examined within a complex task environment: the Kanfer-Ackerman Air Traffic Controller Task©. A computational model is developed in ACT-R to account for such adaptivity using an implicit learning mechanism.

People demonstrate considerable flexibility in adapting their strategies to the changing nature of the environment. Previous studies (Reder, 1987; Siegler, 1988; Lovett and Anderson, 1996) focused on adaptive strategy selection in the context of simple tasks. For example, Lovett and Anderson (1996) applied a model of adaptive strategy choice to the Building Sticks Task, an isomorph to the Luchins water jug task. In the BST they manipulated the success of the alternative strategies over time. Their model accounted for the strategy choices of subjects by using the success and failure history of the available strategies and picking the more successful strategy. Noise added to this selection process allowed the model to sample the less successful strategy in proportion to its relative success rate, a commonly observed feature of human choice data.

Although these efforts have succeeded with simple tasks, it is possible that complex tasks may place different demands on people and make it difficult to capture the essence of their behavior in a complex task within this type of simple computational framework. This paper reports an effort to model adaptive strategy choice within a complex task, the Kanfer-Ackerman Air Traffic Control Task© (KA-ATC, Ackerman and Kanfer, 1994). In addition, we investigate assumptions about monitoring of strategy choice.

Many models of strategy choice are commonly thought of as explicit models of choice, where the choice is controlled through explicit metacognitive monitoring. Another possibility proposed by Reder and Schunn (1996) is that strategy choice is made through implicit memory and implicit learning. In this case the strategy itself may be explicit, but the mechanism of choosing between these overt strategies is assumed to be implicit. Alternatively, the strategy and choice may both be implicit, and people may be unaware of the strategies they used as well as their shifts between them (e.g., Reder, 1987). In all these situations, learning of the new strategy involves success or failure with the strategy or *blame assignment.*

Due to the complexity of the task environment in the current study, blame assignment is a central issue. In a simple task, strategy choice leads to immediate consequences. In the KA-ATC task, poor strategy choices may not cause immediate problems, and may instead lead to difficulties several moves later.

Blame assignment is a fundamental problem in artificial intelligence and has been explored within many frameworks. One of the simplest representations of the problem of optimal choice is the two-armed bandit. In this problem, the goal is to determine the optimal payoff of a choice between two options, say A and B, where the possible payoffs are 0 or 1. Choices A and B are random variables with a fixed mean and variance about which we have no initial information. If we know the mean payoff of A is higher than B then the optimal strategy is to always select A. Since we do not know which strategy will be more successful, we must test both. Further, no finite number of samples of either strategy can completely determine the strategy with the higher mean payoff. Trials therefore have two functions: information gathering and payoff accumulation. Optimal choice is a tradeoff between collecting enough information to determine the more successful strategy, and exploiting the more successfully strategy to maximize the overall payout. Too little sampling of both strategies can make the less successful strategy look more successful and result in selection of the less successful strategy. Excessive sampling results in too many trials of the less successful strategy (Holland, 1992).

The current task is a more general case of this problem which is complicated by a) changing relative payoffs of different strategies over time, b) possible multiple strategies available at each step, c) the difficulty in determining what constitutes a success or failure, and d) delays in finding out whether a choice was successful or unsuccessful.

The Kanfer-Ackerman ATC Task©

The task was designed to simulate some difficult aspects of air traffic control. It presents the subject with a dynamic environment in which they must attend to changing weather, different combinations of plane types and landing restrictions, time pressure, and other real-time factors.

The stated goal of the KA-ATC task is to accumulate as many points as possible across the trials of the session. Points are accumulated by landing planes without breaking rules. Rule infractions result in point deductions and the amount of points deducted depends on the severity of the infraction (crashing a plane due to low fuel is more severe than attempting to land on an illegal runway). The KA-ATC interface has three major screen regions: the hold area, the weather area, and the runway area. The hold area consists of 12 hold positions (North/East/South/West x 3 levels), each of which can hold one plane. When a hold position is occupied, it indicates the flight number, the type of plane (747, DC-10, 727, or Prop) and the remaining fuel (in minutes). The weather area shows the current wind speed, wind direction, and ground conditions for the runways. The runway area consists of two north/south runways, one long and one short, and two east/west runways, one long and one short. When planes are landed they slowly move across the runway over a 15 second period.

The KA-ATC has a concise set of rules that govern legality of landings and other plane moves:

1. Planes can move only 1 hold level at a time, and only into an unoccupied hold slot.
2. Planes can only land from hold level 1.
3. Planes must land into the wind (e.g., in a north wind the plane must land on the north/south runway.
4. Planes with low fuel (less than 3 minutes remaining) must be landed immediately.
5. Only one plane at a time can occupy a runway.
6. All planes can always land on the long runway. The current weather and plane type determine whether the short runway is legal. 747s can never land on the short runway. DC-10s can land on the short runway when the runways are not icy and the wind speed is less than 40 knots. 727s can land on the short runway when the wind speed is 0-20 knots or when the runway is dry.

An important aspect of this task is that some feedback about strategy success is immediate, while some feedback is delayed. For example, if a subject chooses to land a 727 on the long runway, that runway will be temporarily unavailable. Although the 727 lands successfully, it may prevent the landing of 747s that are low on fuel, resulting in fewer landings (because of suboptimal runway usage) and possibly even crashes. On the other hand, violating the rules by attempting to land a DC-10 on an icy runway causes a popup window to provide immediate feedback.

To model this assignment of blame backwards through time, blame is assigned to all actions on the causal path for the current goal and subgoals. It is possible that the causal events in this task are sufficiently remote in time that a current predicament cannot reach back to the true cause to assign blame. In this case we would expect to be unable to demonstrate a model that adapts to the structure of the KA-ATC task. On the other hand, if the causal events are sufficiently proximal to a success or failure, we would expect that a model could adapt to changing task conditions.

The aspect of the task we are focusing on is the behavior involving landing planes. Other researchers have investigated the behavior of subjects in moving planes within the queues (e.g., John and Lallement, 1997; Lee, Matessa, and Anderson, 1995; Lee and Anderson, 1997). We will focus on the choices made by subjects in assigning the different plane types to runways under varying weather conditions and proportions of plane types in the incoming flights.

Data Set

The KA-ATC data we modeled are reported in Reder and Schunn (in press) and Schunn and Reder (in press). Overall, subjects demonstrate great similarity despite the task complexity. As can be seen in Table 1, subjects land 747s on the long runway, the only legal runway for this plane type, almost exclusively. Mistakes with this plane type are rare, even in the first several trials when subjects are learning the task.

Table 1: For the adaptive subjects in Block 1, the percentage of all planes landed, and percentage of landings on the short and long runways for each plane type.

Plane type	% of all landed	% on short runway	% on long runway
Prop	30	75	25
DC-10	40	20	80
727	5	30	70
747	25	0	100

Subjects also consistently land props on the short runway and infrequently on long runway even though either runway is legal and landing long requires fewer keystrokes. One possible explanation is that subjects view the long runway as a scarce resource and choose this strategy to conserve it. However, subjects associate props with the short runway before they have had a chance to determine the relative scarcity of the long runway in this task. A second possibility is that props are the only plane type that can always land short so this is a cognitively simple rule. A third possibility is participants use their real world knowledge of planes: large planes such as jumbo jets (747s) need long runways while small planes (props) belong on short runways.

The choices made with DC-10s are of primary interest here. DC-10s may land on both the short and long runway, although they may only land on the short runway under certain wind and ground conditions. Subjects base their runway choices for DC-10s not only the wind and ground conditions, but also on the proportions of plane types in the mix of incoming planes. Subjects land the DC-10s on the short runway more often when there is a mix of planes that make the long runway scarce. Specifically, Reder and Schunn (in press) used runway preference for DC-10s to measure adaptivity in subjects. They varied the proportions of Props to 747s by block while maintaining a high but constant proportion of DC-10s (40%) and a low constant proportion of 727s (5%). The lower proportion of 747s in block 2 (5%) vs. the proportion in block 1 (25%) eases the demand for the long runway, while the highest proportion of 747s in block 3 (50%) creates the greatest demand for the long runway.

Reder and Schunn labeled a landing 'OpShort' when a subject chose to land a DC-10 on the short runway and both runways were open. Hits were defined as attempting to land a DC-10 on the short runway when legal and misses were defined as attempting to land a DC-10 on the long runway when it was legal to land on the short runway. False alarms were defined as attempting to land on the short runway when the wind and ground conditions made such a landing illegal while correct rejections were defined as attempting to land on the long runway when it was illegal to land on the short runway.

Although the manipulation was quite heavy-handed, some subjects did not shift their landing patterns to take advantage of the changing mix of planes. Reder and Schunn labeled these subjects as nonadaptive. A second group of subjects, on which we focus here—the *adaptive* subjects—differentially allocated DC-10s to the long and short runway in response to the demands caused by high proportions of Props or 747s. Adaptive subjects showed a pattern of short runway usage for DC-10s that decreased from blocks 1 to 2, when more Props and fewer 747s were included in the mix, and increased beyond the initial level in block 1 when, in block 3, the proportion of 747s to Props was significantly increased.

The data in Table 2 report the proportion of hits to hits plus misses and the proportion of false alarms to false alarms plus correct rejections for the adaptive subjects. In the first block, hits and false alarms are not significantly different ($p > .1$), indicating no sensitivity to the rules for landing DC-10s in different weather conditions. In the second block hits and false alarms also have similar magnitudes ($p > .5$), but are decreased relative to block 1. In other words the subjects decrease their usage of the short runway for DC-10s during this block, when a high number of props make the short runway a scarce resource. In the third block, when there are a large number of 747s, subjects increase their usage of the short runway for DC-10s relative to both the first and second blocks ($p < .01$). Also, hits increase more than false alarms in the third block, indicating that subjects become sensitive to the weather rules for landing DC-10s ($p<.01$).

Table 2: Hit and false alarm proportions.

Block	hits	false alarms
1	.31	.25
2	.19	.20
3	.66	.38

Model

The framework we chose for the current study is the ACT-R cognitive architecture (Anderson, 1993; Anderson and Lebiere, in press). ACT-R consists of a production system linked to a spreading activation network. These two components provide an architectural separation for procedural and declarative knowledge. Procedural knowledge takes the form of individual productions and the parameters associated with those productions. A goal stack controls the flow of control within the system and determines which productions may execute at any point. Declarative knowledge takes the form of a number of chunks, or node-link structures, contained within the declarative memory of ACT-R.

The ACT-R theory supports blame assignment through the goal stack mechanism. When an error state or success state is reached, that error is propagated back to productions that participated on the route to the error state. For example, if a DC-10 is successfully landed on the long runway, but the short runway is also open and the only available plane is a 747 (which can only land on the long runway), the production responsible for landing the DC-10 receives part of the blame for the error (failure to use an open runway). This feedback makes the goal structure a key part of this modeling project. A goal structure that includes too many prior actions will punish or reward productions that had little to do with the current success or failure. A goal structure that includes too few productions may not allow the system to assign blame far enough back in time to reach the causal action.

Model Description The current model is an ACT-R 4.0 model that interfaces with a LISP simulation of the ATC task. The aspects of the task simulated include the various hold levels, runways, plane types, mix of incoming planes in the queue, weather and ground conditions, rules for moving and landing planes, and the block structure and timing of the original task. Of the rules mentioned above, only rule 4 which requires that planes with low fuel be landed immediately, was not included in the simulation.

The schematic representation used by the ACT-R system includes a structure of chunks that represent the various elements of the game interface, and the goal structures used in the task. The goal types include goals to obtain information about the current game state, to land planes, and to move planes within the hold levels.

The productions used to simulate the behavior of subjects fall into two categories. The first set of productions gather or notice information in the environment such as which planes are in the first hold, what the current wind direction is, and whether the runways are open or busy. The second set of productions act on the gathered information and interact with the game simulation. At the highest level of abstraction, then, the system first examines the current game state, and subsequently chooses an action.

An example production from the model that attempts to land a DC-10 on the short runway is:

```
If the goal is to land a plane
  and there is a DC-10 in hold level 1
  and the short runway is open
then try to land the DC-10 on that runway
```

There are 6 productions that choose to attempt to land the various plane types on available runways. There are initially two enabled productions each for the DC-10 and 727 specific to the long and short runways respectively. There is one production for 747s, which must land long. There is also only one production for Props, which always land on the short runway.

The action chosen at each step of this simulation is constrained by two things: 1) whether a production applies to the current goal, and 2) the history of success and failure of the production. The conflict set includes those productions that match the type and values of the current goal. Each of these matching productions has an expected gain against the goal that is calculated by the formula $E = PG - C$. In this formula P is the probability of succeeding, G is the value of the current goal, and C is the cost of following the current production's path to the goal. Repeated success leads to an increase in P and therefore the chance that this production will be selected, while repeated failure leads to a decrease in P and a decrease in the chance that this production will be selected. Some amount of noise can be added to this calculation to provide variability of behavior. Variability allows sampling of previously unsuccessful strategies, which is necessary to confirm or refute earlier experience in a static environment, and to adapt and use new or more appropriate strategies in a dynamic environment (Lovett and Anderson, 1996; see also Holland, 1992).

Success and error feedback impacts the estimation of P, the probability parameter for each production. Because success and failure is the only source of adaptivity within the model we are exploring, this definition is critical to the performance of the model. Since the stated goal of the task is to accumulate points, and points are gained by landing planes while points are lost by violating rules, we used this guideline to define success and failure for the model. A successful trial is a sequence of actions which result in the legal landing of a plane or planes on the open runway(s). A failure is a sequence of actions which result in a rule violation or an inability to land a plane or plane(s) on the open runway(s). When neither runway is open, the system engages in other activities from which it does not receive feedback, such as checking the weather, moving planes within the hold levels, and accepting planes from the queue of incoming flights.

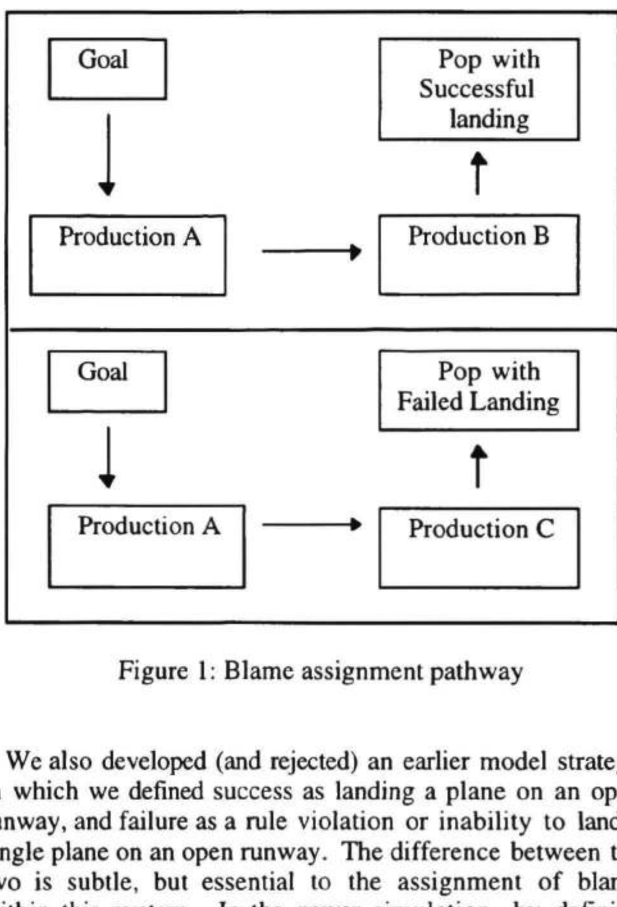

Figure 1: Blame assignment pathway

We also developed (and rejected) an earlier model strategy in which we defined success as landing a plane on an open runway, and failure as a rule violation or inability to land a single plane on an open runway. The difference between the two is subtle, but essential to the assignment of blame within this system. In the newer simulation, by defining success or failure in terms of landing pairs of planes, a landing on one runway that blocks other possible landings on the other runway receives partial blame for the failure. This means that even though a particular action might be legal and successful on its own (e.g., landing a DC-10 on the long runway), if this action creates a subsequent impasse (e.g., unable to land a 747 on the now occupied long runway), then it will become less likely. Without this improved definition of failure (and corresponding goal structure), the simulation cannot learn to avoid such problems. We discuss this point further in the next section.

Through learning success and failure probabilities associated with each production, the model is able to change its overall proportion of OpShort use over blocks. However, there remains the issue of separation of hits and false alarms in OpShort use over blocks (i.e., initially no differentiation in blocks 1 and 2, followed by a large differentiation in block 3).

There are several straightforward alternative explanations for this change over blocks. In the first two blocks, the task is relatively simple, and the subjects were not under heavy pressure to use the weather information, and they were busy learning other aspects of the task. However, in the third, more difficult block, in order to frequently land DC10s on the short runway, the subjects had a larger incentive to make use of weather information. This change in use may reflect either: 1) the creation of new productions which encode the weather; 2) raising the estimates of probability of success associate with productions which check the weather; 3) the creation of new productions which actually make use of the weather information in deciding to use the short runway; or 4) some combination of the above. Changes in both encoding and the introduction of new productions upon learning have both been used to explain adaptation within a production system framework (e.g., Siegler, 1976).

As a first pass, our ACT-R model simulates the gradual emergence of proper use of the rule for DC-10s by encoding the complete weather information (wind direction, wind speed, and ground conditions) differentially in the three blocks. In blocks one and two, the complete weather is encoded 10% of the time when the system checks the weather. The other 90% of the time the wind direction is encoded, but the wind speed and ground condition are not properly encoded. In the third block, the complete weather information is encoded 80% of the time. The model therefore assumes that the obstacle to proper use of the short runway with DC-10s is insufficient encoding.

An alternative mechanism within the ACT-R framework that would provide similar results is the production learning mechanism. If the model does not have a production that applies the rule for landing DC-10s, but has the opportunity to learn the rule and does so for most simulated subjects by early in the third block, parameter learning will produce a separation in probability of success between the new production and the existing production (and therefore hits and false alarms) in the third block.

Model Fit to Data The key aspects of data to be captured include the overall landing pattern, and more importantly the pattern of landing DC-10s when both runways are open. Qualitatively, the model should show a reduction in hits and misses from block one to block two and an increase in hits in block three with smaller increase in false alarms. This separation of hits and false alarms can only be modeled by sensitivity to ground conditions—otherwise hits and false alarms will go together.

Our model provides both a good qualitative and quantitative fit to the data as is shown in table 3 below:

Table 3: Model hits and false alarms proportions, with deviations from empirical data.

Block	hits	Δ	false alarms	Δ
1	.34	-.03	.22	.03
2	.25	-.06	.20	.00
3	.58	.08	.32	.06

From the table it is apparent that the success and error information provided to the system placed similar demands on the ACT-R model to those that subjects were responding to. The qualitative fit emerged from two things: the pressure of the planes in the mix, and the enabling of a production sensitive to the rule for landing DC-10s.

What the table does not say is that there were some formulations of success and error feedback for which there was no apparent easily discoverable ACT-R model that fit the data. We experimented with several representations of the problem and learned several things from these efforts.

First, the inclusion of some plausible strategies did not change the model behavior because the model abandoned those strategies due to low probability of success. For example, picking a plane before looking at which runways were open or picking an open runway to try to use with no knowledge of the planes present in the first hold level were both ineffective strategies, and were discarded by the model. If only these strategies were given to the model, it resulted in very low use of the short runway, well below levels reached by human subjects, especially in block 3.

Second, we found that the error feedback must reach far enough back in time to pass the blame to the productions causing the problem. Landing planes in pairs when both runways were open provided a sufficiently large window in time to provide effective error feedback. Landing one plane at a time gave error feedback only effecting the current landing that was too local in nature. As an example, landing a DC-10 on a long runway will never fail. If the production that performs this action only receives feedback from that landing, then the model will learn to always prefer landing DC-10s on the long runway. On the other hand, attempting to fill both runways when they are both open does allow for proper feedback. If landing a DC-10 on the long runway consistently blocks a 747s from landing, as happens during the early part of block 3, the model learns to avoid using the long runway for DC-10s.

Finally, change in the probability of weather encoding is necessary to allow for an increase in the proportion of hits over false alarms. For a difference between hits and false alarms to occur in any block, both human subjects and the model must encode the complete weather conditions and use those conditions to land DC-10s on the short runway when legal. Any model that accounts for this data must be able to land the DC-10s on the short runway taking the rules and weather into consideration.

Conclusions

This modeling effort has important implications for three different domains: the understanding of human cognition, ACT-R, and the KA-ATC task.

We have demonstrated that implicit learning of local success and failure can help account for the pattern of behavior that human subjects exhibit even in a relatively complicated task. Although it seems intuitive that the only way to solve a task and perform well is to deliberately plan the moves, this model demonstrates that the necessary feedback actually is present at a fairly local level. This possibility may have been overlooked in other studies concluding that performance resulted from detailed, goal oriented planning. In these cases, subjects may have been aware of their options without understanding that their basis for choosing an action was their history of success and failure (e.g., Lovett and Anderson, 1996). It is important to note that this adaptivity over blocks, in which the model decreased and then increased use of OpShort, did not depend upon the differential encoding over time.

Some structural change is necessary within the representation of the problem we chose for this model to capture the separation of hits and false alarms in block 3. As a first pass, we simulated this by a shift in encoding from partial weather information (wind direction) to complete weather information (wind direction, wind speed, ground conditions). The inability of subjects to take advantage of this information early on may be the result of several factors. It is possible that subjects simply cannot encode chunks to maintain the weather information early in the game. As they gain experience with the chunks, it becomes easier to both encode, retain and use the information. It is also possible that early on subjects are attending to other non-informative features of the game and do not pay attention to the complete weather information. Alternatively, the subjects may encode the complete weather information but may have difficulty remembering the rule in order to apply the information. They may need extensive practice before they can reliably recall the proper rule. It would not be surprising if the human data is best explained by some combination of these factors rather than any one.

We are currently exploring extensions to the current model that will let it gradually accumulate experience with the rule for DC-10s. We are also exploring accumulating experience with the interface as the driving force behind improving the ability to encode the information presented in the task. This interface learning should provide an account for the gradual shift in strategy from rarely encoding the weather information to usually encoding the weather. This type of strategy shift is another example of strategy adaptivity via implicit learning—through experience, the model learns the success and effort levels of trying to encode whether or not bothering to encode weather and uses this information to selection among those strategies.

Another interesting lesson learned from the model is that the strategies that have a higher percentage of success require the model to maintain more information in working memory. The model strategy that best simulated human performance required simultaneous access to the current weather conditions, runway status, and planes waiting to land.

Maintaining all of this information while choosing an option is demanding, and may explain why some human subjects were not able to adapt to the changing proportions of planes.

Finally, this model succeeded in the ACT-R framework not because parameters were exhaustively experimented with, but because a representation of success and failure was found that allowed the model to effectively assign blame. Some representations could not provide the necessary feedback for ACT-R to learn the task. This demonstrates that the probability parameters in ACT-R productions are not free parameters that can be used to fit curves. Instead, the production parameter learning mechanism puts realistic constraints on these parameter values, and therefore on the action of the ACT-R system.

Acknowledgments

This research was supported by ONR grant N00014-95-1-0223 to the third author. We thank John Anderson and Christian Lebiere for their comments and suggestions.

References

Ackerman, P. L., and Kanfer, R. (1994). Kanfer-Ackerman Air Traffic Controller Task© CD-ROM Database, Data Collection Program, and Playback Program. Office of Naval Research, Cognitive Science Program.

Anderson, J. A. (1993). *Rules of the Mind.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Anderson, J. A., and Lebiere, C. (in press). *The Atomic Components of Thought.* Mahwah, NJ: Lawrence Erlbaum Associates.

Lee, F. J., Anderson, J. R., & Matessa, M. P. (1995). Components of Dynamic Skill Acquisition. *Proceedings of the Seventeenth Annual Conference of the Cognitive Science Society* (pp. 506-511). Hillsdale, NJ: Lawrence Erlbaum Associates.

Holland, J. H. (1992). *Adaptation in Natural and Artificial Systems.* Cambridge, MA: MIT Press.

Holland, J. H., Holyoak, K. J., Nisbett, R. E., & Thagard, P. (1986). *Induction: Processes of inference, learning, and discovery.* Cambridge, MA: MIT Press.

John, B., & Lallement, Y. (1997). Strategy use while learning to perform the Kanfer-Ackerman Air Traffic Controller Task. In the *Proceedings of the 19th Annual Conference of the Cognitive Science Society.* Mahwah, NJ: Erlbaum.

Lovett , M. C., & Anderson, J. R. (1996). History of success and current context in problem solving: Combined influences on operator selection. *Cognitive Psychology, 31*, 168-217.

Newell, A., & Simon, H. A. (1972). *Human problem solving.* Englewood Cliffs, NJ: Prentice-Hall.

Reder, L. M. (1987). Strategy selection in question answering. *Cognitive Psychology, 19*, 90-137.

Reder, L. M., & Schunn, C. D. (1996). Metacognition does not imply awareness: Strategy choice is governed by implicit learning and memory. In L. M. Reder (Ed.), *Implicit memory and metacognition* (pp. 45-78). Mahwah, NJ: Erlbaum.

Reder, L. M., & Schunn, C. D. (in press). Bringing together the psychometric and strategy worlds: Predicting adaptivity in a dynamic task. To appear in Gopher, D. & Koriat, A. (Eds.), *Cognitive regulation of performance: Interaction of theory and application. Attention and Performance XVII.*

Schunn, C. D., & Reder, L. M. (In press). Individual differences in strategy adaptivity. To appear in D. L. Medin (Ed.), *The psychology of learning and motivation.*

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

Siegler, R. S. (1976). Three aspects of cognitive development. *Cognitive Psychology, 8*, 481-520.