An Integrated Model of Set Shifting and Maintenance

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

Abstract

Serial attention involves adopting a task set (or intention to do a task), maintaining that set for a time, then shifting set. When set must be shifted often, serial attention is difficult because old sets cannot be forgotten instantly and will interfere with the current one. This memory perspective on set shifting is the basis for a computational cognitive model that explains a variety of effects in serial attention, some well-documented (e.g., RT switch cost) and some novel (e.g., within-run slowing). The mechanisms producing these effects are tightly integrated with each other and with memory theory. The model absorbs extant accounts of set shifting, including failure to prepare (De Jong, Berendsen, & Cools, 1999) and exogenous cueing (Rogers & Monsell, 1995), and promises to unify the effects presented here with other set-shifting phenomena.

Introduction

Serial attention is the process of serially attending to one task after another in some sequence appropriate to the environment. Serial attention is an iterative process of adopting a task set (an intention to act on a stimulus in a particular way), maintaining that task set for a time (say over multiple trials), then shifting to a new task set. In circumstances of sustained performance, the difficulty of serial attention lies in updating one's memory for the current set, such that old, incorrect sets are masked. Cast as a memory problem, serial attention is difficult because of interference from old task sets.

We present a model of serial attention grounded on the assumption that decay is important for preventing interference. The model is comprehensive in that it explains five effects we find in our data, and is integrated in that every explanatory mechanism has some other reason to exist other than effect it explains. That is, each mechanism is grounded in an independent constraint – an existing theoretical premise, a functional relationship to another mechanism in the model, or an incidental explanation of another effect (in addition to the effect that the mechanism directly explains). Extant models of set shifting generally fail on this criterion of independent constraint. The typical model focuses on one or two effects and not an entire family, and its mechanisms are verbal rather than computational.

We first introduce a serial attention paradigm that generates a diverse set of phenomena. Several of these are well known, for example the response time (RT) cost incurred immediately after switching tasks (Allport, Styles, & Hsieh, 1994; De Jong et al., 1999; Gopher, Greenshpan, & Armony, 1996; Mayr & Keele, in press; Rogers & Monsell, 1995). However, at least one effect is entirely novel. After describing these effects, we introduce a model that integrates them into a coherent whole.[1] We conclude that serial attention phenomena are due to cognition compensating for its limited capacity to forget, and argue that our model is on the path toward a unified theory of serial attention.

---

[1] ACT-R code for the model is available via Internet at http://hfac.gmu.edu/people/altmann/sa.txt.

The Serial Attention Paradigm

Our serial attention paradigm involves presenting participants with several thousand <u>choice</u> trials. On each trial the participant categorizes a stimulus like "aaaaaaa". When one trial ends, the next trial starts immediately. Trials are presented in <u>runs</u> of 7 to 13, with the number of trials per run determined randomly. A run begins with an <u>instruction trial</u>, which is terminated by pressing the space bar. The instruction indicates the task, or which categories to choose from on that run. The <u>Groupsize</u> instruction says to choose whether the number of characters in the choice-trial stimulus is greater or fewer than five. The <u>Whichend</u> instruction says to choose whether the repeated character is near the start or the end of the alphabet. The character is one of a, b, c, d, w, x, y, or z, and is repeated 1, 2, 3, 4, 6, 7, 8, or 9 times.

Instruction trials are chosen randomly, constrained such that half give <u>Switch</u> instructions and half give <u>Noswitch</u> instructions. A Switch instruction indicates the task other than the previous one, and a Noswitch instruction indicates the same task again. The instruction must be retained in memory, because there is no cue on choice trials. A session contains 384 instruction trials, and hence 3840 choice trials (an instruction is followed by 10 choice trials, on average). A choice trial takes about half a second, so the participant must update his or her task set every five seconds or so for roughly an hour. On the premise that decay is necessary but not instantaneous, this memory updating is the primary challenge facing the cognitive system.

Sample Data

Figure 1 shows empirical RTs for the instruction trial (P0) and the first seven choice trials afterwards[2]. The data show five effects[3]. The first is the duration of P0, which takes roughly twice as long (~1000 msec) as the average choice trial (~500 msec). This difference is the opposite of that predicted by a simple task analysis. Instruction trials do not require response selection, whereas choice trials do. Moreover, instruction trials do not rely on information retained in memory, but choice trials do depend on memory and hence incur the cost of memory retrieval. Despite extra stages on choice trials, they are faster than instruction trials.

The second effect is the difference in P0 RT across Switch and Noswitch conditions, which is one measure of switch cost in our data. The size of this switch cost is 96 msec. The third effect is the relative slowness of P1 (the first choice trial) compared to subsequent choice trials. The difference between P1 and P2 is 153 msec. The fourth effect is the difference in P1 RT across Switch and Noswitch runs, which is a second measure of switch cost in our data. The size of this switch cost is 60 msec. This and the switch cost mentioned above replicate previous findings in a similar paradigm (Gopher et al., 1996). The fifth effect is the slowing trend starting with P2 and continuing through P7. This effect is small (about 5 msec per trial) but monotonic. To our knowledge, this slowing trend has not been reported previously, though such trends have appeared in other data sets (Rogers & Monsell, 1995, Exp. 6).

These five effects are a diverse assortment, and pose a challenge to any model that would explain them coherently. Below we describe one such model.

---

[2] Analysis is limited to the first seven trials in a run because every run has at least that many; starting with the eighth trial the number of observations on each trial begins to decline monotonically.

[3] All five effects are significant at the 0.05 level. The data reported here are a superset of the data from Experiment 1 Letters in Altmann and Gray (1999a), which does not report data for P0 or P1.

Overview of the Model

Our theoretical premise is that masking or inhibiting an old item in memory cannot be done instantaneously, but takes time and active processing. The notion that it takes time to forget a trace is grounded in decay theory. The specific formulation of decay theory we adopt is the base-level learning mechanism of the ACT-R cognitive architecture (Anderson & Lebiere, 1998). According to this mechanism, activation is a function of number of retrievals over time. Also, whether activation is increasing or decreasing at any given moment is governed by the spacing of retrievals in the history of the item.

The notion that it takes active processing as well as decay to forget an item is our approach to specifying the processes at work in inhibition-based accounts of set shifting (Allport et al., 1994; Mayr & Keele, in press). Empirical evidence suggests that inhibition is an active, resource-bound process (De Jong et al., 1999; Engle, Conway, Tuholski, & Shisler, 1995), and in functional terms one item is inhibited relative to another if it is the less active of the two (Lovett, Reder, & Lebiere, 1999). This functional interpretation suggests that one item can be inhibited by increasing the activation of a new item, such that the new one is the more active of the two. In the context of serial attention, this activation-difference account of inhibition depends critically on our first assumption that items also decay on their own. Decay is necessary to avoid an infinite regress in which each new instruction has to be made more active initially than its predecessor was initially. The regress is avoided if an instruction decays from when it was first encoded in memory. This decay ensures that when a new instruction is encoded it will be more active, assuming that it is encoded to the same level of activation as its predecessor (Altmann & Gray, 1999a).
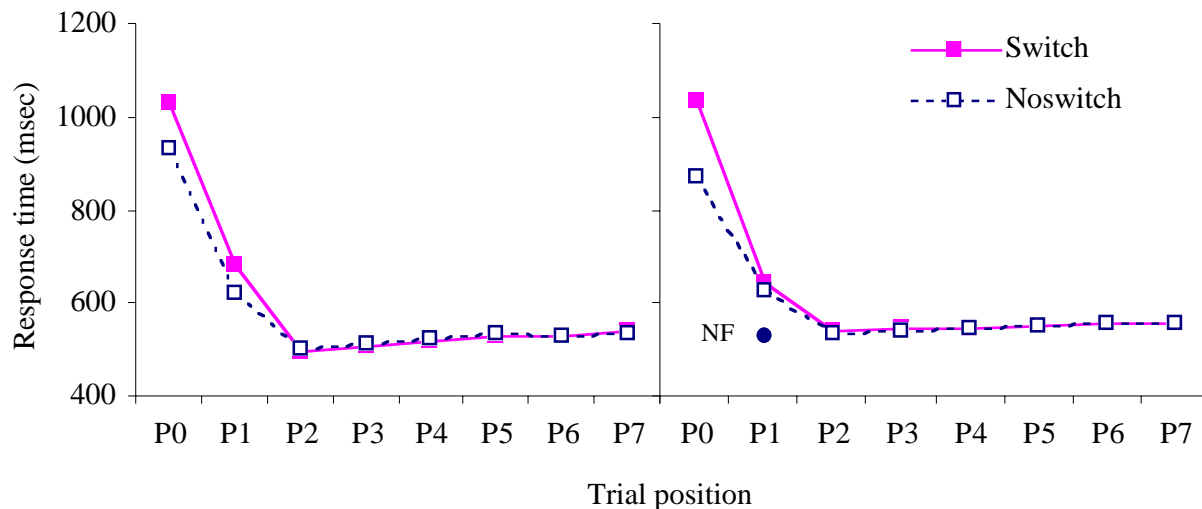

Figure 1: Serial attention response times (RTs). P0 is the instruction trial, and P1 to P7 are the subsequent choice trials. Summary of effects: (1) P0 is slow relative to choice trials; (2) P0 is slower when switching tasks than not; (3) P1 is slow compared to subsequent choice trials; (4) P1 is slower after a switch than not; and (5) RT slows gradually from P2 to P7. The point labeled NF is from the simulated Nofollow condition, in which there is no encoding on P1.

The two main processes in the model are introduced here and related to the data in below. The <u>encoding</u> process encodes an instruction in memory when the instruction is visually available, making it active enough to mask all previous instruction traces. This process is time consuming, and in our model is the locus of RT switch costs. The <u>maintenance</u> process maintains task set between instruction trials. The procedure for maintaining task set is simply to retrieve the most recent instruction trace from memory. Because the most recent trace is also the most active (as a function of encoding), the maintenance process simply takes the first trace returned from memory. This procedure fails only when noise in the memory system accidentally makes an old instruction trace more active than the current one during a given retrieval attempt. To ensure that the instruction trace decays before the next instruction is presented, the set-maintenance mechanism retrieves an instruction only once per trial. We assume that a single retrieval is the minimum needed for performance to be reliably guided by a task set on a given trial. This retrieval rate is insufficient to sustain the high initial activation produced by encoding, and the instruction trace decays gradually.

In sum, we assume that active processing and decay are both necessary to update memory for the current instruction. From this assumption flows a full account of the diverse effects in Figure 1, as we describe below.

Comparing Data and Model

Below we describe how our model accounts for the data. The five empirical effects introduced above are organized into three categories derived from the structure of the model: encoding time, switch costs, and within-run slowing. Simulation data from our model appear in the right panel of Figure 1.

Encoding Time

The role of the encoding process is to activate the current instruction so as to mask the previous one. Two constraints make it possible to use this process to make quantitative predictions. First, as we touched on earlier, the process must generate enough activation that functional decay ensues (Altmann & Gray, 1999a). Functional decay occurs when an instruction trace decays because its retrievals are relatively spaced and cannot sustain the trace's high initial activation. The second constraint specifies how much activation the encoding process can produce per unit time. Encoding in ACT-R occurs in discrete <u>cycles</u> of 100 msec duration (Altmann & Gray, 1999b). ACT-R's base-level learning equation lets us predict that for an instruction trace to decay over N retrievals, initial encoding must run for N cycles (Altmann & Gray, 1999a). Thus, at 100 msec per encoding cycle, and for an average run length of 10 choice trials, encoding time should be roughly one second.

This prediction of roughly a second for encoding explains why P0 is so much slower than the average choice trial. Far from requiring less processing than choice trials, as our simple task analysis suggested earlier, the instruction trial involves a substantial investment to activate the current instruction to the level needed for accurate performance.

The other effect we attribute to the encoding process is the slowness of P1 compared to later choice trials. In the model, this effect is due to <u>follow-up encoding</u>, a second phase of the encoding process that follows the first phase that ran on P0. Follow-up encoding improves performance by reducing the effects of activation noise on P0 encoding. Following ACT theory, we assume that memory is subject to noise, and that this noise is represented as

transient fluctuations in activation levels. In our model, this noise can affect the model's encoding process by causing it to terminate prematurely. If encoding stops before the instruction has reached the target activation level, it will remain weaker through its lifetime. Running the encoding process again causes the instruction trace to regress to its expected activation level, because the trace gets a second chance to be strengthened.

The reason that the encoding process can run again on P1 is that it has access to a veridical representation of the instruction for a brief period after P0 offset. We assume that when the instruction is visually available, a veridical, post-categorical representation of the instruction exists in the cognitive system. Based on the parameters of visual short-term memory (Phillips, 1974) and conceptual short-term memory (Potter, 1993), we estimate that this representation persists briefly after the offset of P0 but not beyond the offset of P1 (roughly 650 msec later). At P2 onset, therefore, this veridical representation is no longer available. Thus P1 is the only opportunity to run the encoding process after P0.

Further evidence that a functional process causes P1 slowing comes from the error data. Whereas RT on P1 is significantly higher than on later trials, the trend is for errors to be fewer on P1 than on other choice trials (Altmann & Gray, 1999a). This is strong evidence against interpreting P1 slowing as a general performance deficit of some kind. One might interpret P1 slowing as a kind of recovery from being interrupted by an instruction in the midst of performing choice trials. However, if P1 slowing reflected disrupted performance, one would expect it to be accompanied by a decrease in accuracy. Instead, there is no hint of error increase on P1, suggesting that RT increase is due to a separate and functional process rather than a generalized performance deficit.

To prove the benefits of follow-up encoding, we ran Monte Carlo simulations of our model in two conditions. In the Followup condition, follow-up encoding on P1 was enabled, whereas in the Nofollow condition follow-up encoding was disabled. The Followup condition produced the simulated RTs in Figure 1. The Nofollow condition produced identical simulated RTs except on P1, which was now faster than the other choice trials, rather than slower. In Figure 1, Nofollow P1 RT is the dot labeled NF in the right panel. For each condition we measured the error rate and average time per choice trial on each of 10 simulated experimental sessions. The error rate in the Followup condition ($\underline{M}$= 0.042) was significantly less than error rate in the Nofollow condition ($\underline{M}$ = 0.053), $\underline{t}(9) = 4.8$, $\underline{p} < .001$. Moreover, average time per trial was no slower in the Followup condition ($\underline{M}$ = 564 msec), than in the Nofollow condition ($\underline{M}$ = 567 msec). This may seem unusual, because the extra processing due to follow-up encoding should increase average time per trial by increasing RT on P1. However, the function of this extra processing is to improve the strength of instruction traces that were created weak. Because retrieval latency, like retrieval error, is a decreasing function of activation, the model retrieves the instruction faster on subsequent trials if that instruction's strength was increased by follow-up encoding. Thus follow-up encoding reduces error at no extra cost in terms of overall performance time. This demonstration of improved accuracy with no overall performance cost supports the interpretation that P1 slowing reflects an adaptation of cognition to its own limitations, in this case noise in memory.

In sum, we the slowness of P0 and P1 to an encoding process that activates the current instruction. The distribution of encoding over two trials improves performance accuracy by reducing noise in activation levels. In the discussion we suggest that this encoding process subsumes at least two existing accounts of set shifting.

Switch costs

Our model offers an integrated explanation of the two switch costs in our data. The RT switch costs on P0 and P1 have functional significance in that prolonged encoding of Switch instructions helps to compensate for increased likelihood of interference from an incorrect instruction trace. Modulo noise, the most active instruction in memory is the current one, the second most active is from the previous run, and so on. Thus the previous instruction is the most likely to intrude on the current instruction. On a Switch run, the previous instruction is incorrect, whereas on a Noswitch run the previous instruction is correct. Thus on Switch runs, the likelihood of a memory error is systematically higher. Prolonged encoding of Switch instructions helps to compensate for this imbalance.

The trigger for prolonged encoding of Switch instructions is the interruption of cognitive repetition benefit from the previous run. Interruption of repetition benefit appears to be a distinct source of switch cost (Sohn & Anderson, 1999), and one needs to assume a pervasive repetition benefit to account for high-accuracy performance (Altmann & Gray, 1999b). In our model, repetition benefit from the previous run automatically causes Noswitch instructions to be encoded with less activation than Switch instructions[4]. This weaker encoding of Noswitch instructions, or more specifically the relatively stronger encoding of Switch instructions, helps offset the increased chance of error on Switch runs, thereby helping to even out performance across the two conditions.

In sum, the model explains RT switch costs in functional terms. Error is higher on Switch runs, because the most likely instruction to intrude is incorrect (as a function of the switch). However, this cost is mitigated by extra encoding of Switch instructions, with this compensatory processing triggered automatically by the absence of repetition priming.

Within-run slowing

Within-run slowing is the steady increase in RT from P2 to P7. In the model, the cause of this effect is decay of memory for the current instruction, an outcome of fine-grain spacing effects in retrieval of the instruction trace. During encoding, the trace is subjected to concentrated "use" that boosts its activation level. This concentrated use is possible because there is essentially nothing else to do on the instruction trial. In contrast, retrievals on choice trials are relatively spaced, because choice trials require other processing. A spacing effect arises because massed use during encoding produces an initial activation level that is too high to be sustained by the relatively spaced use during the subsequent run. Thus, the instruction decays during the run, despite that fact that it is being regularly retrieved. As the instruction decays, it becomes less available in memory, increasing both retrieval time and error.

In sum, we explain within-run slowing, a novel effect in serial attention, as a side effect of processes that maintain activation superiority of the current instruction.

---

[4] On Noswitch runs, in which the previous task is the same as the current task, repetition benefit from the previous run transiently boosts the activation of the current instruction as it is being encoded. This boost causes the encoding process to exit earlier. However, in ACT the priming effects responsible for repetition benefit are transient rather than permanent. Thus the instruction trace is left weaker on Noswitch runs, precisely because the encoding process exits early.

Discussion

We have described five phenomena in set shifting and maintenance. Moreover, we have integrated these effects into one unified performance model in which every explanatory mechanism is independently constrained. The fundamental constraint is that decay is needed to prevent interference in memory for the current task. This need for decay implies the need for high initial levels of activation, explaining encoding time effects. The dynamics of interference explain switch costs, in that RT switch "cost" reflects processing that mitigates switch cost measured in terms of errors. Finally, within-run performance decline is simply a side effect of functional decay.

One of the benefits of expressing cognitive theory in terms of process models is that it can predict new effects (Newell, 1973). Our model predicts within-run slowing, which seems to lie outside the scope of other set-shifting models. Typically, explanations of task switching focus on the temporal locale of the task switch (Allport et al., 1994; De Jong et al., 1999; Gopher et al., 1996; Mayr & Keele, in press; Rogers & Monsell, 1995) and ignore the time course of performance between task switches.

One of the benefits of integrating multiple phenomena into one model is that this can point to opportunities for theoretical reduction (Newell, 1973). For example, a robust set-shifting effect is residual switch cost, which is the switch cost that remains when participants have foreknowledge and time to prepare for a task switch. It appears to us that residual switch cost is simply a special case of P1 slowing, with preparation time mapping to P0 (the instruction trial). This reduction would explain why residual switch cost is so robust: Because it reflects a process that is functional in improving overall performance (follow-up encoding), there is no reason for the cognitive system to abandon it.

Residual switch cost is nicely explained by the failure-to-engage (FTE) model (De Jong et al., 1999), which proposes that from time to time participants simply fail to take advantage of the foreknowledge and time available for preparation. When such a failure occurs, processing usually conducted ahead of time now prolongs the first trial governed by the new task set. Put another way, the FTE account proposes occasional lapses in attention that are made up only when a stimulus actually arrives. Cast in these terms, the FTE account appears to be subsumed by the encoding process in our model, in which encoding on P1 compensates for attentional lapses on P0. Whereas the stochastic attentional lapses of FTE stand alone, such lapses in two-phase encoding emerge directly from a quantitative, executable, and broadly-tested memory theory (Anderson & Lebiere, 1998).

The exogenous cueing account of residual switch cost (Rogers & Monsell, 1995) is also subsumed by our model. Exogenous cueing proposes that cognition delays shift completion to the first trial under the new task set. This explanation is ad hoc (as the authors note) and fails to specify any functional role for the delay. In contrast, our model suggests that residual switch cost (interpreted as a special case of P1 slowing) is not a cost at all, but reflects a process that evens out activation levels in memory and thereby improves accuracy. It remains to be seen whether our encoding process, or perhaps other processes in our model, extend naturally to the evidence that inhibitory processes are also at play in serial attention (Mayr & Keele, in press).

In conclusion, we have introduced an integrated model that explains an entire family of effects in serial attention and appears to absorb other models of task switching. We have also presented a new effect, within-run slowing, that other models will have to address. In future research, we hope to extend our model to account for residual effects in set-shifting and maintenance and thereby build toward a unified theory of serial attention.

Acknowledgements

References

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

Altmann, E. M., & Gray, W. D. (1999a). Functional decay in serial attention. Manuscript submitted for publication.

Altmann, E. M., & Gray, W. D. (1999b). Serial attention as strategic memory, Proceedings of the twenty first annual meeting of the Cognitive Science Society (pp. 25-30). Hillsdale, NJ: Erlbaum.

Anderson, J. R., & Lebiere, C. (Eds.). (1998). The atomic components of thought. Hillsdale, NJ: Erlbaum.

De Jong, R., Berendsen, E., & Cools, R. (1999). Goal neglect and inhibitory limitations: Dissociable causes of interference effects in conflict situations. Acta Psychologica, 101, 379-394.

Engle, R. W., Conway, A. R. A., Tuholski, S. W., & Shisler, R. J. (1995). A resource account of inhibition. Psychological Science, 6(2), 122-125.

Gopher, D., Greenshpan, Y., & Armony, L. (1996). Switching attention between tasks: Exploration of the components of executive control and their development with training, Proc. HFES 40th annual meeting (pp. 1060-1064). Philadelphia: HFES.

Lovett, M. C., Reder, L. M., & Lebiere, C. (1999). Modeling working memory in a unified architecture: An ACT-R perspective. In A. Miyake & P. Shah (Eds.), Models of working memory: Mechanisms of active maintenance and executive control (pp. 135-182). New York: Cambridge University Press.

Mayr, U., & Keele, S. W. (in press). Changing internal constraints on action: The role of backward inhibition. Journal of Experimental Psychology: General.

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

Phillips, W. A. (1974). On the distinction between sensory storage and short-term visual memory. Perception & Psychophysics, 16(2), 283-290.

Potter, M. C. (1993). Very short term conceptual memory. Memory and Cognition, 21, 156-161.

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

Sohn, M.-H., & Anderson, J. R. (1999). ACT-R does task switching: Effects of foreknowledge and foreperiod on task-switch cost. Paper presented at the Psychonomic Society 40th annual meeting, Los Angeles.