A Long-Term Memory Competitive Process Model of a Common Procedural Error

Franklin P. Tamborello, II (franklin.tamborello.ctr@nrl.navy.mil)
National Research Council Postdoctoral Research Associate
J. Gregory Trafton (trafton@itd.nrl.navy.mil)
United States Naval Research Laboratory
4555 Overlook Ave SW
Washington, DC 20032 USA

Abstract

A novel computational cognitive model explains human procedural error in terms of declarative memory processes. This is an early version of a process model intended to predict and explain multiple classes of procedural error a priori. We begin with postcompletion error (PCE), a type of systematic procedural error that people are prone to commit when there is one step to perform after they have accomplished their main task goal. Participants in an experiment demonstrated increased PCE rates following an interruption in a realistic form-filling task. The model explains PCE as a consequence of two declarative retrieval processes, spreading activation and base-level activation, competing with each other because of features of task and working memory structure. Our intention is to generalize the model to other classes of procedural error in complex task environments.

Keywords: computational cognitive model; human error; human-computer interaction; interruption; long-term memory; working memory

Introduction

If you have ever left an original document on a photocopier after walking away with the copies then you have committed a postcompletion error (PCE). PCE is one example of a systematic procedural error, an error people tend to commit in familiar tasks that follow a specific sequence of actions each time the task is performed. Systematic procedural errors seem to be products of a combination of stable human cognitive structures and processes as well as certain task environments. PCE, in particular, tends to have a much higher rate of incidence than chance slips and seems to be very resistant to training (Byrne & Davis, 2006). Our goal is to understand the cognitive structures and processes underlying PCE and, ultimately, to extend that same model to account for other systematic procedural error types.

Studying human error is important because with increasing capability and complexity of our technological systems (e.g., transportation, power generation) the amount of damage that can result from error is magnified. While chance slips occur because humans are fundamentally stochastic, systematic error occurs when certain features of human cognition meet certain task environmental conditions. If we learn about those cognitive and environmental features then we can learn to avoid them in our technological systems such as by exclusion from designs (Chung & Byrne, 2008) or prediction and prevention (Ratwani & Trafton, 2011).

Studying human error is difficult because of the variability of error behavior. Furthermore, error often arises from the dynamic interactions of several cognitive processes that normally perform with with very little error. Models of human error are often complex compared to models of other behavior because these models must capture these interactions in ways that lead to proper proportions of both correct and incorrect behaviors.

For PCE, Byrne and Bovair (1997) explained it as a function of limited-capacity working memory. They addressed high and low working memory demand as well as individuals' high and low working memory capacities. Their model assumed a hierarchical goal representational structure. This was based on a GOMS (Card, Moran, & Newell, 1983) analysis of an experiment task also reported in their study. Their CAPS model (Just and Carpenter, 1992) propagated activation necessary for retrieval of step representations downward from the task supergoal to subgoals to individual steps. Subgoals had to have their activations maintained above a certain threshold in order for them to remain accessible. Crucially, the main goal of the procedure would be satisfied before it was time to perform the postcompletion step. The presence of other information to maintain in an active state, in this case a three-back memory task, taxed the system to capacity such that it failed to maintain the postcompletion subgoal above threshold.

Another account of systematic error, *Memory for Goals* (Altmann & Trafton, 2002), posits that we encode episodic traces of our goals as we complete tasks. Each goal is encapsulated in an episodic memory, which sparsely represents a behavioral context at the time of its encoding. The strength of these memories decay over time such that it may be difficult to remember the correct point at which we resume a task after an interruption. Memory for Goals provides a process-level theory for why certain types of errors are made during a well-learned task as a consequence of retrospective, episodic memory (Altmann & Trafton, 2007; Ratwani & Trafton, 2010, 2011; Trafton, Altmann, & Ratwani, 2009). Memory for Goals implies that people are able to retrieve suspended goals successfully if and only if there are cues that prime them (Altmann & Trafton, 2002). Here decay is indexed by time, so postcompletion steps, being at the end of their tasks, have relatively more time to decay compared to other steps that come earlier in the task.

The model presented in this paper draws upon both previous works, predicting PCE to occur as a combination of goal decay and a limited-capacity to spread activation from working memory to long term memory. Ultimately what we want is a unified framework with which we can make predictions about PCE, and later, other types of human error. A unified framework is important because one cognitive system, i.e. the human mind, produces all error types. Getting the explanation correct for one type then acts as a constraint on getting the explanation correct for the next

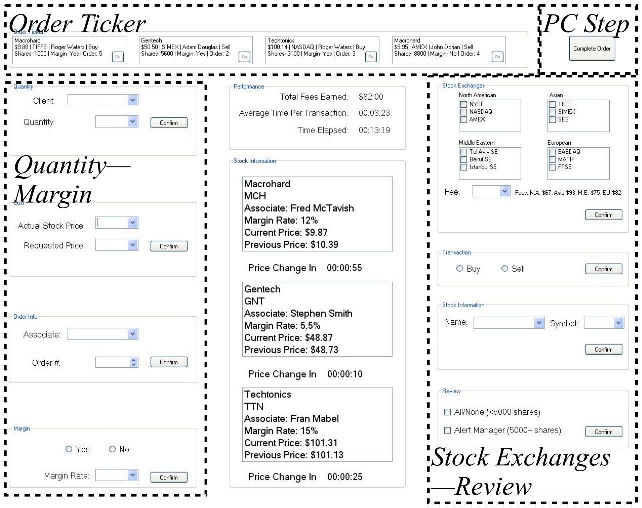

Figure 1: The financial management task interface resembled a web form. Subgoals assumed for the model are grouped by dotted lines and labeled for purposes of illustration here, but were not so in the task.

type tackled by the theory. Furthermore, if we are to predict error in complex task environments multiple error types must fall naturally out of the theory.

Experiment

Participants performed a version of Ratwani and Trafton's (2011) financial management task (Figure 1). This is a type of form-filling task wherein participants, using a graphical user interface, click a series of buttons in a specific order. The goal of the task is to fill out an order form according to information available within the display. An arithmetic task occasionally interrupted the financial management task for 15 seconds at a time.

The final step of the task consisted of a single button not placed within a box and placed above the right column of boxes. This arrangement broke with the Western reading convention followed by the progression of all of the other steps. This step was arranged this way because we intended it to serve as a postcompletion step.

Design and Procedure

Each order on the financial management task constituted a single trial. Control and interruption trials were manipulated in a within participants design; participants performed 12 trials. Half of the trials were control trials with no interruption and half were interruption trials with two interruptions each. The order of trials was randomly generated and participants did not have prior knowledge as to which trials would be control or interruption trials.

There were eight possible interruption points in the financial management task. These points occurred after clicking the Confirm button following the first seven modules, including just prior to the postcompletion action. The location of the interruptions on a trial by trial basis was randomized with the constraint that exactly two interruptions occurred just prior to the postcompletion step and at least one interruption occurred at each of the other seven possible locations. The were 12 postcompletion error opportunities, one during each trial. Six of these opportunities were during control trials with no interruptions, two opportunities were immediately following an interruption, and four opportunities were during interruption trials where an interruption occurred at a point that did not immediately precede the postcompletion step.

Participants were seated approximately 47cm from the computer monitor. After the experimenter explained the financial management task and interrupting task to the participant, the participant completed two training trials (one trial with and one trial without interruptions) with the experimenter. Following these two training trials, participants had to perform two consecutive randomly selected trials on their own without making a postcompletion error before the participant could begin the experiment. Forcing participants to perform two consecutive error free trials was a method for ensuring that participants were proficient at the task before beginning the actual experiment. Each participant was instructed to work at his/her own pace. When performing the interrupting task, participants were instructed to answer the addition problems as soon as the solution was known and to answer as many addition problems as possible in the time interval. Upon resumption of the financial management task, there was no information available on the interface to indicate where to resume.

For modeling purposes the important points about the financial management task were:

1. It featured a primary task that was occasionally interrupted by a secondary task,
2. Participants had to follow a specific procedure.
3. The spatial layout of the interface (working from top to bottom down the left column and then the right column of Figure 1) and the operations required to perform the task were quite intuitive.
4. After entering information in each module, the participant clicked the Complete Order button (upper right corner). Clicking the Complete Order button was the postcompletion step and failing to click the Complete Order button constituted a PCE.
5. The spatial layout of the task grouped steps by proximity. This encouraged use of an intuitive heuristic ("go down the column"), as well as having an isolated "clean-up" step at the end. This format followed the form of other tasks shown by GOMS analysis to lead to subgoaling (e.g., Byrne & Bovair, 1997).
6. No information remained on the interface after clicking the confirm button within each module (i.e. no global place keeping (Gray, 2002)).
7. Measures: A PCE was defined as failing to click the last step's button and instead making an action that was in service of the next order on the financial management task (e.g. attempting to start a new trial by clicking an Order Ticker). The PCE rate was the number of PCEs divided by the number of opportunities to make a PCE. Skipping the next correct step, at any other time, was classified as an anticipation error.

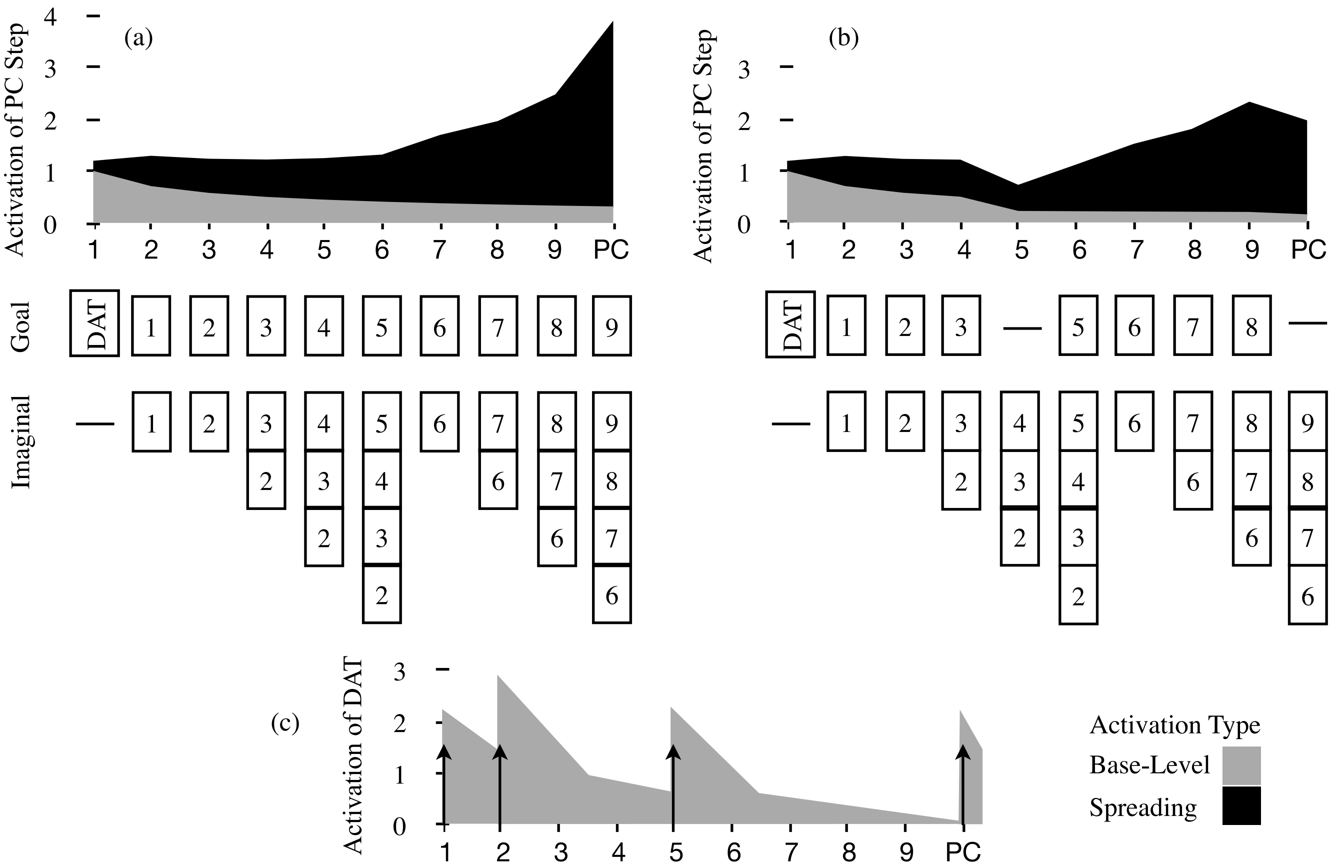

Figure 2: Base-level and spreading activation of the model's postcompletion step chunk in control (a) and interruption (b) trials. X-axes indicate the step to be performed, by step ordinal number. Interruptions occur for the interruption trial type between steps four and five and nine and ten in this example. Consequently the goal buffer chunk lacks task context representation in these two spots, indicated by dashes. The chunk encoding the main goal of the task is *Do a Trial* (DAT) and it is associated to subsequently performing the first step. *Do a Trial,* as well as each step's representation, acts as context to cue retrieval of the next step. *Do a Trial's* activation is depicted in panel c. Arrows indicate times at which *Do a Trial* receives activation boosts because the model retrieves it at the end of each subgoal. The model's behavior with regard to *Do a Trial* is the same in both control and interruption conditions. The model's internal context representations, encoded in the chunks referenced from the slots of the goal and imaginal buffer chunks, are depicted at each step of the task beneath panels a and b. These are the procedure step representations the model had retrieved when performing previous steps. To conserve space task steps are indicated by procedure sequence order, so the first step of the task is 1 and so on.

Model

A model constructed using the ACT-R 6 cognitive architecture (Anderson et al., 2004) performed an abstract version of the financial management task. ACT-R is a hybrid symbolic and subsymbolic computational cognitive architecture that takes as inputs knowledge (both procedural and declarative about how to do the task of interest) and a simulated environment in which to run. It posits several modules, each of which perform some aspect of cognition (e.g., long-term declarative memory, vision). Each module has a buffer into which it can place a symbolic representation that is made available to the other modules. ACT-R contains a variety of computational mechanisms and the ultimate output of the model is a time stamped series of behaviors including individual attention shifts, speech output, button presses, and the like. One of the benefits of embodying a theory in a computational architecture, such as ACT-R, is that it allows researchers to develop and test concrete, quantitative hypotheses and it forces the theorist to make virtually all assumptions explicit. To the extent that the model is able to simulate human-like performance the model provides a sufficiency proof of the theory.

In essence, the model worked by cyclic, activation-based retrieval from long-term memory of the task step representations encoded as *chunks*. At each step there were two sources of retrieval activation: 1) spreading activation from the contents of the goal and imaginal representations (these constituted the model's working memory), and 2) each chunk's base-level activation. Sometimes these activation sources conflicted with each other, particularly for the postcompletion step. At such times the model was likely to commit an error.

Activation spreading from the model's working memory to the long-term memory encoding the postcompletion step increases with advancing task context because of the inverse association strength function we used (Equation 1).

$$\left(\frac{1}{i\text{-}j}\right) \text{m} \qquad\qquad 1$$

That in turn is based on step co-occurrence. For the model, doing one step cues the next (Figure 2a). *Do a Trial,* the main goal of the task, gets retrieved at the end of every subgoal. With each retrieval its base-level activation gets a sharp increase that decays gradually over time (Figure 2c).

The difference at the postcompletion step between control and interruption is that in the interruption condition, the model lacks spreading activation from its working memory (Figure 2b). This is because when the interruption occurs, the model clears that resource of primary task representations so that secondary task representations may reside there. Then when the model resumes task execution, it restores only a part of its task context representation.

The model lacks context, and thus spreading activation, at resumption because the sparse representation of the episodic memory trace only records reference to one chunk encoding context (Altmann & Trafton, 2002), in this case the imaginal buffer chunk. Consequently when the model resumes it has only the imaginal buffer contents and not the goal buffer contents available to it.

In the example depicted in Figure 2b, the model is shown as having been interrupted before steps five and ten. Because of the way the model encodes its episodic memory and uses that to resume task execution, the chunks encoding steps four and nine are not referenced from the goal buffer chunk when it is time for the model to retrieve from long-term memory the chunks encoding how to perform steps five and ten.

The model produces PCE at resumption because total activation for the postcompletion step chunk and *Do a Trial* are approximately equal. In that context and with transient retrieval activation noise, each has an approximately equal chance of being retrieved.

Spreading Activation and Strength of Association

An architectural feature of ACT-R is that it uses a limited pool of spreading activation from sources—a chunk in a module's buffer—to associated chunks in declarative memory as one of its mechanisms of declarative retrieval. Our model used ACT-R's goal and imaginal buffers as sources of activation, each providing one unit of spreading activation.

Activation spreads from source chunks in ACT-R's buffers to chunks residing in ACT-R's declarative memory as a function of the *strength of association* between the value of each slot in source chunk *j* to chunk *i* in declarative memory (Anderson, 2007; Anderson et al., 2004). This gives ACT-R a way to adjust its behavior according to context as the strength of association indicates the probability that chunk *i* will be needed in context *j*. The limited pool of activation is divided equally among all the slots of source chunk *j*. This means that ACT-R implements a limited-capacity working memory.

Our model set strengths of association from each step's representation to the next at the beginning of each model run according to Equation 1.Association strengths remained static for the duration of each model run. Here, *j* is the serial position within the financial management task of the step encoded by a chunk representing some part of the model's current context (i.e., the last step performed). *I* is the serial position within the financial management task of the step encoded by an associated chunk in declarative memory. *M* is for a global ACT-R parameter to set the maximum association strength, set to 3.5 for this model.

For example, if the model had just performed the first step, *Order Ticker*, the association strength to the chunk encoding the second step, *Quantity,* would be 3.5. The strength of association to the third step, *Cost,* would be 1.75. This enabled associative chaining from the model's current context to the next procedure step. This produced a graded representation that decreased in strength with increasing psychological distance, a feature borrowed from Altmann and Trafton (2007).

Base-Level Activation

Base-level activation is an estimate that a declarative chunk will be needed in the future, given how recently it has been needed and how often it has been needed. This is another architectural feature of ACT-R and the idea is that given a limited capacity to retain information, those chunks not retrieved for a long time are allowed to have their activation decay below a threshold beyond which their retrieval will become less likely. Conversely, chunks that are retrieved frequently will have a high base-level activation contribution to their *total activation*. The model used ACT-R's default decay rate of 0.5 and activation noise of 0.2.

We assume spatial grouping of steps leads to Millerian (Miller, 1956) chunking of steps into groups, or subgoals. Anderson et al. (Anderson, Bothell, Lebiere, & Matessa, 1998), in their model of sequence memory, determined it crucial that sequence items be recalled in groups. Their model traversed a hierarchy of list item chunks, grouping chunks, and a chunk encoding the current list.

The financial management task model abstracted this process by adding a retrieval reference to the *Do a Trial* chunk upon completion of each financial management task subgoal: *Order Ticker*, *Quantity* through *Margin*, *Stock Exchanges* through *Review*, and *Complete Order* (see Figure 1). Each retrieval reference boosted *Do a Trial's* base-level activation. The idea is that after completing one subgoal, the task main goal is retrieved and used to retrieve the next subgoal. Therefore *Do a Trial's* base-level activation tended to be relatively high.

The postcompletion step happened to be needed immediately after a retrieval reference to *Do a Trial* (after completion of the preceding subgoal). Furthermore, a long time might have elapsed since the postcompletion step's last retrieval, especially when there had been two 15s interruptions during the trial. The relatively long time elapsed between retrievals of the postcompletion step lead to much decay of its base-level activation. Meanwhile, *Do a Trial* had received four retrieval references, one at the end of each of the subgoals. Each retrieval reference contributes to a chunk's base-level activation.

This combination of the postcompletion step's decay and *Do a Trial's* repeated retrieval was crucial for the model's commission of PCE at resumption. Because of these base-level activation mechanics the postcompletion step would then need a large quantity of spreading activation to have enough total activation to be retrieved reliably at postcompletion step time. Otherwise since *Do a Trial* had the second-highest retrieval activation because of its high base-level activation, it might be retrieved instead of the post-completion step's representation.

An Example Model Run

The model started its run by retrieving a procedure step representation. Because its context at the time would indicate that it was starting the task and the first step is most associated with starting, the first step would usually be the procedure step representation retrieved. After that the model simply looped through its basic behavioral cycle until it either finished a trial of the financial management task or until it was interrupted.

During the interruption, the model cleared its representations of its financial management task context from its working memory constructs—the goal and imaginal buffers—and replaced them with ones representing the interrupting task. At the end of 15s the financial management task interface replaced the interrupting task's, whereupon the model detected that its visual environment had changed back to the financial management task and so then it initiated its resumption subroutine.

When the model resumed the financial management task it began so by retrieving an episodic chunk. Because which episodic chunk retrieved was a function of base-level activation and transient noise, the most recent episodic chunk was usually the one retrieved.

The episodic chunk held a reference to an imaginal buffer chunk, which the model copied to the imaginal buffer. That imaginal buffer chunk held a record of the subgoal's steps completed at the time the episodic chunk was created. The restored imaginal buffer chunk provided the link necessary to retrieve the next step's representation at resumption.

The imaginal buffer chunk could contain references to as many as four step representations, all previous to the next correct step and all having varying strengths of association to it. This means that the limited activation source from the imaginal buffer could be divided by up to four.

Furthermore, the farther away in the procedure those steps were from the postcompletion step, the weaker their strength of association, and so the less source activation would propagate to the retrieval of the postcompletion step. The eighth step associated less strongly to the postcompletion step than did the ninth, but the eighth step took as much of the imaginal buffer's activation as the ninth.

Expressed in terms of maximum association strength, when it was time to perform the postcompletion step the imaginal buffer chunk spread only 25/48ths of available activation to the postcompletion step ($\approx$1.8 with :mas = 3.5). Roughly *half* of the activation source available from the imaginal buffer was diverted away from retrieval of the postcompletion step because of the presence of the previous steps' representations.

The model predicted more PCE for interrupted steps than non-interrupted steps because although the goal buffer chunk also held a reference to the just-completed step, the episodic chunk only encoded the imaginal buffer chunk. And because only one other goal slot was occupied, the association from the ninth step to the postcompletion step would get half of goal's available spreading activation, 1.75 units. Thus with the goal buffer chunk present the postcompletion step would get twice as much spreading activation as when the goal buffer chunk was absent due to interruption. This was enough to make the difference between reliable postcompletion step execution and equal chance of PCE when combined with base-level activation.

Furthermore, because *Do a Trial* got retrieval references four times during each trial—including once immediately before the postcompletion step—it tended to have a much higher base-level activation than did the postcompletion step. So when the model's only source of context representation was the imaginal buffer chunk and the task context was time to perform step the postcompletion step, the postcompletion step and *Do a Trial* would have similar amounts of total activation. Transient noise added at retrieval time (a standard feature of ACT-R) could tip the balance one way or the other.

Model Fit

We used our model to simulate data from 1,000 subjects. This large number of model runs allowed effects to converge on the model's true predictions. The model's means closely matched those of the participants, r = .976, RMSD = .0334. Figure 3 plots the model's means against the participants' means and 95% confidence intervals.

Discussion

PCE's distinction from anticipation is illustrated by comparison of their rates. If PCE were simply a matter of an anticipation error happening to fall at the last step then PCE and anticipation rates should be identical. However, Figure 3 shows clearly that the two error types are different.

What makes PCE unique is that it is a product of: 1) goal base-level activation decay below that of a competing goal's, 2) working memory structures with limited capacity to spread activation to long-term memory retrieval, 3) the size and structure of working memory representations—a preceding, large subgoal meant there were more items in working memory that would steal some of the available spreading activation away from the postcompletion step's retrieval, and 4) some context representation was not immediately available upon resumption.

Issues

Rather than learning the task, the model relied on assumptions about task representation structure. However, those are based on previous efforts with regard to sequence learning and memory (Anderson, Bothell, Lebiere, & Matessa, 1998a) and are also congruent with well-established methods of task analysis, particularly GOMS (Card, Moran, & Newell, 1983). We adapted some procedural and structural aspects of the Anderson et al.

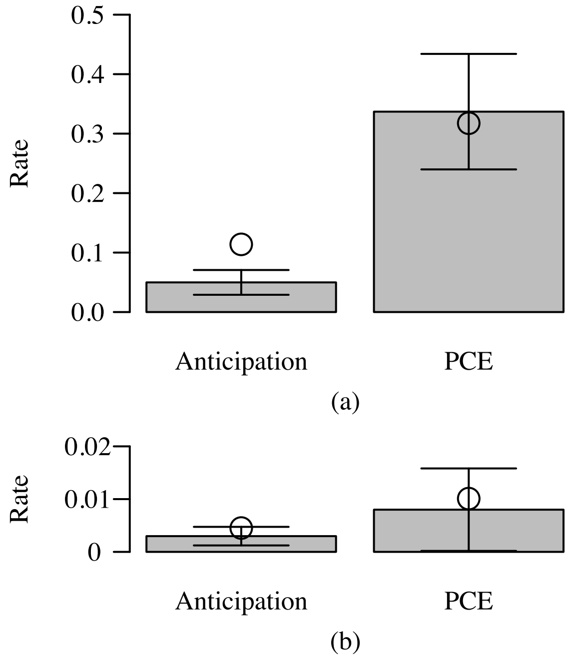

Figure 3: Mean error rates, human (bars) and model (circles). Error bars display the 95% confidence interval of the mean. Panel a depicts interruption trials, controls in b.

model because procedures are a kind of sequence. Mapping spatial groupings of task interface widgets to procedural representation groups of steps led to two important portions of the model's explanation of PCE in the interruption paradigm: groups of procedure steps held in working memory and high availability of the supergoal, *Do a Trial*.

Implications

The model explains PCE partially as a result of working memory constraints, following in the footsteps of Byrne and Bovair (1997). This implies that it should also explain PCE as a function of working memory capacity as their model did. In fact, this model has done just that with very little change (Tamborello and Trafton, submitted). Its anticipation error performance, while imperfect, suggests that the model should be extendable to other types of systematic procedural error, such as perseverations.

The decay process in the model has a cost, which is that suspended goals are forgotten gradually, making them harder to resume. The model carries with it an implied assumption that goals are retrieved at the outset of task execution, and then may decay from working memory before they are actually executed. With respect to PCE, the model implies that the default tendency is to make such errors, not avoid them.

Overall the model is encouraging with regard to our ultimate goal of developing a unifying framework of human error. But it is also encouraging from the standpoint of developing models of human procedural memory and execution, since the same cognitive systems are involved. Eventually it may also prove useful for models of error detection and recovery.

Acknowledgments

We are grateful to Laura M. Hiatt, Sangeet Khemlani, and Magdalena Bugajska for their helpful comments. This research was performed while the first author held a National Research Council Research Associateship Award at the US Naval Research Laboratory.

References

Altmann, E.M., & Trafton, J.G. (2002). Memory for goals: Activation-based model. *Cognitive Science, 26(1)*, 39-83.

Altmann, E. M., & Trafton, J. G. (2007). Timecourse of recovery from task interruption: Data and a model. *Psychonomic Bulletin & Review, 14(6)*, 1079-1084.

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological Review, 111(4)*, 1036-60.

Anderson, J. R., Bothell, D., Lebiere, C., & Matessa, M. (1998). An integrated theory of list memory. *Journal of Memory and Language, 38*, 341–380.

Byrne, M. D., & Bovair, S. (1997). A working memory model of a common procedural error. *Cognitive Science: A Multidisciplinary Journal, 21(1)*, 31-61.

Byrne, M. D., & Davis, E. M. (2006). Task structure and postcompletion error in the execution of a routine procedure. *Human Factors, 48(4)*, 627-38.

Card, S. K., Moran, T. P., & Newell, A. (1983). *The psychology of human-computer interaction*. Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Chung, P. H., & Byrne, M. D. (2008). Cue effectiveness in mitigating postcompletion errors in a routine procedural task. *International Journal of Human-Computer Studies, 66(4)*, 217-232.

Gray, W. D. (2002). Simulated task environments: The role of high-fidelity simulations, synthetic environments, and laboratory tasks in basic and applied cognitive research. *Cognitive Science Quarterly, 2(2)*, 205-227.

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

Miller, G. A. (1956). The magical number seven, plus or minus two. *Psychological Review, 63*, 81-97.

Ratwani, R. M., & Trafton, J. G. (2010). A generalized model for predicting postcompletion errors. *Topics in Cognitive Science, 2(1)*, 154-167.

Ratwani, R. M., & Trafton, J. G. (2011). A real-time eye tracking system for predicting and preventing postcompletion errors. *Human-Computer Interaction, 26(3)*, 205-245.

Tamborello, F. P., II, & Trafton, J. G. (submitted). A Long-Term Memory Competitive Process Model of Procedural Error, Part II: Working Memory Load and Capacity. *The 12th International Conference on Cognitive Modeling.*

Trafton, J. G., Altmann, E. M., & Ratwani, R. M. (2011). A memory for goals model of sequence errors. *Cognitive Systems Research, 12*, 134-143.