A Long-Term Memory Competitive Process Model of a Common Procedural Error, Part II: Working Memory Load and Capacity

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

Abstract

Postcompletion error (PCE) is 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. A computational cognitive model developed previously for PCE in an interruption paradigm extends to a working memory load and capacity paradigm. The model explains PCE in terms of long-term declarative memory mechanisms, opposing base-level activation with spreading activation.

Keywords: computational cognitive model; human error; human-computer interaction; 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 tends to have a higher incidence rate than chance slips and is very resistant to training (Byrne & Davis, 2006). Our goal is to understand the cognitive structures and processes underlying PCE and, ultimately, why people make procedural errors of all kinds.

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 have stochastic components, 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).

PCEs are pervasive and can occur during routine performance of a task. Furthermore, PCE rates consistently increase when cognition is stressed in some way. For example, if people are interrupted just before the PCE step (Ratwani & Trafton, 2010) or if working memory load is high (Byrne & Bovair, 1997).

We are developing a unified theory of PCE. A unified framework is important because one cognitive system, i.e. the human mind, produces PCE in all circumstances. Getting the explanation correct for one stressor type then acts as a constraint on explaining the next type.

Byrne and Bovair (1997) constructed one model of PCE that explained it in terms of working memory. Their model assumed a hierarchical goal representational structure derived from a GOMS (Card, Moran, & Newell, 1983) analysis of an experiment task also reported in their study. Their CAPS model (Just & 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. However, it is not clear how their model would explain PCE beyond the working memory capacity paradigm.

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 behavioral contexts at the time of their encoding. The strength of these memories decay as an exponential function of time. 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).

The decay process has a cost, which is that suspended goals are forgotten gradually, making them harder to resume. If goals are prospectively set at the outset of task execution, they may decay from working memory before it is time to execute them. With respect to PCE, this implies that the default tendency is to make such errors, not avoid them.

The model described in this report represents our attempt to construct a unifying explanation for PCE in multiple paradigms. To that end this model 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. The current study extends another model of PCE originally developed for an interruption paradigm (Tamborello & Trafton, 2013) and extends to Byrne and Bovair's working memory loading paradigm.

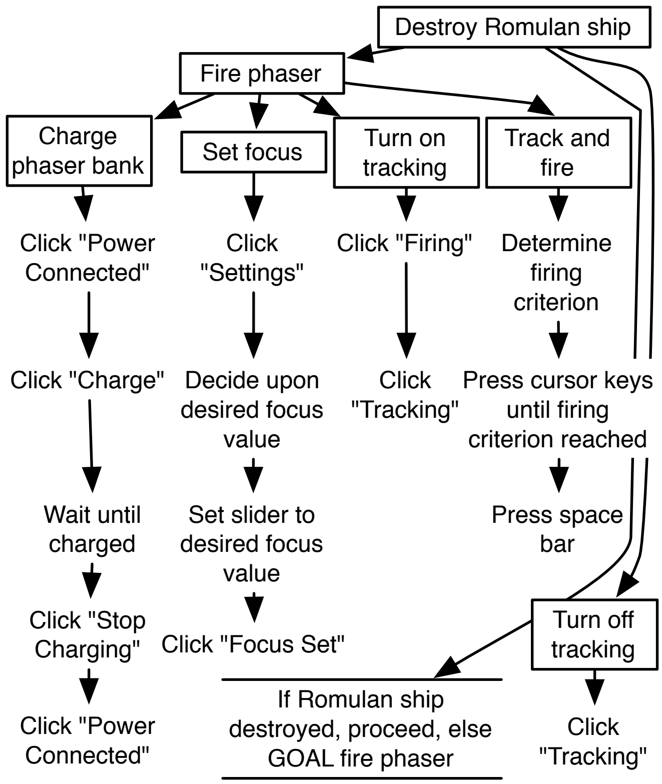

Figure 1: Goal structure of Byrne and Bovair's (1997) phaser task. Closed rectangles represent the task goal and subgoals while unenclosed text represents individual steps. The open-ended rectangle represents a decision to be made based on the task environment's status. Adapted from "A working memory model of a common procedural error," by M. D. Byrne and S. Bovair, 1997, *Cognitive Science, 21:1*, p. 43. Copyright 1997 by the Cognitive Science Society.

Experiment

We applied our model to Byrne and Bovair's (1997) postcompletion phaser task from their second experiment. For our purposes the important points about that task were:

1. Working memory load varied on a within-subjects basis, implemented by a three-item memory task.
2. Participants varied in their own working memory capacities. Byrne and Bovair treated this as a two-level factor, split on the median.
3. Participants had to follow a specific procedure.
4. The spatial layout of the task grouped steps by proximity. This encouraged use of an intuitive heuristic ("do all the items in the cluster"), as well as having an isolated "clean-up" step at the end. Byrne and Bovair's own GOMS analysis of their phaser task resulted in a hierarchical task representation that they used in their CAPS model (Figure 1).
5. 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 trial on the phaser task (e.g. attempting to start a new trial by clicking Power Connected). The PCE rate was the number of PCEs divided by the number of opportunities to make a PCE.

Model

This ACT-R 6 (Anderson, 2007a; Anderson et al., 2004b) model, developed originally for another paradigm, is described here. The model used cyclic, activation-based retrieval from long-term memory of the task step representations encoded as *procedure-step chunks*. At each step there were two sources of retrieval activation: 1) spreading activation from the contents of the goal and imaginal representations, 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.

The amount of spreading activation from the goal and imaginal buffers to the chunk encoding the postcompletion step increased with advancing task context because of the inverse association strength function (Equation 1), which in turn is based on step co-occurrence. Association strengths were static for the duration of each run. Here, j is the serial position within the phaser task of the step encoded by a chunk representing the model's context (i.e., the last step performed), i is the serial position within the phaser task of the step encoded by an associated chunk in declarative memory, and m is a global ACT-R parameter to set the maximum association strength, set to 3.5 for this model.

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

However, *Destroy Romulan Ship,* the main goal of the task, was retrieved at the end of every subgoal of the task. With each retrieval its base-level activation increased. The postcompletion step, being its own subgoal, immediately follows another subgoal so that when it is time to retrieve the postcompletion step's representation, *Destroy Romulan Ship* has just received more base-level activation. Because ase-level activations decay gradually over time for all chunks and because *Destroy Romulan Ship* had been retrieved much more recently than the postcompletion step, the former's base-level activation was much greater, leading to its total activation being approximately equal to the latter's.

The model implemented working memory loading by filling three slots of the imaginal buffer chunk with chunks each representing a letter of the alphabet—an abstracted three-back memory task. The presence of those chunks reduced the amount of activation spread to the procedure step chunk from all of the activation available to the imaginal buffer to just one-fourth of it.

The model implemented individual differences in working memory capacities by taking different values of two of ACT-R's global parameters: activation noise (.4 for low-capacity and .225 for high-capacity) and imaginal activation (.8 and 2.25, respectively), similar to Byrne and Bovair's (1997) manipulation of CAPS' activation ceiling parameter.

Principles of the PCE Model

Basic Behavioral Cycle The model operated cyclically by retrieving from long-term memory details specifying each procedure step. The model carried out the individual operations necessary to accomplish that step of the task as specified by the representation it had just retrieved. Then it used association from that step to retrieve the next.

Individual mental operations of the model—*productions*, in ACT-R parlance—were few and were sparse in their representations so that they could be generic. For example, looking for a button and then clicking it was abstracted from the details of the individual button. The details of each button were specified by the declarative representation retrieved at one point in the cycle.

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.

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, 2007a; Anderson et al., 2004b). 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 run according to Equation 1.

For example, if the model had just performed the first step, *Power Connected*, the association strength to the chunk encoding the second step, *Charge,* would be 3.5. The strength of association to the third step, *Stop Charging,* 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 rationale 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.

Built into the model is the assumption that 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 phaser task model abstracted this process by adding a retrieval reference to the *Destroy Romulan Ship* chunk upon completion of each phaser task subgoal: *Charge phaser bank*, *Set focus*, *Turn on tracking, Track and fire* and *Turn off tracking* (see Figure 1). Each retrieval reference boosted *Destroy Romulan Ship's* base-level activation. This is meant to abstract an Anderson et al. (1998)-like hierarchical goal traversal process: After completing one subgoal, the task main goal is retrieved and used to retrieve the next subgoal. Therefore *Destroy Romulan Ship's* base-level activation tended to be relatively high.

The postcompletion step occurred immediately after a retrieval reference to *Destroy Romulan Ship* (after completion of the preceding subgoal). Furthermore, enough time would have elapsed since the postcompletion step's last retrieval for the postcompletion step chunk's base-level activation to decay substantially. Meanwhile, *Destroy Romulan Ship* had received four retrieval references, one at the end of each subgoal. Since each retrieval reference contributes to a chunk's base-level activation, *Destroy Romulan Ship* tended to have a much higher base-level activation than any individual step's representation, including the PC step.

Because of its high base-level activation, *Destroy Romulan Ship* usually had the second-highest total activation. If working memory was loaded or of low capacity, transient retrieval noise would sometimes give *Destroy Romulan Ship* higher total activation than the postcompletion step's representation. This combination of the postcompletion step's decay and *Destroy Romulan Ship'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 overcome *Destroy Romulan Ship's* base-level activation so that the postcompletion representation could be retrieved reliably at postcompletion step time.

Suspended and Resumed Goals

We assume the model suspended its goals for nine seconds as subjects performed the *track and fire* step. This is because it required an intensive period of perceptual-motor tracking, followed by a move of attention, reading a phrase of text, and then deciding whether to continue or to restart. This process is significant because the model does not immediately regain all of its task context representation upon resumption.

The model incorporated Altmann and Trafton's (Altmann & Trafton, 2002) Memory for Goals (MfG) mechanism for encoding a retrievable episodic trace of every action it performed. As part of its execution cycle, the model created an episodic chunk which contained a unique identifier as well as a reference to the model's current imaginal buffer chunk. The imaginal buffer chunk contained references to each of the procedure-step chunks encoding steps performed in the current subgoal. It, together with the goal buffer chunk, comprised the model's context representation. The episodic chunk acted as a trace of partial context, containing a record of the model's progression through the task in the form of a sequence of the imaginal buffer chunk.

However, the episodic trace only recorded the model's problem state representation, the imaginal buffer chunk. It did not record the model's control state representation, its goal chunk. Because the goal buffer chunk was not saved by the episodic trace it was therefore unavailable at resumption, and so the model had only the imaginal buffer chunk to act as activation source when it retrieved the next step to perform. Furthermore, when the model was in the working memory load condition, three-fourths of the imaginal buffer's activation spread to the three working memory task chunks, rather than to the PC step's representation in declarative memory. This meant that *Destroy Romulan Ship's* base-level activation would often overcome the total activation of the postcompletion representation because of its relatively reduced spreading activation.

Structure of Task Representation The model relied on one critical assumption about the structure of procedure representation: Steps are organized into groups of one to four and the procedure's main goal (e.g., to perform a trial of the phaser task) is retrieved after the completion of each subgoal. This assumption is adapted from the Anderson et al. (1998) model of sequence memory, although it is congruent with Byrne and Bovair's model. This assumption was important for shaping the structure of the model's working memory representations and influencing the declarative retrieval process.

Competition of Spreading Activation and Base-Level Activation at the Postcompletion Step Action selection was a product of declarative retrieval, in turn a product of two key theoretical constructs in ACT-R's declarative memory system: 1) spreading activation from the model's current contextual representation, and 2) base-level activation of the step memories. Critically, under certain conditions these two constructs worked in opposition and balanced each other so that at the postcompletion step the model was just as likely to retrieve the postcompletion step's representation as *Destroy Romulan Ship*.

Base-level activation, however, favored chunks that had been retrieved recently and often. The main goal also competed for retrieval with the procedure-step chunks on the basis of total activation. When the model retrieved *Destroy Romulan Ship* it started a new trial or subgoal. *Destroy Romulan Ship* was retrieved often and so tended to have a very high base-level activation. When the model was at the end of a subgoal it had just retrieved *Destroy Romulan Ship*, and so total activation—the sum of both retrieval mechanisms—tended to be very close for the chunk encoding the correct next step and for *Destroy Romulan Ship*.

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 iterated through its basic behavioral cycle until it either until it got to the shoot step.

During the shoot step, the model cleared its representations of its task context from its working memory constructs—the goal and imaginal buffers—and replaced them with ones representing manipulation of the target and shooting. At the end of nine seconds the model initiated its resumption subroutine.

When the model resumed its main task goal 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 then 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. Restoration of the imaginal buffer chunk provided the link necessary to retrieve the next step's representation at resumption. However, this was a relatively weak link compared to other times when the model also had its goal buffer chunk as an available source of spreading activation.

The model predicted higher rate of PCE for loaded trials than non-loaded, and for low working memory capacity than high working memory capacity because these two factors both impacted the amount of retrieval activation available to spread from the imaginal buffer chunk. Furthermore, 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. Thus with the goal buffer chunk present the postcompletion step would get much more spreading activation as when the goal buffer chunk was absent (Figure 2b). This was enough to make the difference between reliable postcompletion step execution and equal chance of PCE when combined with base-level activation.

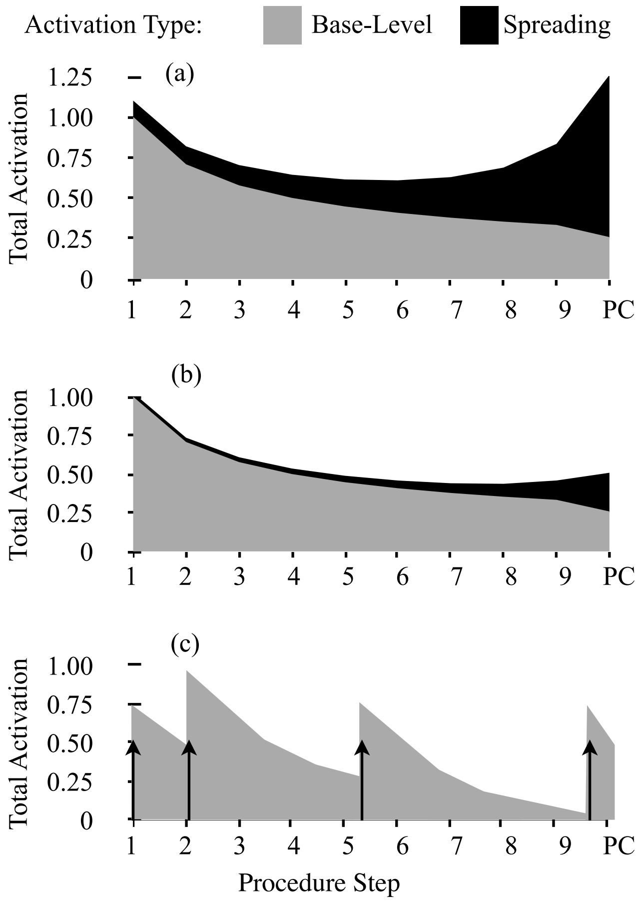

Figure 2: Base-level and spreading activations of the model's postcompletion step chunk as a function of context in no load (a) and load (b) conditions. X-axes indicate the step to be performed, by step ordinal number and by designating the postcompletion step as "PC." The chunk encoding the main goal of the task is associated to subsequently performing the first step. It receives retrieval references at the end of each subgoal, indicated by arrows (c). Note how in the load condition (b) at the time to perform the PC step, the PC step's representation has approximately the same total activation as does the main task goal (c); this is what causes PCE.

Furthermore, because *Destroy Romulan Ship* 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 (Figure 2c). So when the model's only source of context representation was the imaginal buffer chunk and the task context was time to perform the postcompletion step, the postcompletion step and *Destroy Romulan Ship* would have similar amounts of <u>total activation</u>. 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, $r = .920$, RMSD = .0657. Figure 3 plots the model's means against the participants' means and 95% confidence intervals.

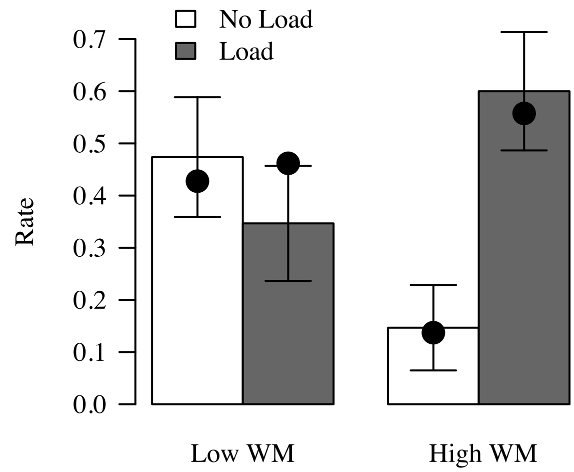

Figure 3: Mean postcompletion error rates, human data from Byrne and Bovair's Experiment 2 phaser task (bars) and model (circles), as a function of memory load and capacity. Error bars display the 95% confidence interval of the mean.

Discussion

We developed a multi-paradigm PCE model. The model depends upon sparse procedural memory representation with details filled in from declarative memory. This structure is important because declarative memory mechanisms form the core of the model's explanation of PCE. Our key mechanism for generating PCE is base-level activation's opposition with spreading activation at the PC step.

The main task goal is retrieved repeatedly during task performance as the model traverses the task goal hierarchy. Each retrieval increases its base-level activation slightly. Meanwhile, because each step's representation is retrieved only once during task performance, the correct retrieval of each step is relatively more dependent upon spreading activation.

However, with a memory loading task's chunks referenced in the imaginal buffer chunk at the postcompletion step, spreading activation in the load condition was divided over the four slots filled in the imaginal buffer chunk, compared to only one slot filled in the no load condition. So the difference in spreading activation becomes one fourth available to retrieve the PC step versus all of the imaginal buffer's spreading activation, respectively.

The division of current problem state representation and current control state information is well-supported by other modeling and also neuroimaging work (Anderson, 2007b; Anderson et al., 2004a). The present model implemented this structure in the imaginal and goal buffers, respectively. It played an important role in reducing the amount of spreading activation available to retrieval of the PC step chunk.

When the main task goal is resumed after the *track and fire* step, only the problem state information is retrieved from episodic memory. The control state information is constructed anew as the model goes along. But because this new control state representation lacks a reference to the last step representation retrieved, the control state cannot contribute spreading activation to the retrieval of the next procedure step as it does normally when it does not follow goal resumption. Now that spreading activation is only available from the imaginal buffer chunk, the total activation for the PC step chunk is reduced to being close enough to the total activation of the task main goal for transient noise to occasionally make the task main goal more active. This in turn leads to the model starting a new trial of the task, therefore committing a PCE.

There is one important caveat to the model. Rather than learning the task, the model relied on assumptions about task representation structure. However, this assumption stems from GOMS analysis, which is a well-supported technique. Furthermore, the success of the current model lends additional credence to this style of hierarchical, symbolic goal representation.

We take these results as converging evidence in favor of our account of PCE. Now the model is nearly constrained by two datasets from differing paradigms. If the revised model presented in this report can provide as good a fit to the previous report's (Tamborello & Trafton, 2013) data then we should be in good position to begin to address other types of systematic procedural error, such as anticipation, perseveration, and capture error.

We (, 2013) were correct in our speculation that our model could capture working memory effects in PCE rates. And this is because the model explains PCE partially as a result of working memory constraints, as in Byrne and Bovair (1997). This also represents to some extent validation of our strategy to pursuing a unifying cognitive theory of systematic procedural error. This is a good development because the same cognitive systems—namely, the human mind—are involved in all error types. Eventually it may lead to a cumulative science that proves 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: An 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. (2007a). *How can the human mind exist in the physical universe?* New York, NY: Oxford University Press.

Anderson, J. R. (2007b). *How can the human mind exist in the physical universe?* New York, NY: Oxford University Press. Retrieved from Google Scholar.

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004a). An integrated theory of the mind. *Psychological Review*, *111*(4), 1036-60. doi: 10.1037/0033-295X.111.4.1036

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004b). An integrated theory of the mind. *Psychological Review*, *111*(4), 1036-60. doi: 10.1037/0033-295X.111.4.1036

Anderson, J. R., Bothell, D., Lebiere, C., & Matessa, M. (1998). An integrated theory of list memory. *Journal of Memory and Language*, *38*, 341–380. Retrieved from http://act-r.psy.cmu.edu/publications/pubinfo.php?id=116

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

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. Retrieved from Google Scholar.

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

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

Ratwani, R. M., & Trafton, J. (2010). A generalized model for predicting postcompletion errors. *Topics in Cognitive Science*, *2*(1), 154-167. doi:10.1111/j.1756-8765.2009.01070.x

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. doi:10.1080/07370024.2011.601692

Tamborello, F. P., & Trafton, J. G. (2013). A long-term competitive process model of a common procedural error. In *Proceedings of the 35th annual conference of the cognitive science society.* Austin, TX: Cognitive Science Society.

Trafton, J. G., Altmann, E. M., & Ratwani, R. M. (2009). A memory for goals model of sequence errors. In *Proc. 9th international conference of cognitive modeling.* Retrieved from Google Scholar.