An Activation-Based Model of Routine Sequence Errors

Laura M. Hiatt (laura.hiatt@nrl.navy.mil)
J. Gregory Trafton (greg.trafton@nrl.navy.mil)
U.S. Naval Research Laboratory
Washington, DC 20375 USA

Abstract

Models of errors during routine sequential action are typically interface-independent. We present here evidence that different task spatial layouts, however, result in different patterns of sequence errors. We explain this data by expanding upon the Memory for Goals framework's activation-based, sequential process to include environmental (such as visual) contextual cues, as well as a richer priming structure. We show a strong qualitative and quantitative fit to experimental data.

Keywords: Priming; routine sequence errors; cognitive modeling.

Introduction

Sequence errors are errors in the order of steps ideally taken to complete a task. Typically, routine sequence errors take the form of either repeating previous steps (perseveration errors), or skipping one or more steps (anticipation errors) (Reason, 1984).

Various accounts exist for sequence errors (Cooper & Shallice, 2006; Botvinick & Plaut, 2006). One successful model is the memory for goals (MFG) model (Trafton, Altmann, & Ratwani, 2011), which uses episodic control codes to direct step-by-step progression through the task. At any point, the most activated code is selected as the current step to work on; this is based on activation strengthening (i.e., frequency and recency of use), as well as activation from priming effects (i.e., associated cues from the current goal).

These current theories of sequence errors are interface-independent; that is, they do not depend on spatial specifics of the task interface. We present here evidence, however, that changes in a task's spatial layout can lead to different patterns of sequence errors. This difference could be explained by spatial reasoning, but there are not currently spatial reasoning theories integrated into sequence error theories. Instead, we explain this data with a model that utilizes the main principles of MFG, including activation and priming; crucially, however, it expands MFG's notion of priming with a richer priming structure, and allows it to capture a fuller environmental context (Hiatt & Trafton, 2013; Thomson, Bennati, & Lebiere, 2014).

In this work, priming can stem from anything in working memory, including visual representations, and can be the result of explicit correspondences between concepts, as well as more implicit relationships, such as co-occurrence. This fuller view on priming allows our model to explain the changes in error patterns stemming from different task interfaces, in large part because our account of priming includes visual cues. We next describe an experiment showing how task layout affects error patterns; then we discuss our model, show that we provide a good qualitative and quantitative account for the data, and end with a discussion of the implications of our approach.

Experiment

Forty-three participants performed a version of Ratwani and Trafton's financial management task (Ratwani & Trafton, 2011). The task is a form-filling task where steps need to be performed in a specific order to buy or sell stocks. The layout primarily consists of two columns; unlike previous versions of the task, where the task step sequence moves down the columns (a columnar layout; e.g., Ratwani & Trafton, 2011; Trafton et al., 2011), here, the step sequence goes across before going down (a horizontal layout; see Figure 1). Each step, save the first and the last, consists of selecting the appropriate widget, and then selecting the appropriate value before hitting a "submit" button. The first step consisted of choosing a stock to trade; the last involved hitting a "Complete Order" button (a post completion step; Byrne & Bovair, 1997). The task had no place-keeping, so upon completing a step, there were no cues about the correct step to take next.

Participants performed three training trials, followed by 20 testing trials. Occasionally, after completing a step, the screen cleared and the participants were interrupted to perform a simple arithmetic task; the interruption lasted 15 seconds. After the interruption, the participants were expected to resume the task and continue with the next appropriate step. For five of the testing trials, there were three interruptions; another five trials had two interruptions each; five trials had one interruption each; and five trials had zero interruptions. The trial order was randomly determined outside the knowledge of the participants to keep participants from guessing whether a step would be followed by an interruption.

Occasionally, participants made an error by selecting the wrong step to work on next. The highest percentage of the errors by participants occurred when resuming from this interruption, especially given that there was no place-keeping; these errors are what we analyze and model here. We describe these sequence errors in terms of how far ahead or behind of the correct step the selected step was. So, if a step is repeated, it is considered a -1 (perseveration) error, since the selected step is one step behind the correct step. If a step is skipped, it is considered a +1 (anticipation) error, since the selected step is one step ahead of the correct step. If a step two steps back is repeated (such as performing step 7, "Associate", after performing step 8, "Order Info" in Figure 1), that would be a -2 error, since the selected step is two behind the correct next step; and so forth. If an incorrect step was selected, the

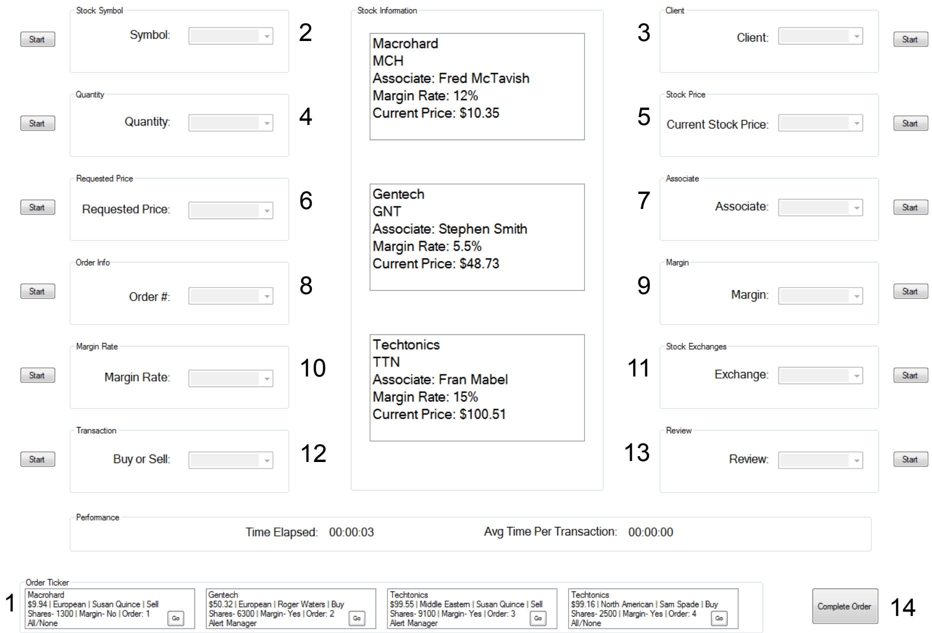


Figure 1: Horizontal stock trader interface. The steps are numbered to indicate the order in which they should be completed; they are shown here for illustrative purposes only.

system beeped and the correct step was highlighted in a red color to allow the participant to recover and continue.

The data from the horizontal task show both similarities and differences with the columnar versions of this task (Figure 2). As with the columnar versions (and other routine sequence error tasks, as well), the most common error for the horizontal task was the immediate (-1) perseveration error, and there were more perseveration errors overall than anticipation errors. Also in accordance with the columnar data, the distribution of errors clusters around the +/-1 errors, and falls away in both directions as the error type gets farther from the correct step (Altmann, Trafton, & Hambrick, 2014).

The horizontal data, however, also show a different pattern of this gradation: namely, a higher proportion of +/-2 errors that occur, compared to +/-1 errors, caused by its distinct spatial layout; an effect which other approaches are unable to explain. The horizontal data also have a wider distribution spread, overall. To preview our approach, we explain the higher proportion of perseveration errors as due to differences in strengthening and priming activation values, which are true regardless of the task interface. In contrast, the difference between the two patterns of data stems primarily from visual priming, which can lead to interface dependent effects. We discuss this further below.

Model Framework

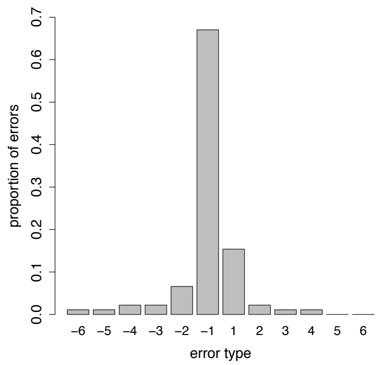
We investigate our account of error prediction within the cognitive architecture ACT-R/E (Trafton et al., 2013), an embodied version of ACT-R (Anderson, Bothell, Lebiere, & Matessa, 1998). ACT-R is an integrated theory of human cognition in which a "production system operates on a declarative memory" (Anderson et al., 1998). In ACT-R, activation of memories has three main components – strengthening, priming, and noise – which are added together to represent a memory's total activation. We next discuss each in turn.

Activation Strengthening

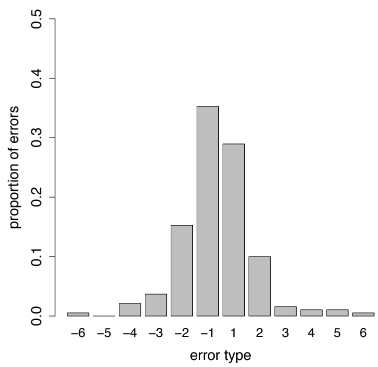
ACT-R's well-established theory of activation strengthening has been shown to be a very good predictor of human declarative memory (Anderson et al., 1998; Anderson, 2007; Schneider & Anderson, 2011). Intuitively, activation strengthening depends on how frequently and recently a memory has been relevant in the past. It is designed to represent the activation of a memory over longer periods of time and, generally, is highest right after the memory has been accessed in working memory, slowly decaying as time passes. Activation strengthening is calculated according to:

$$A_s = \ln\left(\sum_{j=1}^{n} t_j^{-d}\right)$$

where n is the number of times a memory i has been *referenced* (e.g., used in working memory) in the past, t_j is the time that has passed since the jth reference, and d is the a strengthening learning parameter, which defaults to 0.5. Importantly, the negative exponent in this equation implies that recent memories are more differentiated from each other than memories farther in the past.



(a) Distribution of sequence error after an interruption in a columnar interface. Data is from Trafton et al., (2011).



(b) Distribution of sequence errors after an interruption in a horizontal interface. Note the higher proportion of +/-2 errors, and wider distribution spread, that occur.

Figure 2: Contrasting patterns of errors are produced when task versions have different spatial layouts.

Activation Priming

While priming has long been a part of the ACT-R framework (e.g., Anderson, 1983), we adopt a newer, richer notion of priming as part of our approach (Harrison & Trafton, 2010; Hiatt & Trafton, 2013; Thomson et al., 2014). One substantial difference is that, here, activation priming sources from any part of the model's working memory, including the model's goal, intermediate problem representations, and visual representations of what the model is looking at. It then spreads, along associations, to other memories related to those in working memory.

Another main difference is the richer structure of associations. Relevant to our discussion here, associations can be created not only because of explicit correspondences, but also due to *co-occurrence* and *residual* relationships. Co-occurrence associations are created between memories i and j when they are both referenced in working memory at the same time. Residual associations are created between memories that have been referenced in working memory in temporal proximity to one another, even if they are not in working memory at the same time.

Once established, associations have an associated strength value which affects how much activation is spread along them. Mathematically, the strengths (S_{ji}) are:

$$S_{ji} = mas \cdot e^{\frac{-1}{al \cdot R_{ji}}}$$

$$R_{ji} = \frac{f(N_i C_j)}{f(C_j) - f(N_i C_j) + 1}$$

These equations reflect two parameters: *mas*, the maximum associative strength; and *al*, the associative learning rate. The function f tallies the number of times that memory j has been referenced, either independently (C_j) or at similar times to when i has been referenced ($N_i C_j$). An associative strength, intuitively, reflects how strongly a memory, when currently being referenced in working memory, predicts that a memory it primes will be referenced next, and are a function of how often the two memories are referenced by working memory at the same time, versus how often each one is referenced in working memory without the other (represented by R_{ji}). These equations are explained further in Hiatt and Trafton (2013); residual associations are discussed further in Thomson et al. (2014). The associative strengths' qualitative properties are what are key here: namely, that residual associations are typically weaker than co-occurrence associations.

To summarize, associative priming provides the models built in this framework with a rich network for spreading activation that can capture correspondences between memories that are frequently relevant at roughly the same time, as well as correspondences between memories or concepts of different modalities. We rely on both of these features of priming for our model, described below.

Activation Noise

The activation noise of a memory is drawn from a logistic distribution with mean 0 and standard deviation the parameter σ_c. It is a transient value that changes each time it is used, and models the neuronal noise found in the human brain.

Perception and Action

Finally, the model interacts with the world using ACT-R/E's built-in functionality for interacting with the world. Models can view computer interfaces on a simulated monitor; they can act on the world by pushing keys on a simulated keyboard and clicking a simulated mouse.

Activation-Based Model of Sequence Errors

The model's general principles are that it uses activation strengthening and priming to drive progression through a sequential task's steps. At all times, the model maintains in working memory a representation of its goal of completing the task. Before any given step, the model decides what step to perform next by first performing a free retrieval of an episodic code representing the last step that has been completed (or, if there is already an episodic code in working memory, it simply retains it). Once there is an episodic code in working memory, it then performs a free retrieval of a step, and considers the retrieved step to be the correct next step to perform. It then repeats this process to move on to the next step. At any of these points, as we will show below, priming can come from any item in working memory, including what the model is working at, and can greatly influence the progression of the model through the steps.

Specific to this task, we assume that the representation of the step includes a visual location for where the step is located on the screen, which the model uses as a guide when moving to complete that step. The model is abstract in the sense that it was not concerned with the actual values to fill in to the widgets; instead, its primary responsibility is to attempt to complete the steps of the task in the correct order during each trial. After selecting a widget to work on, it thus clicks the submit button without filling in any values. At the end of a task, the entire working memory is cleared before the model begins the next trial. The model does not perform the post-completion step (e.g., step 14 of Figure 1).

During an interruption, the arithmetic task requires the use of the entire working memory, and so all stock task-related memories (including goals and episodic codes) are removed from working memory. Upon completion of the interrupting task, the model adds a new task goal to working memory, and decides what step to perform next using the process described above. Here, however, in addition to performing a free retrieval of an episodic code, the model looks at the position of the previous step in the interface before attempting to retrieve the next step. Eye-tracking data collected during the experiment showed that, after an interruption, participants look at the correct next step only 13% of the time. Participants, instead, first looked most often at the previous step upon resumption (15%), with the rest generally looking at locations or steps in close proximity to the previous step (such as the step above or below). We assume that this wide spread of where participants look is, in part, due to error in the eye tracker as well as due to error stemming from participants noisily remembering the last location at which they were looking before the interruption. We do not have a theoretical model of visual location memory, but instead model the visual location noise by adding a small amount of Gaussian noise to the position of the last step, and focusing the model's visual attention on the step nearest to that noisy location.

As the sequence process unfolds, many associations are created between the various components involved. Critical to our approach, associations are created between nearby sequential steps, as well as between the visual representation of a step and the next step. Figure 3 illustrates associations for the eighth step of the stock trader experiment; all of the associations are created from co-occurrence except for those between 6.RequestedPrice and 8.OrderInfo, and between 8.OrderInfo and 10.MarginRate, which are residual associations.

During a normal, non-resumption step, the free retrieval

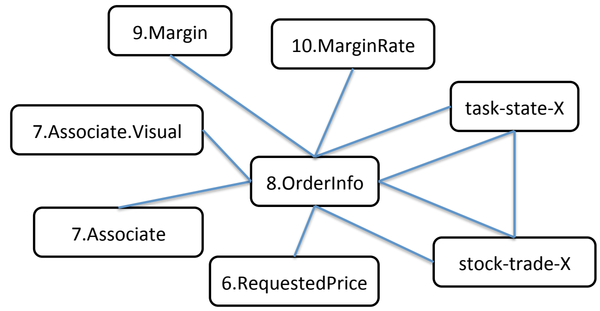


Figure 3: Illustrative associations for the eighth step of the stock trader experiment, Order Info. "Task-state-X" is a placeholder for various episodic codes associated with Order Info over time. "Stock-trade-X" is a placeholder for various goals associated with Order Info over time; episodic codes are associated with the goal in working memory that they co-occur with. Other associations are made, as well, but for clarity we omit those not relevant to our discussion.

of the episodic code is highly affected by both the very near recency of the episodic code (activation strengthening), and from priming from the current goal. Then, the free retrieval of the next step is primarily influenced by priming stemming from the episodic code of the prior step in working memory. These priming and strengthening effects result in a very low error rate during normal task sequence execution.

When resuming after an interruption, however, the model's path is not as clear-cut. For example, since some time has passed since the previous episodic code was last in use, it is more apt to be confused with previous episodic codes, potentially leading to an incorrect retrieval; this is exacerbated by the new goal in working memory, which does not provide priming cues to the previous episodic code. The model also may look at the wrong previous step due to its noisy visual location memory. These potentially incorrect sources of activation mean that the model does not always retrieve the correct step to perform after an interruption. Additionally, even if both these steps do go correctly, priming from the previous episodic code may lead to an anticipation error, because of the residual associations between non-sequential steps. These competing sources of activation comprise the crux of our approach and are explained in more detail, below.

Model Predictions

The model makes a number of activation-based predictions for post-interruption errors in this task. There are two factors that contribute to the final set of activations for a resumption step: the prior episodic code that is retrieved, and what the model is looking at. Different outcomes of these two potential process affect the overall pattern of errors for the retrieved step:

- Retrieve the correct episodic code: Here, priming activation from the correct, prior episodic code biases the model towards the correct answer. It also spreads some activation, however, to the +1 anticipation step via residual associations (such as how 6.RequestedPrice primes 8.OrderInfo in Figure 3), leading to a possibility of a +1 anticipation error. Occasionally, residual associations can also result in a +2/+3/etc. error.

- Retrieve the wrong episodic code: This happens because of activation strengthening decay. After an interruption, recent episodic codes are close enough in activation that earlier codes may be retrieved. This always results in a bias towards perseveration. A bias towards errors of -1 are the most common, here, but it is possible that errors of type -2/-3/etc. could stem from an incorrectly retrieved prior episodic code as well.
- Look at the right previous step: This biases the model towards performing the correct step next.
- Look at the wrong previous step: Because of visual proximity, the wrong step being looked at is either above or below the previous one. This will bias the model towards the incorrect step being looked at.

Based on these potential process errors, the model makes several predictions for sequence errors. First, the model predicts more perseveration errors than anticipation errors because, intuitively, the model is more likely to retrieve an incorrect past episodic code (potentially leading to an anticipation error) than it is to retrieve an incorrect step based on residual priming (potentially leading to an anticipation step). More technically, the difference in activation strengthening between the past episodic codes is less than the difference in activation priming that an episodic code spreads to the correct vs. future step, leading to more errors occurring there. It follows that this difference also predicts that the most common error type is the -1 perseveration errors. These predictions are interface-independent.

Because of its inclusion of visual priming, the model also predicts the pattern of errors will differ depending on the task's spatial layout. In the previous columnar layout, looking at the wrong previous step spreads activation to the +/-1 steps. In the horizontal layout, however, looking at the wrong previous step spreads activation to the +/-2 steps. Based on this, the model makes two spatial-dependent predictions for the horizontal task version.

First, it intuitively predicts a relatively high proportion of +/-2 errors because of the increase in priming activation those steps receive when the model looks at the incorrect step. Second, it predicts a wider distribution spread, overall. This is because the set of steps commonly competing for retrieval (e.g., {-2, -1, 0, 1, 2}) is larger than the set commonly competing in the columnar task version (e.g., {-1, 0, -1}), making the distribution of steps ultimately selected more spread out.

To reiterate, the key difference between this model and the original memory for goals model is the depth to which priming is utilized by the model. In the MFG model, priming derived from explicit correspondences between the goal and episodic code, and so environmental context (such as priming from visual sources) was not a factor; this makes it unable to capture correspondences between visual objects and other memories, and so it does not predict any shift in error patterns between the two task interface layouts. In addition, MFG does not include residual priming associations in its account, making it very difficult for it to account for +2 errors, even in the columnar version of this task. Finally, in MFG, priming relied upon explicit correspondences between features, some of which were assumed *a priori*, unlike our model which assumes no associations to begin with and builds up its rich network as it trains for, and then tests on, the task.

Model Fit

We ran the model 43 times to simulate data from each of the 43 participants from the horizontal stock trader study. Before beginning testing, the model first performs 3 training trials, where it assumes it is being instructed with the task sequence as it moves through the steps. During these trials, the framework of associations is set up that it will rely upon as it continues on to perform the 20 testing trials, where it continues to learn and update associations as well. Interruptions during the experiment had the same structure as in the original study.

ACT-R/E includes several parameters that affect activation dynamics and, thus, model behavior. The associative learning rate, which affects the rate at which associations are strengthened, was set to 6.5, which represents a fairly brisk rate of learning. There is no standard value for this parameter. The maximum associative strength was within its normal range at 3.0. The activation noise parameter σ_c was 0.08, which is also within its typical range. All other parameters were set to their default values.

We compared the proportion of errors of each type that the model made with the proportion of errors of each type from the study; this allows us to compare the data both qualitatively (overall error trends) and quantitatively (specific distribution of results). The results are shown in Figure 4. Overall, the model's results matched the data very well, with $R^2 = 0.99$ and $RSE = 3.3$. It also qualitatively matches the data's trends, with -1 perseveration errors being the most common error type, and with a higher proportion of +/-2 errors and relatively wider distribution.

Discussion

This work utilizes the underlying principles of memory for goals – that sequential steps are driven by episodic codes, and that those episodic codes are selected based on activation – while expanding its scope. In addition to the strengthening, goal-based priming and noise activation components present in the memory for goals model, our model provides an expanded view of priming that includes priming activation from all items in working memory, and fosters a richer priming structure enabled by additional types of associations. This allows our approach to account for data from sequential tasks with different spatial layouts, something the MFG model was not previously able to do, while also keeping it connected with existing MFG models of sequential errors, postcompletion errors, and recovery time that have been shown to

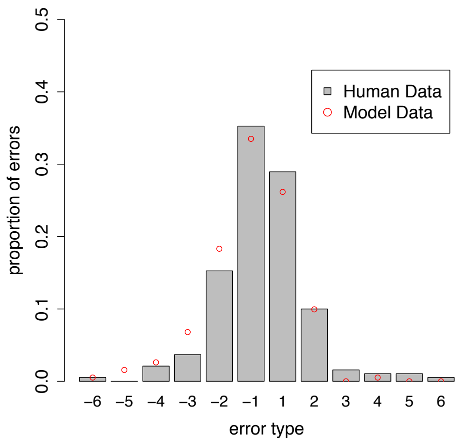


Figure 4: Graph showing the proportion of each type of sequence error from both the experiment and the model.

be successful (Altmann & Trafton, 2002, 2007; Tamborello, II & Trafton, 2014).

Other models of routine sequence errors, such as the interactive activation network (IAN) model (Cooper & Shallice, 2006) and the simple recurrent network (SRN) model (Botvinick & Plaut, 2006), also ignore the specifics of the task interface, and so cannot account for the differences in error patterns that results from an interface layout shift. The ideas behind our expanded priming approach, however, could apply to IAN, which uses environmental and contextual activation to select between schemas that determine the next step.

Although our model does not account for other types of errors, such as capture errors, it does provide some intuition about how those errors take place. Capture errors, for example, occur when a task sequence switches, mid-execution, to a task sequence from a different, but usually related, task; for example, checking e-mail after sitting down at a computer when one originally intended to check the weather. Capture errors, intuitively, involve much environmental context and visual cues, which can be accounted for by our approach.

Acknowledgments

We would like to thank Malcolm McCurry and Kevin Zish for providing the experimental data we utilize here. We would also like to thank Frank Tamborello for his helpful comments. This work was supported by the Office of the Secretary of Defense / Assistant Secretary of Defense for Research and Engineering (LH) and the Office of Naval Research (GT). The views and conclusions contained in this paper do not represent the official policies of the U.S. Navy.

References

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

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

Altmann, E. M., Trafton, J. G., & Hambrick, D. Z. (2014). Momentary interruptions can derail the train of thought. *Journal of Experimental Psychology: General*, *143*, 215-226.

Anderson, J. R. (1983). A spreading activation theory of memory. *Journal of Verbal Learning and Verbal Behavior*, *22*(3), 261–295.

Anderson, J. R. (2007). *How can the human mind occur in the physical universe?* Oxford University Press.

Anderson, J. R., Bothell, D., Lebiere, C., & Matessa, M. (1998). An integrated theory of list memory. *Journal of Memory and Language*, *38*(4), 341–380.

Botvinick, M. M., & Plaut, D. C. (2006). Such stuff as habits are made on: A reply to cooper and shallice (2006). *Psychological Review*.

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

Cooper, R. P., & Shallice, T. (2006). Hierarchical schemas and goals in the control of sequential behavior. *Psychological Review*.

Harrison, A. M., & Trafton, J. G. (2010). Cognition for action: an architectural account for "grounded interaction". In *Proceedings of the Annual Meeting of the Cognitive Science Society.*

Hiatt, L. M., & Trafton, J. G. (2013). The role of familiarity, priming and perception in similarity judgments. In *Proceedings of the Annual Meeting of the Cognitive Science Society.*

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.

Reason, J. T. (1984). Absent-mindedness and cognitive control. *Everyday Memory, Actions and Absent-mindedness*.

Schneider, D. W., & Anderson, J. R. (2011). A memory-based model of hick's law. *Cognitive Psychology*, *62*(3), 193–222.

Tamborello, II, F. P., & Trafton, J. G. (2014). A generalized process model of human action selection and error and its application to error prediction. In *Proceedings of the Annual Meeting of the Cognitive Science Society.*

Thomson, R., Bennati, S., & Lebiere, C. (2014). Extending the influence of contextual information in ACT-R using buffer decay. In *Proceedings of the Annual Meeting of the Cognitive Science Society.*

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

Trafton, J. G., Hiatt, L. M., Harrison, A. M., Tamborello, II, F. P., Khemlani, S. S., & Schultz, A. C. (2013). ACT-R/E: An embodied cognitive architecture for human-robot interaction. *Journal of Human-Robot Interaction*, *2*(1), 30-55.