Predicting Postcompletion Errors using Eye Movements

Raj M. Ratwani
George Mason University
4400 University Drive, Fairfax, VA
rratwani@gmu.edu

J. Malcolm McCurry & J. Gregory Trafton
Naval Research Laboratory
4555 Overlook Ave. SW, Washington D.C.
{trafton} {mccurry}@itd.nrl.navy.mil

ABSTRACT
A postcompletion error is a distinct type of procedural error where one fails to complete the final step of a task. While redesigning interfaces and providing explicit cues have been shown to be effective in reducing the postcompletion error rate, these methods are not always feasible or well liked. This paper demonstrates how specific eye movement measures can be used to predict when a user will make a postcompletion error. We describe a real-time eye gaze system that provides cues to the user if and only if there is a high probability of the user making a postcompletion error.

Author Keywords
Errors, Eye Tracking, Interruptions

ACM Classification Keywords
H5.2. [Information interfaces and presentation]: Graphical-user interfaces.

INTRODUCTION

The occurrence of routine procedural errors is all too common in both simple and complex systems. While most errors are generally harmless, there have been instances where errors have resulted in disastrous outcomes. For example, a routine procedural error by a New York stock exchange clerk resulted in the liquidation of 11 million shares of stock (approx 500 million dollars) as opposed to the intended 11 million dollars worth of stock simply because numerical values were typed in the wrong field on a computer display [5]. This routine error caused massive chaos on the stock exchange floor and a market plummet.

The goal of this paper is to examine whether a user's behavior can be used to predict when they are going to make an error in computer based tasks. We focus on one specific type of procedural error: the postcompletion error [1, 3, 4]. A post completion error is associated with an action that is required after the main goal of the task has been completed [3]. One common example of a postcompletion error is leaving your ATM card at the teller machine after making a transaction. Such errors are thought to occur because the main goal of the task has been satisfied, yet there is still an additional step to complete the task. After satisfying the main goal of the task, one moves on to the next goal before completing the last step in the current task [3].

The frequency of post completion errors has been shown to be exacerbated by interruptions. While interruptions increase all types of errors [2], interruptions that occur just prior to the postcompletion step drastically increase the postcompletion error rate [9].

Given the frequency and potential costliness of postcompletion errors, the obvious question becomes how to reduce these error rates. There are three ways of reducing postcompletion error rates: system redesign, explicit visual cues, and error prediction (i.e. provide cues when needed). The most effective way of reducing postcompletion errors is to redesign the task interface or system to eliminate the postcompletion step [4]. For example, instead of inserting an ATM card to make a transaction, the system can be designed such that only a swipe of the card is necessary. This redesign eliminates the postcompletion step of retrieving the card after the transaction. While this solution is the most effective, it is not always feasible given large complex systems.

When system redesign is not possible, another way to reduce postcompletion error rates is to consistently use explicit visual cues to remind the user of the correct action [6]. Unfortunately, there are several issues with this method, such as the fact that the cue must be blatantly obvious to be effective [6-7,11]. These persistent blatant cues may not be aesthetically pleasing in terms of interface design, users may become annoyed by their constant presence, and the effectiveness of the visual cues may attenuate over time.

A third possible method for reducing postcompletion errors is to predict when users will make an error and provide a visual cue at that moment. By only providing visual cues when there is a high probability that the user will make an error, several issues with the constant presence of a visual cue are avoided. In this paper we present a mechanism for predicting postcompletion errors.

How can postcompletion errors be predicted? The prominent theories of postcompletion errors are memory

CHI 2008, April 5–10, 2008, Florence, Italy.
Copyright 2008 ACM 978-1-60558-011-1/08/04…$5.00.

based [3]. For example, [3] showed that working memory load is associated with postcompletion error rates: when working memory load is high, the final step of the task is more likely to be forgotten and a postcompletion error is likely. Unfortunately, using memory theories to predict postcompletion errors in complex tasks is difficult because the role of memory is not apparent at every step a user makes. If the goal is to predict postcompletion errors in a timely fashion (i.e. less than one second), assessing working memory load is not feasible.

Instead of focusing on memory, we focus on the perceptual level and examine users' explicit eye movements as they interact with the user interface. Examining the patterns of eye movements when users make an error as compared to when they do not make an error may allow for the prediction of postcompletion errors. To do this, we tracked participants' eye movements as they performed a complex computer based task with a specific postcompletion step; participants were interrupted in order to increase the number of postcompletion errors. We then examined different eye track measures to determine if it is possible to predict when a user will commit a postcompletion error.

EXPERIMENT

We sought to elicit a large number of postcompletion errors in order to examine participants' eye movement patterns to generate several predictors of when an error would occur. The predictors were evaluated with logistic regression to determine the feasibility of predicting the occurrence of an error.

Previous literature pointed us towards three specific predictors for the logistic regression model. First, error responses are generally associated with longer reaction times compared to correct responses. Second, cognitive processing demands have been shown to be positively correlated to total number of fixations [8]. A greater number of fixations until the postcompletion action may indicate the user is taking more time to actually retrieve the next step in the task from memory. Finally, because memory theories of postcompletion error suggest that the final step is simply forgotten [3], a predictor at the perceptual level may be whether or not the postcompletion action on the task interface is fixated on. A higher probability of error may be associated with the user not fixating on the postcompletion action.

Method

Participants

Thirty-eight GMU students participated for course credit.

Task and Materials

The primary task was a complex production task called the *sea vessel* task (based on [9]). The goal is to successfully fill an order for two different types of sea vessels by entering in order details through various widgets on the interface (see Figure 1). There is a correct sequence of actions required to complete the task, thus, any deviation was considered an error. After entering information in each of the widgets, the order was processed by clicking the *process* button. Finally, the participant must click the *complete contract* button to finish the order.

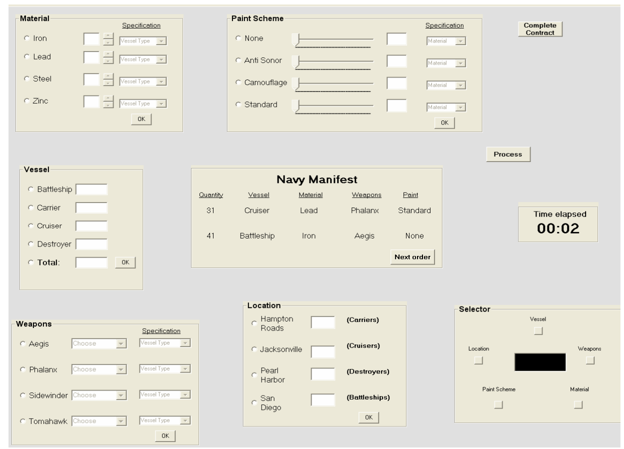

Figure 1. Screenshot of the Sea Vessel Production Task.

Upon clicking the *process* button a small window popped up indicating the order has been submitted and the window provided details about how many sea vessels were ordered. This indicator served as a false completion signal [9, 10]; consequently, clicking the *complete contract* button is a postcompletion step. Failure to click the *complete contract* button after acknowledging the signal constituted a postcompletion error. When an error was made the computer emitted a beep alerting the participant.

The secondary interruption task required participants to answer addition problems with four single digit addends; the interrupting task window completely occluded the primary task interface.

Design and Procedure

Each order on the sea vessel task constituted a single trial. Participants were given instructions and completed two trials as part of training; one had no interruptions and one had two interruptions. All subjects were proficient at the task before beginning the actual experiment.

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.

There were six predefined interruption points in the sea vessel task. For the purposes of this paper, the interruption point of interest is after the *process* button is clicked and the pop-up window is acknowledged (just prior to the post completion step). During the experiment there were a total of 12 interruptions (6 interruption trials x 2 interruptions in each trial); each lasting 15 seconds. Participants were instructed to answer as many addition problems as possible in this time interval. The interruptions were equally distributed among the six interruption locations, providing two postcompletion interruption opportunities.

Measures

Keystroke and mouse data were collected for every participant. Eye track data were collected using a Tobii 1750 operating at 60hz. A fixation was defined as a minimum of five eye samples within 30 pixels (approx 2˚ of visual angle) of each other, calculated in Euclidian distance. The time interval of interest in regard to predicting an error following the interruption was the resumption lag [1]. The resumption lag was the time interval from the moment the sea vessel task was restored following the interrupting task to the first action back on the sea vessel task. For the purposes of this study we were only concerned with postcompletion errors, thus all analyses focus on the postcompletion steps.

Results and Discussion

Time Cost of Interruptions

We first examined whether the interruptions were disruptive to primary task performance in terms of reaction time. To do this, we compared the resumption lags from the interruption trials to the inter-action intervals (IAI) from the control trials. The IAI was the average time between clicking the *process* and *complete contract* buttons in the control trials. The resumption lags (M = 3041 ms) were significantly longer than the IAIs (M = 1112.9 ms), $F(1,37) = 78$, MSE = 905499.5, $p<.001$, suggesting that the interruptions were disruptive in terms of time cost.

Postcompletion Error Rates

Next, we examined whether the interruptions influenced the postcompletion error rates. Participants made significantly more postcompletion errors immediately following an interruption (M = 51.3%) as compared to the control trials with no interruption (M = 2.6%), $F(1,37) = 67.1$, MSE = .07, $p<.001$. Thus, being interrupted just prior to the postcompletion step substantially increased the error rate; this is consistent with the working memory load explanation of postcompletion errors.

Predicting Postcompletion Errors

In order to predict postcompletion errors we used a logistic regression analysis with the data from steps in the control trials and the postcompletion steps in the interruption trials where an interruption occurred just prior to the postcompletion action (i.e. interruptions just prior to the *complete contract* button). Three theoretically motivated predictors were used in the logistic regression: postcompletion step time, total fixation count and postcompletion fixation.

The postcompletion step time predictor was a measure of the amount of time from the acknowledgement of the false completion signal until the next interface action was made. In the interruption trials this would be the time from the end of the interruption until the first action on the primary task. The total number of fixations was a count of the number of fixations that occurred during this total time period. The third predictor we used was a dichotomous variable indicating whether or not the correct postcompletion action button on the task interface was fixated on (i.e. fixating on the complete contract button) during the total time period (0 = no, 1 = yes). The outcome variable was whether the action was an error or not (0 = no error, 1 = error). Equation 1 shows the results of the analysis:

Equation 1: Predicted logit of Error = intercept + (total fixations x .63) + (postcompletion fixation x -5.7)

The overall logistic regression equation was significant, $\chi^2(2) = 57.84$, $p<.001$. The log odds of an error occurring were positively related to the total number of fixations ($p<.001$) and negatively related to postcompletion fixation ($p<.001$), see Table 1. As the count of total fixations increased the likelihood of making an error increased. Also, if participants did not look at the postcompletion action button the likelihood of making an error increased. Postcompletion step time itself was not a significant predictor ($p=.2$), presumably because it is highly correlated with total number of fixations.

Predictor	β	SE β	Walds χ^2	df	p
Intercept	-5.62	.87	-6.39	1	.001
Postcompletion Step Time	-.001	.001	-1.2	1	.21
Total Fixation Count	.63	.19	3.42	1	.001
Postcompletion Fixation	-5.7	.87	6.61	1	.001

Table 1. Logistic Regression Table.

How well does the model fit the current data? The c statistic is a measure of model fit. This statistic represents the proportion of postcompletion pairs with different observed outcomes for which the model correctly predicts a higher probability for observations with the event outcome (e.g. commit error) than the probability for nonevent observations (e.g. no error). The c value for this logistic regression is .98, which means that for 98% of all possible pairs of postcompletion actions, the model correctly assigned a higher probability to postcompletion actions that were errors than to actions that were not errors. A c value of .98 is considered excellent.

Using the logistic regression equation it is possible to determine the probability of a participant making an error given their total number of fixations and whether or not they fixate on the postcompletion action button. Figure 2 shows how the probability of making an error changes given different values of these predictors. The dots at the top and bottom of the graph are raw data representing error and non-error status that have been jittered to show density. The dashed line represents predicted probabilities of committing an error when the postcompletion action button *is* fixated on; the solid line represents cases where the postcompletion action button *is not* fixated on.

Figure 2 illustrates several important points. First, when participants fixate on the postcompletion action button within 11 fixations their probability of making an error is under 20% (dashed line). As the number of fixations increases, the probability of making an error increases despite the fact that they have fixated on the correct button. After 17 fixations, there is a 75% chance an error will be made. When the postcompletion action button is not fixated on (solid line) there is a very high probability of making an error (~50%), even with few fixations. This probability quickly increases with each fixation.

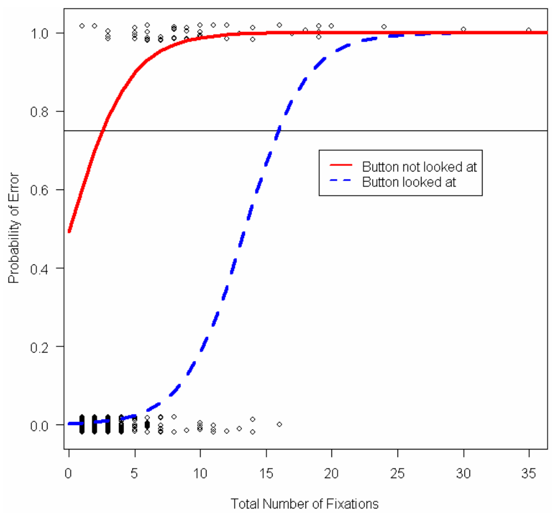

Figure 2. Probability of Postcompletion Error

GENERAL DISCUSSION

Our logistic regression model has shown that it is possible to predict when users will make postcompletion errors based on their eye movements. Total number of fixations and whether or not the postcompletion action button was fixated on were strong predictors of whether an error was going to be made. Can the logistic regression model actually be used to prevent postcompletion errors?

The advantage of using eye movement data as predictors of postcompletion errors is that the eye movement data can be analyzed in real-time and feedback can be provided to the user immediately. While this work is at a preliminary stage, we have used these data to develop a real-time eye gaze system. This system uses our logistic regression model to predict when a user will make an error. Our system monitors and analyzes their eye movements at the postcompletion step and a visual cue is provided if and only if the probability of making an error is greater than 75%. Thus, if the user fixates on the postcompletion action button and does not make a response within 17 fixations the system provides a visual cue in the form of a red arrow pointing at the correct button. In addition, if the user has not fixated on the postcompletion action button and has made more than four fixations, the same visual cue is provided.

This type of real-time eye gaze system prevents many of the problems associated with systems that provide constant visual cues to their users. Only providing cues to the user when there is a likely chance of the user needing the cue is effective and less disruptive. Further, in safety critical environments more extreme measures can be taken as opposed to simply providing a cue (e.g. explicit warning messages etc.). Our current system serves as a proof of concept; we are currently refining and testing this system.

Acknowledgements

This work was partially supported by grant 55-8122-08 from the Office of Naval Research to Greg Trafton. We thank Simon Li for letting us use his primary task as a model to develop our task.

REFERENCES

1. Altmann, E. M., & Trafton, J. G. (2002). Memory for goals: An activation-based model. *Cognitive Science 26*:39-83.
2. Bailey, B. P., & Konstan, J. A. (2006). On the need for attention aware systems: Measuring effects of interruption on task performance, error rate, and affective state. *Journal of Computers in Human Behavior* 685-708.
3. Byrne, M. D., & Bovair, S. (1997). A working memory model of a common procedural error. Cognitive Science 21:31-61.
4. Byrne, M. D., & Davis, E. M. (2006). Task structure and postcompletion error in the execution of a routine procedure. *Human Factors 48*:627-638.
5. Casey, S. (1998). Set Phasers on Stun, 2 ed. Aegean Publishing Company, Santa Barbara.
6. Chung, P. H., & Byrne, M. D. (in press). Cue effectiveness in mitigating postcompletion errors in a routine procedural task. *International Journal of Human-Computer Studies*.
7. Czerwinski M., Cutrell E. & Horvitz E. (2000) Instant messaging and interruption: Influence of task type on performance, in: C. Paris, N. Ozkan, S. Howard & S. Lu (Eds.) *Proceedings of OZCHI 2000: Interfacing Reality in the New Millennium*, Academic Press, 356-361
8. Just, M.A. & Carpetner, P.A. (1976). Eye fixations and cognitive processes. *Cognitive Psychology*. 8: 441-480.
9. Li, S. Y. W., Cox, A. L., Blandford, A., Cairns, P., & Abeles, A. (2006). Further investigations into post-completion error: the effects of interruption position and duration., Proceedings of the 28th Annual Meeting of the Cognitive Science Society. Erlbaum.
10. Reason, J. (1990). *Human error*. Cambridge University Press.
11. Trafton, J. G., Altmann, E. M., & Brock, D. P. (2005). Huh, what was I doing? How people use environmental cues after an interruption. Proceedings of the Human Factors and Ergonomics Society 49th Annual Meeting.