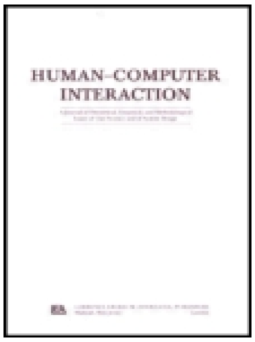


Human–Computer Interaction

ISSN: 0737-0024 (Print) 1532-7051 (Online) Journal homepage: https://www.tandfonline.com/loi/hhci20

A Real-Time Eye Tracking System for Predicting and Preventing Postcompletion Errors

Raj M. Ratwani & J. Gregory Trafton

To cite this article: Raj M. Ratwani & J. Gregory Trafton (2011) A Real-Time Eye Tracking System for Predicting and Preventing Postcompletion Errors, Human–Computer Interaction, 26:3, 205-245

To link to this article: https://doi.org/10.1080/07370024.2011.601692

Published online: 02 Sep 2011.

HUMAN–COMPUTER INTERACTION, 2011, Volume 26, pp. 205–245
Copyright © 2011 Taylor & Francis Group, LLC
ISSN: 0737-0024 print / 1532-7051 online
DOI: 10.1080/07370024.2011.601692

A Real-Time Eye Tracking System for Predicting and Preventing Postcompletion Errors

Raj M. Ratwani[1] **and J. Gregory Trafton**[2]
[1]*George Mason University*
[2]*Naval Research Laboratory*

Procedural errors occur despite the user having the correct knowledge of how to perform a particular task. Previous research has mostly focused on preventing these errors by redesigning tasks to eliminate error prone steps. A different method of preventing errors, specifically postcompletion errors (e.g., forgetting to retrieve the original document from a photocopier), has been proposed by Ratwani, McCurry, and Trafton (2008), which uses theoretically motivated eye movement measures to predict when a user will make an error. The predictive value of the eye-movement-based model was examined and validated on two different tasks using a receiver-operating characteristic analysis. A real-time eye-tracking postcompletion error prediction system was then developed and tested; results demonstrate that the real-time system successfully predicts and prevents postcompletion errors before a user commits the error.

1. INTRODUCTION

When performing a routine procedural task, people occasionally make errors despite having the correct knowledge of how to perform the task (Reason, 1990). These types of skill-based errors (Rasmussen, 1982) are generally known as procedural errors and have been documented in several different domains. The base rate of procedural errors is quite low and most of the time harmless. However, in high-risk domains such as aviation, medicine, or military settings, procedural errors can have serious consequences (Perrow, 1999). In fact, there have been several unfortunate disasters attributed to these types of errors (Reason, 1990).

Raj Ratwani is an applied cognitive scientist with an interest in human–computer interaction; he is senior research scientist at Perceptronics Solutions Inc. and an affiliate of the Psychology Department at George Mason University. **Greg Trafton** is a cognitive scientist with an interest in human–computer interaction; he is a section head in the Intelligent Systems section of the Naval Research Laboratory.

CONTENTS

Many researchers have studied human error with the overarching goal of being able to prevent these errors from occurring in the first place. There have been two general approaches to error prevention. One approach has been to identify error prone steps in a task and then to redesign the task to remove the error prone steps. For example, task analysis techniques have often been used to analyze the structure of a task to determine which steps are more prone to error (Barber & Stanton, 1996; Kirwan, 1992a, 1992b; Shappell & Wiegmann, 2001; Stanton & Stevenage, 1998); if possible the task can then be redesigned to remove these steps. A concrete example of preventing procedural errors by task redesign has been demonstrated with a specific type of procedural error called a postcompletion error.

Postcompletion errors are associated with forgetting a final step that occurs after the main goal of a task has been completed (Byrne & Bovair, 1997). For example, leaving a bank card at a teller machine after completing a transaction is a postcompletion error. This type of error commonly occurs on teller machines that dispense the bank card after a transaction (e.g., retrieving the money). Most teller machines have now been redesigned such that users swipe their bank card, thus eliminating the postcompletion step from the task. In addition to teller machines, there are several other common tasks that have postcompletion steps (e.g., replacing the gas cap on a car, including an attachment on an e-mail message, retrieving the original document from a photocopier, etc.). The consequences of postcompletion errors in common tasks may be trivial, but postcompletion errors can occur in high-risk environments as well, and the consequences can be severe. For example, in January 2005 an accident occurred in which a freight train carrying hazardous chlorine gas collided with a standing local train; the accident resulted in nine deaths. The accident was attributed to an improperly lined switch, which sent the freight train down an unintended track. There had been a construction crew working on the track that day, and after the construction crew completed work for the day the crew forgot to return the mainline track switch back to the normal position (National Transportation and Safety Board, 2005). This was a postcompletion error: The crew's main goal was to complete the track work, and the step of returning the mainline track switch back to the normal position was to be performed after the main goal was accomplished.

Byrne and Davis (2006) have empirically demonstrated that by redesigning a task to remove the postcompletion step altogether, postcompletion errors can be eliminated. However, there are limitations to the task redesign method. First, although particular task steps might be removed because of the high likelihood of error, procedural errors can still occur on other task steps because of extraneous factors (e.g., fatigue, stress, distraction, etc.). Thus, this method cannot completely eliminate procedural errors from a task. Second, it is not feasible to redesign every task or system that is susceptible to procedural errors. There are several constraints that may prevent the redesign of all existing systems. Specifically, it is expensive to redesign and update some of the current large-scale systems, such as the flight management system in a cockpit or the control interface in a nuclear power plant control room. Generally, these large-scale systems are fully integrated, and one cannot easily remove and redesign a single component of the system without a massive overhaul. Further, aside from the obvious monetary cost of redesigning the actual system and implementing the new system, there may be additional retraining time for users and expenses associated with retraining.

A second approach to preventing procedural errors is to predict the circumstances under which a procedural error is likely to occur and then to attempt to prevent the user from making the error. Being able to predict when procedural errors might occur before the error actually occurs has been the holy grail of error research. The error prediction approach, however, requires a deep understanding of the cognitive mechanisms underlying procedural errors. Procedural error research, and the development of specific theories of procedural errors, has had a challenging history. Because of the difficulty in generating large-enough data sets to successfully study procedural errors (Gray, 2000, 2004), the majority of early research on procedural errors relied on data sets composed of observations of errors as they occur in the real-world or on self-report measures. The early studies led to the development of several taxonomies as well as general theories of human error (Baars, 1992; Norman, 1981; Rasmussen, 1982; Rasmussen & Jensen, 1974; Reason, 1990; Sellen & Norman, 1992). Only within the last 10 to 15 years has there been the development of specific testable process theories of distinct types of procedural errors (Altmann & Trafton, 2002; Byrne & Bovair, 1997; Byrne & Davis, 2006; Gray, 2000).

The recent advancements in understanding the theoretical underpinnings of procedural errors have allowed for the development of a predictive model of procedural errors, specifically for postcompletion errors. Ratwani, McCurry, and Trafton (2008) described a cognitive theoretic mathematical model to predict when a user will make a postcompletion error based on the user's behavior. The model relies on logistic regression, which is a statistical technique for predicting a discrete outcome (Tabachnick & Fidell, 2001); at a high level, logistic regression is similar to multiple regression. Logistic regression, however, can capture nonlinear relationships between the predictors and the outcome variable. Logistic regression provides the log odds (which can be converted to probability) of the outcome variable given continuous or discrete predictors; the purpose of the analysis is to predict the category of the outcome for individual cases (Peng, Lee, & Ingersoll, 2002). Ratwani et al. (2008) examined user's eye movement data immediately preceding a postcompletion step, developed a set of theoretical eye movement predictors based on these data, and used a logistic regression model to predict the likelihood of a postcompletion error. A comparison of the model's predicted probabilities to the actual occurrence of a postcompletion error (i.e., the individual cases of postcompletion errors) demonstrated that the model successfully predicted the majority of postcompletion errors in the data set upon which the model was based.

Although this predictive eye movement approach is promising and relieves some of the shortcomings of the task redesign approach, the feasibility of such an error prediction system is yet to be determined. The logistic regression model has only been applied to a single task; the model should to be applied to tasks different from the task upon which the model was originally based to determine the robustness of the model. Critically, a true predictive system was never created or tested by Ratwani et al. (2008). Thus, although the predictive approach may account for postcompletion errors after a task has been completed and the eye movement data have been analyzed, because the predictive model has never been implemented in a real-time eye-tracking system it is unclear how successful such a system would be at predicting and preventing postcompletion errors before they actually occur. The goal of this article is to (a) examine whether the cognitive-theoretic model of postcompletion errors is robust enough to adequately predict when a user will make a postcompletion error by applying the model to data sets from two different tasks, and (b) to determine whether the predictive model can be instantiated in an eye-tracking system to predict and prevent postcompletion errors in real time.

1.1. Background

Cognitive Components of the Predictive Model

The logistic regression model used to predict postcompletion errors leverages two different theoretical frameworks, a specific theory of postcompletion error (Byrne & Bovair, 1997) and a broader theory of goal memory (Altmann & Trafton, 2002, 2007). Both of these theories are activation-based memory accounts. The theory of postcompletion error, put forward by Byrne and Bovair, suggests that postcompletion errors are due to goal forgetting and inattention to the postcompletion step. Specifically, postcompletion errors occur because the postcompletion step of a task is not maintained in working memory and thus is not executed as part of the task. The main goal of a task and the subsequent subgoals are stored in working memory and must remain active to be executed. The main goal provides activation to the subgoals. When the main goal of a task is satisfied, the goal no longer provides activation to the subgoals; thus, the remaining subgoals may fall below threshold and will no longer be maintained in working memory. In the bank teller machine example used previously, the main goal of the task is to make a transaction, and when this is completed the final subgoal of retrieving the bank card may no longer be provided activation by the main goal. Consequently, this subgoal may fall below threshold and may not be executed resulting in the postcompletion error of leaving the bank card at the teller machine.

Byrne and Bovair (1997) developed a computational model of these processes and provided empirical support of the associated processes. Empirically, Byrne and Bovair manipulated the working memory load of participants with low and high working memory load capacities. They found that when participants with low working memory load capacities were given high working memory load tasks they were more likely to make a postcompletion error. There was no difference in postcompletion error rates for the high working memory load participants when the working memory load of the task was manipulated. These results provide strong support for the Byrne and Bovair theory.

The postcompletion error prediction equation also leverages a broader theory of goal memory, called the memory for goals theory (Altmann & Trafton, 2002, 2007). The memory for goals theory accounts for how the cognitive system remembers its goals; for example, when completing a hierarchical task, how does one manage to accomplish subgoals that are part of a more general high-level goal? The theory suggests that behavior is directed by the current most active goal and that the activation level of goals decay over time. For a goal to direct behavior, the goal must have enough activation to overcome interference from previous goals; thus, the goal must reach a certain threshold to actually direct behavior.

Goal activation is determined by two main constraints. The strengthening constraint suggests that the history of a goal (i.e., how frequently and recently the goal was retrieved) will impact goal activation. The priming constraint suggests that a pending goal can receive activation (priming) from an associated cue. These cues can be in either the mental or environmental context; for example, particular information in a task interface may provide priming of a pending goal. Thus, a goal can overcome decay and direct behavior if appropriate environmental cues are attended to (Ratwani & Trafton, 2008). These cues may prime the goal and boost activation, allowing the goal to overcome the interference level.

Altmann and Trafton (2002, 2007) have instantiated their theory in the ACT–R framework. Memory for goals is a general theory of memory, but several researchers have applied the theory to task interruption paradigms. There has been explicit empirical support for the decay of goals over time (Hodgetts & Jones, 2006b) as well as support for the strengthening (Trafton, Altmann, Brock, & Mintz, 2003) and priming constraints (Hodgetts & Jones, 2006a; Trafton, Altmann, & Brock, 2005). Together, these studies provide broad support for the memory for goals theory. It should be noted that there is considerable overlap between the predictions from the memory for goals theory as it is applied to an interruptions paradigm and other theories (Dodhia & Dismukes, 2009) that are specific to task interruption.

Real-Time Measures of Cognitive State. The postcompletion error prediction equation was directly based on the Byrne and Bovair (1997) and Altmann and Trafton (2002, 2007) memory theories. However, instead of focusing solely on memory processes, Ratwani et al. (2008) developed eye movement measures that co-occurred with memory processes proposed by these theories. One advantage of eye movement measures is that they can be collected in real time. Eye movement measures have been used as indicators of cognitive process in previous research (Just & Carpenter, 1976; Rayner, 1998). Using eye movement measures as predictors of the likelihood of a postcompletion error allows for the development of a real-time feedback system if the user's eye movements are analyzed in parallel with performance of the task. If there is a high likelihood of error, the user can be alerted that he or she may be likely to make a postcompletion error before the user actually makes an error.

Developing real-time measures of cognitive processes is not a new research endeavor. Several methods have been used to measure cognitive processes online and to provide feedback to users. For example, eye movement measures have been developed to discriminate between cognitive states on a variety of tasks, such as a user being alert or fatigued (Marshall, 2007). Real-time measures of cognitive state have been implemented in real-world systems to provide feedback to a user; for example, systems have been developed to monitor driver fatigue (Ji, Zhu, & Lan, 2004). Methods other than eye tracking have been used to assess cognitive state as well. For example, electroencephalography (EEG) has been used to assess the workload of air-traffic control operators in real time (Wilson & Russell, 2003).

There are, however, no real-time systems that can detect when a user will make a procedural error before the error is actually made. Using simple repetitive stimulus response tasks, researchers have shown that there is a correlation with brain activity and error actions (Eichele et al., 2008). However, these correlations were demonstrated after an extensive amount of data postprocessing; these correlations were not demonstrated in real time.

Relying on eye movement measures, as opposed to behavioral measures from EEG or functional magnetic resonance imaging (fMRI), has several advantages. The temporal resolution of eye movement data collection allows for near immediate processing as users interact with their environment, whereas other techniques typically incur substantial time delays. Of importance, collecting eye movement data is noninvasive and nonintrusive. By embedding cameras in a computer monitor (e.g., Tobii eye trackers as well as others), eye-tracking technology is now sophisticated enough that one can sit in front of a computer monitor, untethered, while still being monitored by the eye tracker. This method of eye tracking allows for a free range of motion; thus, user interaction with the eye-tracking computer monitor is similar to user interaction with any other computer monitor and is quite natural. Setup time with an eye tracker is extremely fast compared to other methods such as EEG. Calibrating the eye tracker to the user requires a simple procedure of having the user look at several dots on the computer screen and can be accomplished in a matter of seconds. Further, the user's calibration can be saved allowing the user to come and go as the user pleases (e.g., on a coffee or restroom break) and return to be recorded by the eye-tracking computer without having to recalibrate.

The use of eye-tracking technology should not be dismissed as a tool that can be used only for experimentation in laboratory settings. Eye-tracking devices have been successfully implemented in several field experiments. In particular, real-time eye-tracking technology has been implemented in cars and trucks to monitor driver fatigue (Ji et al., 2004) and awareness (Donmez, Boyle, & Lee, 2007; Pompei, Sharon, Buckley, & Kemp, 2002), the technology has been examined as a method for control of aircraft (Schnell & Wu, 2000), and the technology has been used to develop tools to facilitate the interaction between disabled people and machine interfaces (De Santis & Iacoviello, 2009).

Details of the Predictive Model

Ratwani et al. (2008) developed three measures to successfully predict postcompletion errors on a computer-based task. One measure examined whether the postcompletion step within the task interface was fixated on or not (i.e., did the user look at the postcompletion step button). This measure is binary and is called the *postcompletion fixation* measure. The second measure was a count of the number of fixations made by the user after completing the step just prior to the postcompletion step; this measure is called the *total fixation* measure. The third measure was *time*[1] from completion of the action prior to the postcompletion step. The postcompletion fixation and total fixation measures were significant predictors in the model; the time measure was not a significant predictor. The logistic regression equation is as follows:

$$\begin{aligned}\text{Predicted logit of Error} = .12 &- (\text{postcompletion fixation} \times 5.7)\\ &+ (\text{total fixations} \times .63) - (\text{time} \times .001)\end{aligned}$$

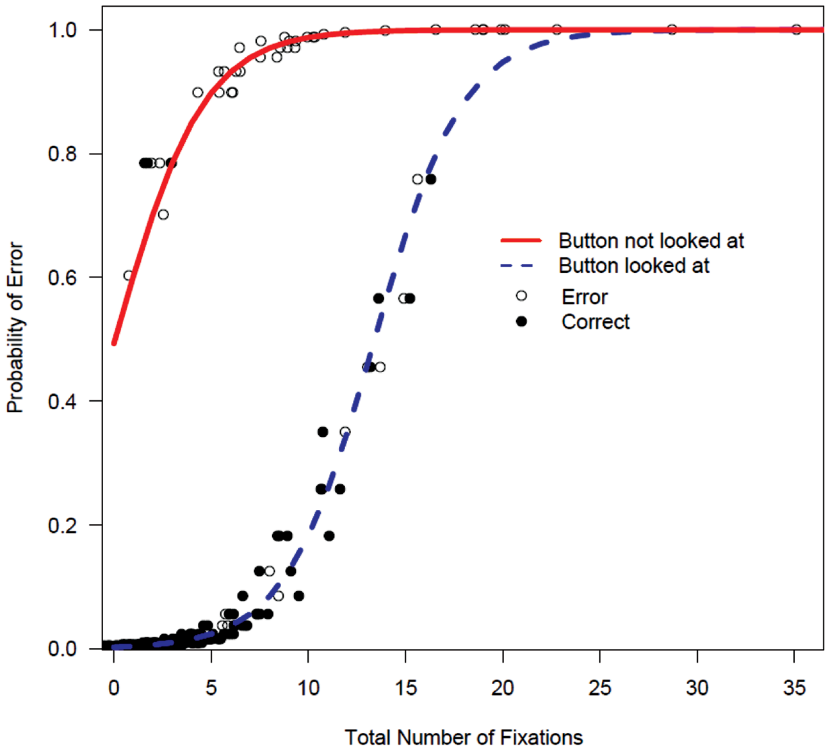
The postcompletion fixation measure instantiates the Byrne and Bovair theory that the reason a postcompletion error is made is because the step itself is forgotten. Clearly, if the step is forgotten, the button or area of the interface associated with the postcompletion step is not likely to be visually examined. The postcompletion fixation measure also instantiates part of the priming constraint from Altmann and Trafton (2002, 2007) which suggests that the environment (in this case, the post-completion button) provides priming to the goal representation. Failing to look at the postcompletion button will not boost activation, leading to a higher likelihood of making an error. Although the process explanations are slightly different for the two models, it is clear that not looking at the postcompletion step increases the probability of making an error.

The total fixation and time measures are derived directly from the memory for goals theory (Altmann & Trafton, 2002, 2007); these measures represent goal decay. The more fixations that occur or the more time that passes, the more the subgoal of the postcompletion step has decayed from memory and the less likely this subgoal will be retrieved to direct behavior. Thus, the more fixations that occur or the more time that passes, the more likely a user will be to make a postcompletion error. Of interest, the total fixations measure was a significant predictor in the model, but the time measure was not. The total fixations measure may also capture individual differences in decay rates and differences in visual and cognitive activity (Just & Carpenter, 1976; Rayner, 1998); thus, this measure may provide greater predictive power than time alone.

Figure 1 illustrates how the predicted probability of a postcompletion error changes when the predictors from the logistic regression model are varied (e.g., how does the probability of error change given a postcompletion button fixation and an increasing number of total fixations). We calculated the actual predicted probabilities by varying the levels of the two significant predictors (postcompletion fixation and total fixations) and then plotted the predicted probabilities; the nonsignificant time predictor is not represented in this figure. The binary postcompletion fixation predictor is represented by two lines in the graph. The dashed line represents cases where a user looks at the postcompletion action button on the task interface, and the solid line represents cases where a user does not look at the postcompletion action button. The continuous total fixations measure is represented on the *x*-axis.

[1] The time measure was originally calculated by Ratwani et al. (2008) as the sum of fixation durations immediately preceding the postcompletion action. The logistic regression model was also run using clock time, and the results were consistent; the time measure was not a significant predictor of postcompletion errors.

FIGURE 1. The logistic regression model developed by Ratwani et al. (2008). (Color figure available online.)



The *y*-axis shows the predicted probability of a user making an error. Finally, the data associated with the error and nonerror cases from the original Ratwani et al. (2008) study were run through the model, and the predicted probabilities for those cases are plotted on the graph. The empty circles represent the error cases, and the filled circles represent the correct postcompletion actions; these circles have been slightly jittered in the *x* direction of the graph to reduce the number of overlapping circles.

As the dashed line in Figure 1 suggests, if a user looks at the postcompletion action button, there is a very low probability that the user will make a postcompletion error, at least until 10 or 15 fixations. Based on the theories just described, fixating on the postcompletion action button suggests that the user is attending to the postcompletion step and that attending to this cue may provide activation to the subgoal. However, as the number of fixations increases (i.e., greater goal decay), the likelihood of a user making the error quickly increases. The majority of participants' actual correct actions (filled circles) fall on or near the dashed line when there are fewer than 10 total fixations (i.e., when participants get activation from looking at the button and there is little goal decay). As the solid line in Figure 1 suggests, if a user does not look at the postcompletion action button, there is a much higher initial chance of the user making an error. The failure to fixate on the postcompletion action button reflects inattention and suggests that the subgoal is not being primed by this environmental cue. As the total number of fixations increases (i.e., greater goal decay), the likelihood of error quickly increases. One can see that the majority of the error actions (unfilled circles) fall on or near the solid line when there are more than five fixations. Overall, correct actions (filled circles) are associated with lower predicted probabilities of error, and error actions (unfilled circles) are associated with higher predicted probabilities of error.

One criticism of the logistic regression model might be that the predictive power of the postcompletion button fixation measure may be explained by the fact that a user must always fixate on the postcompletion action button before clicking the button. After all, most actions on a computer interface require the user to attend to the relevant parts of the interface before interacting with them. Consequently, one might conclude that a single fixation on the postcompletion action button would predict that no postcompletion error will occur. However, the dashed line in Figure 1 which represents when a user looks at the postcompletion button makes the point that a user may look at the postcompletion action button and still make a postcompletion error. There are several instances where this might occur. For example, if the postcompletion goal has undergone a substantial amount of decay, a user might fixate on the postcompletion button and continue searching the interface eventually making a postcompletion error. Thus, the explanation of looking before clicking will not account for all of the postcompletion error cases. The measures of decay (total fixations and time) do, however, contribute to accounting for these cases where a user might look at the postcompletion action button and still make an error.

1.2. The Current Study

Ratwani et al. (2008) suggested that the logistic regression model can be used to predict when a user will make a postcompletion error as the user is working on a particular computer-based task. When the logistic regression model reveals that there is a high likelihood of a postcompletion error, one should intervene and alert the user that an error is imminent before the user actually makes the error. Thus, this logistic regression model can be used to predict when a user will make an error in real time, and then a method can be developed to prevent the user from making the error. To accomplish this goal, the predictive value and robustness of this logistic regression model must first be examined. After determining whether the model can adequately predict postcompletion errors, the model can be implemented in a real-time eye-tracking system to determine whether postcompletion errors can truly be prevented before a user actually makes the error.

Experiment 1 focused on validating the logistic regression model and on establishing an appropriate decision criterion value at which one should intervene based on the predicted likelihood of a postcompletion error. Data were collected on that same task that was used by Ratwani et al. (2008), and the original model was applied to the new data set. Experiment 2 focused on validating the logistic regression model and the determined optimal decision criterion on a new task. In Experiment 3, the logistic regression model was implemented in an eye-tracking system and user's eye movements were monitored in real time to predict and prevent postcompletion errors as the user performed the computer-based procedural task.

2. EXPERIMENT 1

The purpose of the experiment was to answer two main questions. First, what is the appropriate probability value at which a user should be alerted that a postcompletion error is likely? Because logistic regression results in a predicted probability of error, not a binary indicator of error likelihood, an appropriate discrimination criterion of error/nonerror classification must be established. Second, after establishing the optimal discrimination criterion, how well does the logistic regression model categorize instances of errors and nonerrors on a new data set that is different from the data set upon which the model was based? To be useful, the determination to provide an alert should be extremely accurate, providing an alert only when an error is extremely likely to occur, neither so often that it becomes irritating nor so rarely that it does not prevent errors.

To answer these questions, a receiver-operating characteristic (ROC) analysis from signal detection theory was used (see Appendix A). ROC analysis is a method for selecting the optimum criterion when making a positive or negative decision about the occurrence of an event (Swets, 1986a, 1986b, 1992). An ROC curve illustrates how the true-positive (hits) and false-positive (false alarms) rates vary as the decision criterion level is varied. A clear example of how an ROC analysis is useful comes from Swets (1992). Swets gave an example of an inspector examining metal aircraft parts to determine whether the metal is fatigued; if the metal is fatigued, the aircraft part cannot be used. An inspector may conduct a diagnostic test of the part, which results in a meter reading; based on this meter reading the inspector must decide to use the part or not. The meter reading lies along a scale, and the decision to accept or reject the aircraft part requires a decision criterion on the scale (e.g., a point on the scale at which values above the criterion will be identified as fatigued and values below will be identified as nonfatigued).

Using an ROC analysis, the number of aircraft parts that are correctly and incorrectly identified as fatigued can be examined given different meter reading values. Hypothetically, the meter readings range from 0 to 100 and the optimal decision criterion is 55 (i.e., this decision criterion maximizes the true positives and minimizes the false negatives). Thus, when the meter reading is 55 or greater, the inspector should classify the aircraft part as fatigued, and when the reading is below 55 the inspector should classify the part as nonfatigued. Looking at an ROC curve, one could determine the true-positive and false-positive rates at the optimal decision criterion and could see how these rates change as the criterion is changed. Further, the outcomes can be weighted (e.g., it may be important to avoid false negatives) and the decision criterion can be adjusted based on these weightings.

Just as the inspector's meter reading lies on a scale and the decision to accept or reject a part requires a decision criterion, our predictive model results in the probability of making a postcompletion error (ranging 0–100) and the decision to categorize actions as an error or nonerror requires a decision criterion. To determine the optimal decision criterion, participants' eye movement data associated with postcompletion error and nonerror actions were entered into the logistic regression model and the predicted probabilities of a postcompletion error were determined. The predicted probabilities were then compared to the actual occurrence of an error. The ROC curve was calculated by varying the probability values (i.e., the decision criterion) of error classification from 0% to 100% and plotting the associated true positives and false positives (Fawcett, 2006; Macmillan & Creelman, 2005). Based on this ROC curve, the optimal criterion value could be selected by finding the criterion point on the ROC curve that is associated with the highest true-positive rate and lowest false-positive rate. For the purposes of testing the postcompletion error model, the consequences of the predicted outcomes (e.g., true positive or false positives) were equally weighted. Finally, the number of postcompletion error and nonerror actions that were correctly classified at the optimal discrimination criterion was examined to determine the robustness of the logistic regression model.

2.1. Method

Participants

Thirty George Mason University students participated for course credit.

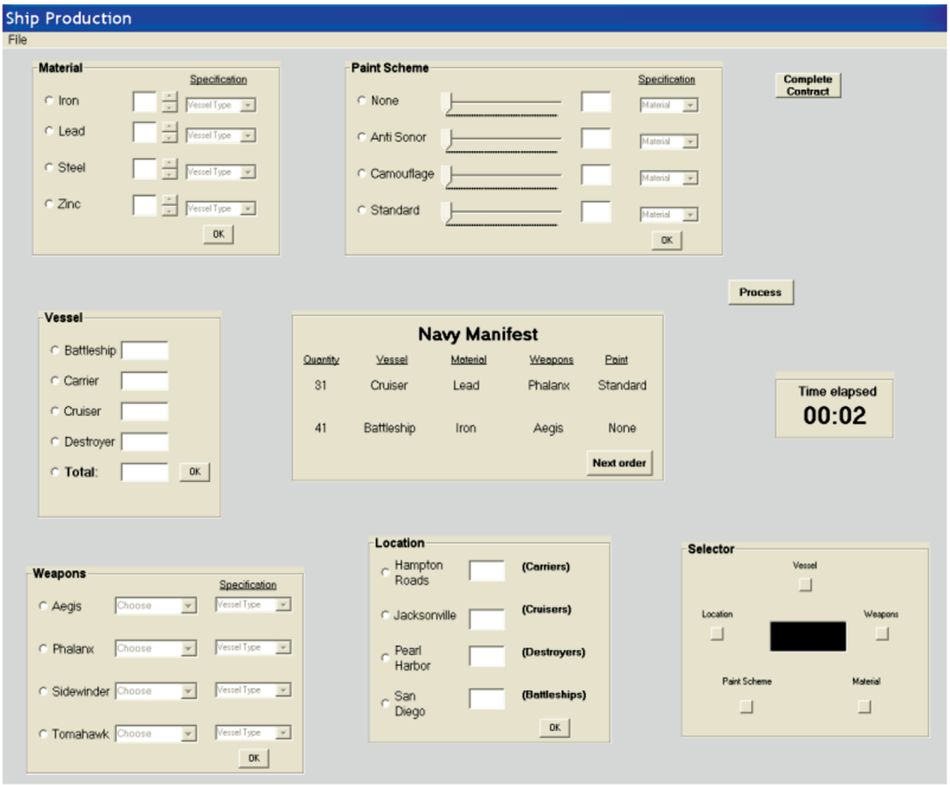
Task and Materials

The same experimental paradigm that was used by Ratwani et al. (2008) was used in this experiment as well. This paradigm consisted of a procedural task that served as the primary task and periodic interruptions from a secondary task to increase the error rates. This allowed for a large-enough data set to study the postcompletion errors.

The primary task was a complex production task called the *sea vessel* task (originally based on Li, Blandford, Cairns, & Young, 2008). The objective of the task is to successfully fill an order for two different types of sea vessels by entering order details in various modules on the interface (see Figure 2). The order of entering information is as follows: vessel information, material, paint, weapons, location. Before entering information in a module, the module must first be activated by clicking the appropriate button in the selector window (lower right corner of Figure 2). After entering information in the module the *ok* button was clicked to submit the information.

After entering information in each of the modules, the order was processed by clicking the *process* button. Upon clicking the *process* button, a small window popped up indicating the order had been submitted, and the window provided details about how many sea vessels were ordered. The participant had to click the *ok* button to acknowledge this window. Finally, the participant must click the *complete contract*

FIGURE 2. Screenshot of the sea vessel production task. (Color figure available online.)



Navy Manifest

Quantity	Vessel	Material	Weapons	Paint
31	Cruiser	Lead	Phalanx	Standard
41	Battleship	Iron	Aegis	None

button to finish the order. Clicking the *complete contract* button is the postcompletion step. Appendix B contains a complete description of the primary task and a diagram illustrating the structure of the task.

The pop-up window that appeared after clicking the *process* button served as a false completion signal, which is characteristic of postcompletion steps (Byrne & Bovair, 1997; Li et al., 2008; Li, Cox, Blandford, Cairns, & Abeles, 2006). Failure to click the *complete contract* button after acknowledging the signal constituted a postcompletion error.

Any deviation from the strict sequence of actions (vessel, material, paint, weapons, location, process, complete contract) was considered an error. If a participant made an error on the primary task, the computer emitted a beep signifying the error. The participant had to make the correct action in order to continue on with the task; thus, if a postcompletion error was made, the participant had to continue working and eventually make the correct postcompletion action to move on.

After entering information in any of the modules and clicking the *ok* button, the information that was entered in the module was cleared from the interface. Thus, no information remained on the interface to indicate which steps have been completed. Clearing the information on the interface was purposefully done to remove global place keeping (Gray, 2000) and to force participants to remember which steps have already been completed without explicit environmental cues.

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. 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 six predefined interruption points in the sea vessel task. There was an interruption point after clicking the *ok* button in each of the five modules. The sixth interruption point occurred after the *process* button was clicked, and the false completion signal was acknowledged by clicking the *ok* button; this interruption point occurred just prior to the postcompletion step. During the experiment, there were 12 interruptions (6 interruption trials × 2 interruptions in each trial) each lasting 15 s. The interruptions were equally distributed among the six interruption locations.

In total, there were 12 postcompletion error opportunities (one on 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 47 cm from the computer monitor. After the experimenter explained the sea vessel 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 or 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 sea vessel task, there was no information available on the interface to indicate where to resume.

Measures

Keystroke and mouse data were collected for every participant. A postcompletion error was defined as failing to click the *complete contract* button and instead making an action that was in service of the next order on the sea vessel task (e.g., clicking *Next Order*). Repeating a step that was already performed was not considered a postcompletion error. Postcompletion error rates were calculated by counting any postcompletion error that occurred during control or interruption trials. The postcompletion error rate was the number of errors divided by the number of opportunities to make a postcompletion error.

Eye track data were collected using a Tobii 1750 operating at 60 Hz. 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. For the purposes of this study we were concerned only with postcompletion errors; thus, all eye movement analyses focus on the postcompletion steps. Based on the eye movement data, the postcompletion fixation measure, the total fixation measure, and the time measure, as used by Ratwani et al. (2008), were calculated. In the control trials, the recording of the three measures began with the acknowledgment of the false completion signal (i.e., clicking the *ok* button) and ended with the correct action of clicking the *complete contract* button or an error action. In the interruption trials in which the interruption immediately preceded the postcompletion step, the recording of these measures began with the offset of the interruption and ended with the correct action of clicking the *complete contract* button or an error action. In the interruption trials in which the interruption did not immediately precede the postcompletion step, the recording of the measures was the same as the control trials.

The postcompletion fixation measure was a dichotomous variable that indicated whether the postcompletion action button (i.e., the *complete contract* button) was fixated on. The total fixation and time measures were continuous variables. Consistent with Ratwani et al. (2008), time was measured as the sum of fixation durations (in milliseconds). To determine whether participants fixated on the *complete contract* button, this button was defined as an area of interest. The width of the button subtended 3.5° of visual angle and the height of the button subtended 1.5° of visual angle. The button was separated by at least 3° of visual angle from any other relevant part of the task interface.

Despite the nonsignificant loading of the time predictor in the original logistic regression model, this predictor was included in the ROC analysis in all experiments. There were theoretical reasons to initially include the time predictor, and although it was not significant in the original formulation of the model this predictor may be important in other tasks.

2.2. Results and Discussion

Postcompletion Error Rates

Fourteen participants made at least one postcompletion error. There were 18 postcompletion errors in the data set; the average postcompletion error rate across all participants and all trials was 5%. Participants made significantly more postcompletion errors in interruption trials ($M = 9.4\%$) than control trials ($M = .6\%$), $F(1, 29) =$

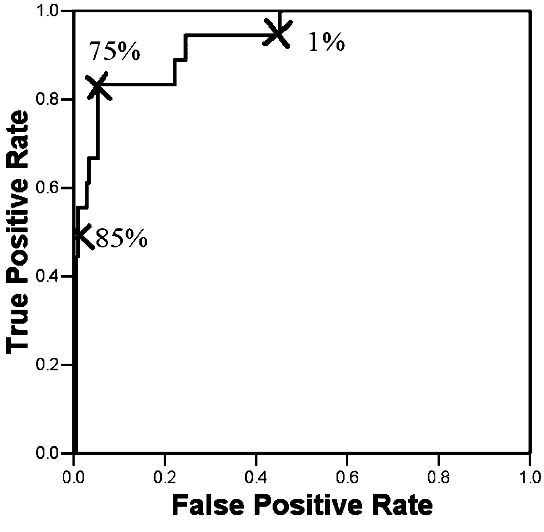
14.2, $MSE = 83.7$, $p < .001$. The extremely low percentage of postcompletion errors in the control trials suggests that participants understood that the postcompletion step was part of the task. Participants were accurate on the interruption task itself, attempting an average of 1.5 addition problems per interruption with an average accuracy of 85.6%.

ROC Analysis

For each postcompletion step, regardless of whether a postcompletion error was committed or not, the eye movement data were used to formulate the postcompletion fixation measure and the total fixation measure. These two measures, in addition to the time measure, were entered in the logistic regression equation (see Equation 1), resulting in a predicted probability of error for each case. All analyses were performed off-line after all of the participant data had been collected. To determine the optimal decision criterion for the logistic regression model, probability values were systematically varied from 0 to 100% (Fawcett, 2006), and the predicted errors and nonerrors at each criterion level were compared to the actual data to generate the true-positive and false-positive rates. For example, if the decision criterion value is set at 10%, cases with predicted probabilities greater than or equal to 10% would be categorized as error cases and cases with predicted probabilities less than 10% would be categorized as nonerror cases.

The pairs of true-positive and false-positive values for each decision criterion point (0–100) were plotted generating the ROC curve seen in Figure 3. In ROC curve space, the upper left-hand corner—that is, point (0,1)—represents perfect prediction: all true positives and no false positives. Thus, the optimal decision criterion for the

FIGURE 3. Receiver-operating characteristic analysis (ROC) curve from Experiment 1.



logistic regression model that will maximize true positives and minimize false positives will be the point on the ROC curve that is closest to the upper left-hand corner.

By visually examining the ROC curve in Figure 3, one can see that the decision criterion value of 75% generates a point on the ROC curve that is closest to the upper left-hand corner of the graph. Note that the decision criterion value itself does not provide information about the true-positive and false-positive rates; rather, these rates are determined by looking at the values on the *x*- and *y*-axis associated with the 75% decision criterion data point on the ROC curve. Looking at the curve, the decision criterion maintains a high true-positive rate (.83) while maintaining a low false-positive rate (.05). The 75% decision criterion value has the maximum sensitivity of all points on the curve with $d' = 2.6$. Thus, a 75% decision criterion value maximizes true positives (benefits) while minimizing false positives (costs).

Figure 4 shows the confusion matrix at the 75% decision criterion; this matrix displays the number of error and nonerror actions from the data set that were correctly and incorrectly classified. Fifteen of the 18 (83.3%) postcompletion errors were correctly classified, and 197 of 208 (94.7%) nonerrors in the data set were correctly classified. Thus, the 75% decision criterion resulted in a false-positive rate of only 5.3%.

To determine how robust the logistic regression model is at predicting postcompletion errors on this data set, the area under the ROC curve (AUC) can be examined. The area under the curve represents the probability that the logistic regression model will rank a randomly chosen positive instance (i.e., an error) higher than a randomly chosen negative instance (i.e., nonerror; Fawcett, 2006; Macmillan & Creelman, 2005). The area under the ROC curve for this particular data set is equal to .93. This is considered excellent and suggests that the logistic regression model is quite robust at classifying errors and nonerrors.

At a high level, this ROC curve analysis demonstrates several critical points as they pertain to postcompletion error prediction. First, the analysis suggests that a decision criterion of 75% should be used as the point at which the user should be notified that a postcompletion error is imminent. This decision criterion has the greatest sensitivity in that it maximizes the number of true positives and minimizes the number of false positives. Second, the ROC analysis demonstrates that the postcompletion error prediction equation is quite robust in that the area under

FIGURE 4. Confusion matrix with a 75% decision criterion applied to the sea vessel task.

	Actual Value	
Predicted Value	True positive 15 (83.3%)	False positive 11 (5.3%)
	False negative 3 (16.7%)	True negative 197 (94.7%)

Note. The actual number of error and nonerror actions are displayed to the left of the percentages.

the curve value was extremely high, suggesting that in most instances the logistic regression model is assigning a higher probability value to errors than to nonerrors.

3. EXPERIMENT 2

Experiment 1 successfully validated the logistic regression model on the task upon which the model was originally based. The purpose of Experiment 2 was to determine whether the logistic regression model is task specific or whether it can be extended beyond the task upon which it was originally based. Can the model account for postcompletion errors on a new computer-based task? There are some unique features of the sea vessel task used in Experiment 1 that might bring the generalizibility of the predictive model into question. In fact, the interface of the sea vessel task was purposefully made nonoptimal to increase error rates (Li et al., 2008). However, because the specific predictors in the model are theoretically based, the model should not be task specific. The general set of predictors should adequately account for postcompletion errors, regardless of task-specific details.

Our goal in this experiment was to take the existing logistic regression model and apply the model to data from a new task. Ideally, the same decision criterion, theoretic variables, and weights would apply across tasks, though it would not be surprising if the weights changed as the task changed. We thus collected data on a new task that had a postcompletion step to evaluate the decision criterion, variables, and weights. We again collected eye movement and RT data and performed an ROC analysis.

3.1. Method

Participants

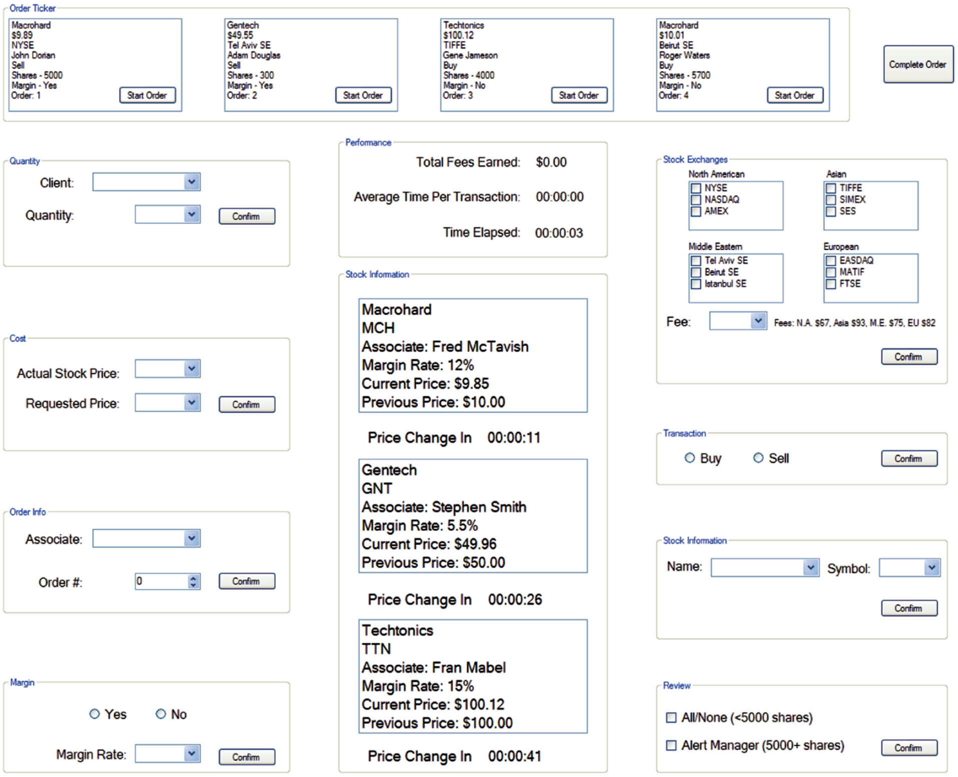
A new sample of 36 George Mason University undergraduate students, different from the previous experiment, participated for course credit.

Task and Materials

Participants performed a computer-based procedural task as the primary task and, similar to Experiment 1, participants were periodically interrupted by a secondary task. The primary task was a complex financial management task. The goal of the task was to successfully fill a client's orders for different types of stocks. The orders were to either buy or sell and were presented four at a time at the top of the screen (see Figure 5). The current prices of the stocks associated with the orders were presented in the center of the screen in the Stock Information column. The actual stock prices fluctuated every 45 s.

To complete an order, participants first had to determine which of the client orders could currently be executed by comparing the client's requested price to the

FIGURE 5. Screenshot of the financial management task. (Color figure available online.)



Order Ticker

Macrohard
$9.89
NYSE
John Dorian
Sell
Shares - 5000
Margin - Yes
Order: 1

Gentech
$49.55
Tel Aviv SE
Adam Douglas
Sell
Shares - 300
Margin - Yes
Order: 2

Techtonics
$100.12
TIFFE
Gene Jameson
Buy
Shares - 4000
Margin - No
Order: 3

Macrohard
$10.01
Beirut SE
Roger Waters
Buy
Shares - 5700
Margin - No
Order: 4

Performance

Total Fees Earned: $0.00
Average Time Per Transaction: 00:00:00
Time Elapsed: 00:00:03

Stock Information

Macrohard
MCH
Associate: Fred McTavish
Margin Rate: 12%
Current Price: $9.85
Previous Price: $10.00

Price Change In 00:00:11

Gentech
GNT
Associate: Stephen Smith
Margin Rate: 5.5%
Current Price: $49.96
Previous Price: $50.00

Price Change In 00:00:26

Techtonics
TTN
Associate: Fran Mabel
Margin Rate: 15%
Current Price: $100.12
Previous Price: $100.00

Price Change In 00:00:41

Order Info

Associate:
Order #: 0

Stock Exchanges

North American: □ NYSE □ NASDAQ □ AMEX
Asian: □ TIFFE □ SIMEX □ SES
Middle Eastern: □ Tel Aviv SE □ Beirut SE □ Istanbul SE
European: □ EASDAQ □ MATIF □ FTSE

Fee:
Fees: N.A. $67, Asia $93, M.E. $75, EU $82

Transaction

○ Buy ○ Sell

Review

□ All/None (<5000 shares)
□ Alert Manager (5000+ shares)

actual market price of the stock from the Stock Information column. If the client's order is to buy a stock, the current stock price must be equal to or less than the requested buy price for the order to be executable. If the client's order is to sell a stock, the current stock price must be equal to or greater than the requested sell price for the order to be executable. Once an order was determined to be executable, the participant clicked the *Start Order* button for the respective order. To actually fill the order, the participant had to enter details from the order itself and the Stock Information column in to eight different modules on the screen. Participants had to follow a specific procedure to complete the order; the specific sequence was Quantity, Cost, Order Info, Margin, Stock Exchanges, Transaction, Stock Info, and Review. Overall, the financial management task has a considerably better interface compared to the sea vessel task. In particular, the spatial layout of the interface (working from top to bottom down the left column and then the right column of Figure 5) and the operations required to perform the task are quite intuitive. Appendix C contains a full description of the primary task and a diagram of the task hierarchy.

After entering information in each module, the participant clicked the *Confirm* button and could then move on to the next module. After clicking confirm on the final module (the Review module), a pop-up window appeared confirming the details of the order. The participant then had to acknowledge the window by clicking *Ok*. Finally, to complete the order 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 postcompletion error.

Similar to Experiment 1, if a participant deviated from the strict procedure, the computer emitted a beep signifying that an error had been made and the participant had to continue working until the correct action was completed. Thus, if a postcompletion error occurred, the participant had to continue working and make a correct action to terminate the trial. No information remained on the interface after clicking the *confirm* button (i.e., no global place keeping; Gray, 2000).

The interrupting task was the same as Experiment 1 with the exception that participants were presented with five possible solutions (four incorrect, one correct) and had to click on the button associated with the correct solution.

Design and Procedure

The completion of one order on the financial management task constituted a trial. The design of control and interruption trials was the same as Experiment 1, except that there were eight possible interruption points in the financial management task. These points occurred after clicking the *Confirm* button following the first seven modules and after acknowledging the false completion signal, 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 number of postcompletion error opportunities during the interruption and control trials was the same as Experiment 1.

The interruption duration and the procedure were the same as Experiment 1.

Measures

Data collection, the definition of a postcompletion error, and the calculation of error rates were the same as Experiment 1.

The postcompletion fixation measure, the total fixation measure, and the time measure, as used by Ratwani et al. (2008) were calculated in the same way as Experiment 1. The width of the *complete contract* button subtended 2.4° of visual angle and the height subtended 1.5° of visual angle. The button was separated by at least 2.5° of visual angle from any other relevant part of the task interface.

3.2. Results and Discussion

Postcompletion Error Rates

Nineteen participants made at least one postcompletion error. There were 27 postcompletion errors in the data set; the average postcompletion error rate across all participants and all trials was 6.3%. Participants made significantly more postcompletion errors in interruption trials ($M = 10.6\%$) as compared to control trials ($M = 1.4\%$), $F(1, 35) = 16.9$, $MSE = 90.9$, $p < .001$. The extremely low percentage of postcompletion errors in the control trials suggests that participants understood that the postcompletion step was part of the task. Participants were accurate on the interruption task itself, attempting an average of 2.1 addition problems per interruption with an average accuracy of 87.6%.

ROC Analysis

For each postcompletion step, regardless of whether a postcompletion error was committed or not, the eye-movement data were used to formulate the postcompletion fixation measure and the total fixation measure. These two measures, in addition to the time measure, were entered in the logistic regression equation (see Equation 1), resulting in a predicted probability of error for each case. All analyses were performed off-line after all of the participant data had been collected.

To quantitatively determine the overall robustness of the model in accounting for postcompletion errors and nonerrors an ROC analysis was performed, similar to Experiment 1. Specifically, the statistic of interest was the area under the ROC curve, which is a measure of the overall robustness of the logistic regression model in classifying error and nonerrors cases. The area under the ROC curve for the logistic regression model applied to this particular data set is .96 with $d' = 2.1$. The AUC value is excellent, and this analysis suggests that the logistic regression model is appropriately assigning a higher predicted probability of error to a randomly chosen true error case compared to a randomly chosen nonerror case the majority of the time.

Next, the performance of the logistic regression model with the 75% decision criterion was examined. If the predicted probability of error was 75% or greater for a particular case, the case was classified as an error. If the predicted probability was less than 75%, the case was classified as a nonerror. The predicted error and nonerror cases were then compared to the actual data to determine the accuracy of the model. Figure 6 shows the confusion matrix for the data set given the 75% decision criterion. The model accurately predicted 25 of the 27 postcompletion

FIGURE 6. Confusion matrix given the 75% decision criterion applied to the financial management task.

	Actual Value	
Predicted Value	True positive 25 (92.6%)	False positive 37 (10%)
	False negative 2 (7.4%)	True negative 328 (90%)

Note. The actual number of error and nonerror actions are displayed to the left of the percentages.

errors (92.6%) and incorrectly classified 10% of the true nonerrors as errors. Overall, the confusion matrix demonstrates that the model performed extremely well, accounting for the majority of postcompletion errors while keeping the false-positive rate low.

The ROC analysis and the confusion matrix demonstrate three important points. First, these analyses confirm the accuracy of the 75% decision criterion that was established in Experiment 1. Using this decision criterion, the model performed extremely well even when applied to a new task; more than 92% of the postcompletion errors were correctly classified, and 90% of the nonerror actions were correctly classified. Second, the area under the curve statistic confirms the accuracy of the variables in the predictive model. The high AUC value shows that the model is not task specific and provides strong support for the theoretical mechanisms underlying the predictors. Finally, the successful validation of the model demonstrates the accuracy of the weights of the predictors and suggests that the model taken as is can accurately classify postcompletion errors on computer-based tasks that are different from the task upon which the model was originally established.

In both Experiments 1 and 2, the original logistic regression model from Ratwani et al. (2008) was applied to a new data set to determine how well the model predicted postcompletion errors in the new data set. An alternative approach is to develop a unique logistic regression model from each data set and then to compare the unique logistic regression models to the original model developed by Ratwani et al. The model comparison approach has been used to examine similarities and differences between the original predictive model from Ratwani et al. and unique models based on the financial management task from Experiment 2 (see Ratwani & Trafton, 2010a). The results of the model comparison approach demonstrate a huge amount of convergence between the different models, and the results are in agreement with the results from the experiments presented here. In this article, we chose to apply the original model to new data sets because the goal of this article is to demonstrate how the original model can be applied to different tasks to successfully predict postcompletion errors without having to take the time and effort to develop a unique model for each task. The approach that has been used here emphasizes the applicability of the model to different tasks, whereas the model comparison approach emphasizes theoretical development. Some of the results of the model comparison approach are discussed in more detail in the general discussion.

4. EXPERIMENT 3

The analyses from Experiments 1 and 2 suggest that the logistic regression model can accurately predict postcompletion error and nonerror actions. The experiments, however, do not speak to the feasibility of implementing this model in an eye-tracking system to predict and prevent postcompletion errors in real time, before a user actually makes a postcompletion error. The goal of this experiment was to determine whether the logistic regression model could be integrated with an eye-tracking system to analyze users' eye movements as a procedural task is being performed in order to predict and prevent postcompletion errors before they occur. The real-time postcompletion error prediction system should detect when a user has a high probability of making a postcompletion error, and the system should notify the user that an error action is imminent before the user actually commits the postcompletion error.

There are two important components to a real-time eye-tracking system to prevent postcompletion errors. First, a user's raw behavioral measures preceding the postcompletion step must be collected and processed such that the data can be entered in to the logistic regression model resulting in a predicted probability of error. The data collection and processing must be done in real time as the user is performing the task at hand. If there is a lag in the processing of the data, the real-time prediction system is rendered useless because the user would have already made a postcompletion error before the system detected the error. Thus, as the user is interacting with the task there must be a near simultaneous online calculation of the predicted probability of error. This process of analyzing eye-movement data in real time is different from the previous experiments and different from Ratwani et al. (2008); those experiments processed and analyzed the data well after the participant performed the task.

The second component of the real-time system is the method in which the user is notified that an error may be imminent. When the predicted probability of error is calculated to be 75% or greater, a notification must be delivered to the user in a timely fashion such that the user immediately processes the notification before the user commits the error. Previous research has suggested that in order for a cue to be effective, it should be salient and blatantly obvious to the user (Chung & Byrne, 2008; Trafton et al., 2005). Thus, the prevention method that has been implemented in the real-time error prediction system is a large red arrow pointing to the next correct action. The red arrow cue is presented on the screen as soon as the 75% decision criterion is reached with the intent of notifying the user of the next correct action before the user makes an error.

To determine whether the real-time error prediction system reduces the number of postcompletion errors, the user's performance with the error prevention system was compared to a baseline condition that did not have the real-time error prediction system. If the error prediction and prevention system can adequately predict when users will make a postcompletion error and the arrow cue can prevent the user from making the error, the postcompletion error rate in the prediction and prevention system condition should be drastically lower than the baseline condition.

4.1. Method

Participants

A new sample of 30 George Mason University undergraduate students, different from the previous experiments, participated for course credit.

Task and Materials

The primary and secondary tasks were the same as Experiment 1, and the same eye tracker was used.

Design and Procedure

Participants were randomly assigned to either a baseline condition or a real-time prediction and prevention condition; there were 15 participants in each condition. The design and procedure of the baseline condition was the same as Experiment 1. Participants performed the sea vessel production task with no real-time eye analyses and no explicit visual cues.

The design of the real-time prediction and prevention condition was the same as Experiment 1, with the exception that the eye-movement data were analyzed in real time using the logistic regression model from Ratwani et al. (2008). To simplify the real-time processing of eye-movement data, only the two significant predictors from the model were used (postcompletion fixation and total fixations); these predictors were calculated and continually updated during the postcompletion step as the participant performed the task. To formulate the total fixation and postcompletion fixation measures, participant's eye movements were first parsed into fixations; fixations were defined in the same way as Experiment 1. The total fixation measure consisted of an online count of these fixations. Once a fixation was defined, the postcompletion fixation measure was determined by comparing the location of the fixation to the location of the *complete contract* button to determine whether the fixation landed on the *complete contract* button or not.

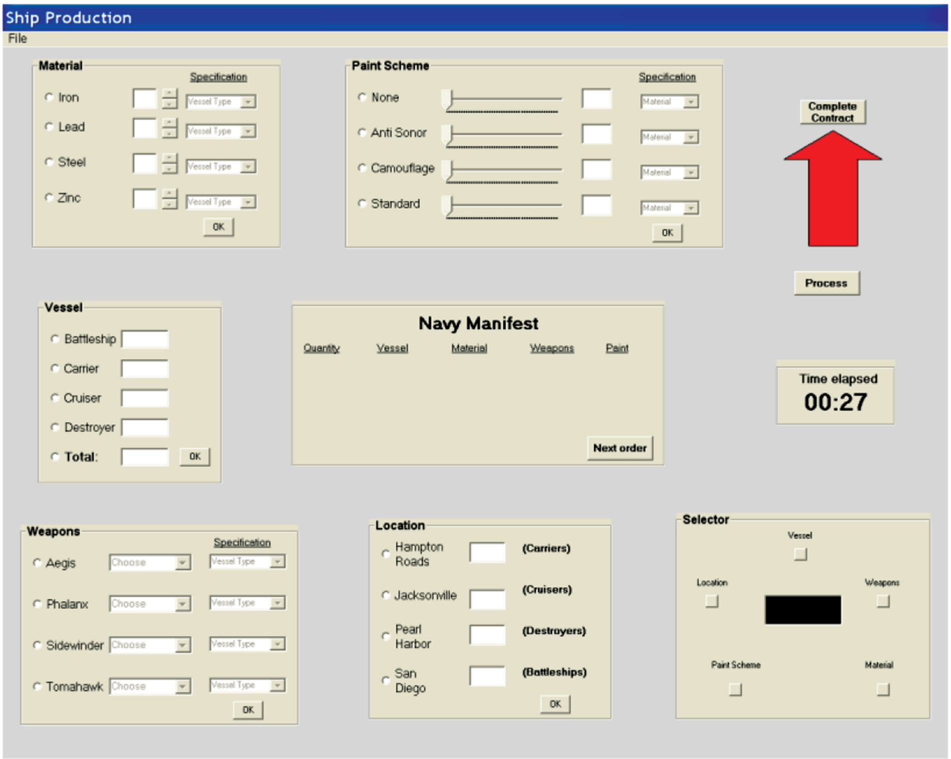
After a fixation was detected, the two measures were entered into the logistic regression equation and a predicted probability was immediately calculated. If the predicted probability was greater than or equal to 75%, a large red arrow immediately appeared on the screen pointing to the postcompletion action (see Figure 7). If the predicted probability was less than 75%, no cue appeared. The periods of measurement in which real-time predicted probabilities were calculated in the control and interruption trials was the same as Experiment 1.

Participants in the prediction and prevention condition received the same training and instructions as participants in the baseline condition. In addition, participants in the prediction and prevention condition were told that the computer system may present a red arrow cue pointing to the action that the computer thinks is the next correct action when the system detects that the user is likely to make an error. Participants were told that the system may present the arrow at any step in the task, not just the postcompletion step. Further, participants were instructed that they could either use or ignore a cue if a cue appeared.

4.2. Results and Discussion

In the baseline condition, 11 participants made at least one postcompletion error; there were 17 postcompletion errors. The average postcompletion error rate

FIGURE 7. Screenshot of the sea vessel production task with the arrow cue on the postcompletion step. (Color figure available online.)

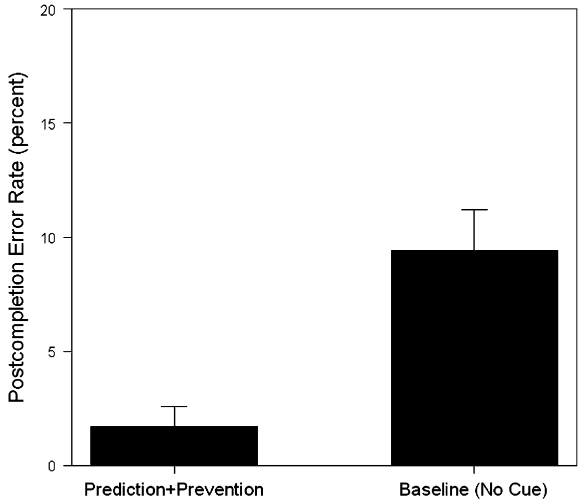


across all participants and all trials in the baseline condition was 9.4%. Participants made significantly more postcompletion errors in interruption trials (*M* = 17.8%) as compared to control trials (*M* = 1.1%), *F*(1, 14) = 15, *MSE* = 138.9, *p* < .01.

In the prediction and prevention condition, three participants made one postcompletion error each for a total of three postcompletion errors. The average postcompletion error rate across all participants and all trials in the prediction and prevention condition was 1.7%. There was no statistically reliable difference between the number of postcompletion errors on interruption trials (*M* = 3.3%) as compared to control trials (*M* = 0%), *F*(1, 14) = 3.5, *MSE* = 23.8, *p* = .08.

To determine whether the real-time postcompletion error prediction system was effective, the postcompletion error rates in the prediction and prevention condition were compared to the baseline condition. As Figure 8 suggests, participants in the prediction and prevention condition made significantly fewer postcompletion errors than participants in the baseline condition, *F*(1, 28) = 15.1, *MSE* = 30.1, *p* < .001. The real-time error prediction system reduced the error rate from 9.4% in the baseline condition to 1.7% in the real-time prediction and prevention condition; the real-time eye-tracking system was highly effective at reducing the occurrence of postcompletion errors.

FIGURE 8. Postcompletion error rates by condition.



Note. **Error bars represent standard error.**

In the prediction and prevention condition, the real-time cue was deployed 13.9% of the time. Thus, the low postcompletion error rate in the prediction and prevention condition is not due to the cue firing on every single postcompletion step. Further, there was no statistically reliable difference between the frequency at which the cue was deployed in the prediction and prevention condition and the 9.4% postcompletion error rate in the baseline condition, $F(1, 28) = 1.6$, $MSE = 148.1$, $p = .2$. Of the three postcompletion errors that were committed in the prediction and prevention condition, two errors were missed by the real-time system (i.e., the cue did not deploy), and in one error instance the cue deployed in time but the participant still made an error.

There was no statistically reliable difference in accuracy on the interrupting task in the prediction and prevention condition ($M = 79.6\%$) and the baseline condition ($M = 79.9\%$), $F(1, 28) = .002$, $MSE = 319.1$, $p = .9$. There was also no statistically reliable difference in the number of addition problems attempted per interruption in the prediction and prevention condition ($M = 1.6$) and the baseline condition ($M = 1.8$), $F(1, 28) = 1.2$, $MSE = .36$, $p = .3$. These results suggest that participants were taking the interrupting task seriously in both conditions.

Overall, these results demonstrate that the logistic regression model can be integrated with an eye tracker to monitor a user's eye movements as he or she performs a procedural task, successfully predict when the user is likely to make a postcompletion error, and notify the user of the error before the error is committed. This real-time error system successfully predicted when errors would be made, and the red arrow cue served to successfully prevent the postcompletion errors, as indicated by the lower postcompletion error rate in the real-time cue condition.

5. GENERAL DISCUSSION

The three experiments presented here demonstrate that the predictive logistic regression model of postcompletion errors developed by Ratwani et al. (2008) can be successfully instantiated in an eye-tracking system to predict and prevent postcompletion errors before they occur. Experiment 1 served to establish an optimal decision criterion of intervention assuming equal weighting of the predicted outcomes. Experiment 2 validated the model on a task different from the task upon which the model was originally based, suggesting that the model is not task specific and may be generalizable to a larger set of computer-based tasks. Instantiating the predictive model in a real-time eye-tracking system in Experiment 3 resulted in a substantial and significant reduction in the postcompletion error rate compared to a baseline condition without the real-time system (from more than 9% to less than 2%).

One of the major contributions of this work is that it illustrates that there are behavioral differences, measured by eye movements in this case, preceding error, and non-error actions and that these differences can be quantified and leveraged to predict a user's actions before the user makes the action. The use of eye movements in our predictive and preventive model is drastically different from previous research that has used eye movements to assess cognitive state (Marshall, 2007). First, our approach measures behavior to predict a future outcome, whereas the research assessing cognitive state is measuring behavior to determine the current cognitive state of the user and is not predictive. Second, our approach functions on a very short time scale (generally .5 s to 2 s) to predict behavior on an action to action basis. Using eye-movement measures to evaluate the cognitive state of a user (e.g., fatigue, boredom, etc.) relies on a longer time horizon and the transition between cognitive states is much more gradual than the time between actions on a procedural task.

The time scale of prediction and prevention granted by eye-movement measures, and the predictive model presented here is starkly different from the EEG and fMRI approach, which correlates preceding brain activity and errors (Eichele et al., 2008). The temporal resolution of the EEG and fMRI techniques, as well as the data-processing time, may be limiting factors in developing a predictive system from these measures, at least in the near future. Further, measuring brain activity is far more intrusive and constraining than the eye-tracking system presented here. Currently, the research correlating brain activity and errors is confined to simple highly repetitive tasks (e.g., the flanker task). The brain activity measures have not been demonstrated on more complex tasks like the ones used in the current study.

Note that one of the assumptions of logistic regression is independence of data, which we are clearly violating in our predictive model (e.g., a participant contributes multiple data points to the data set). There are several reasons to believe that our predictive model has some validity, even while violating this assumption. First, when comparing unique models developed from different data sets and different people, there is a large amount of similarity between the models (Ratwani et al., 2008; Ratwani

& Trafton, 2010a), suggesting that in this case the violation is not critical. Second, as seen in the experiments presented here, the original model has been validated on two different data sets and the original model was used to prevent errors in real time. Thus, the original model has been successfully used on at least four different data sets. These results suggest that the data independence violation is not critical to the success of the model.

5.1. Theoretical Implications

The success of the predictive model in accounting for postcompletion errors across different tasks provides additional support for the components of the Byrne and Bovair (1997) and the Altmann and Trafton (2002, 2007) theories. The predictive model demonstrates that inattention to the postcompletion step, attending to appropriate environmental cues, and goal decay are important components in accounting for postcompletion errors. Although the cognitive theories do not make explicit predictions about which of these predictors are more important in accounting for postcompletion errors, in other work (Ratwani & Trafton, 2010a) we have examined the relative contribution of each of the predictors. These analyses have demonstrated that fixation on the postcompletion action button contributes the most to the predictive power of the model. Thus, goal forgetting and cue association are critical mechanisms underlying postcompletion errors.

The contributions of the total fixation and time predictors, although still important to the overall power of the model, do not contribute as strongly to the overall model as the postcompletion fixation predictor does. When comparing the significance and weights of the total fixation and time predictors across tasks, there is also more variability in these predictors. The total fixations predictor has consistently been a significant predictor (Ratwani et al., 2008; Ratwani & Trafton, 2010a). The time measure did not load significantly in the model from Ratwani et al. (2008), which was based on the sea vessel task, but it was a significant predictor when a unique logistic regression model was developed based on the financial management task (Ratwani & Trafton, 2009).

The variability associated with the time and total fixations predictors may be attributed to individual differences. The time measure is a global measure of decay that essentially represents equal decay rates across all users. Fixation durations, however, are different across individuals and differ depending on the visual item being processed. Thus, in a fixed duration of time, different users may have different total fixation counts. These fixation counts may capture individual differences in the decay rate of memory items as well as individual differences in processing time. It would be considerably easier to use time to represent goal decay as opposed to having to monitor eye movements, but because fixations may be capturing individual differences, the total fixation predictor may be more powerful than the time predictor. Because of the variability in both of these predictors, it is important to include both predictors in the model.

5.2. Generalizability of the Predictive Model

The ROC analyses comparing the performance of the logistic regression model across two different tasks suggests that the model is robust across tasks. However, to generate a large-enough data set of errors to develop and test the logistic regression model, given the generally low base rate of procedural errors, manipulations were used to increase the postcompletion error rate. Specifically, no environmental cues were present on the interface of the sea vessel and financial management tasks; thus, there was no easy method of global placekeeping (Gray, 2000). This manipulation increased the memory load of participants and increased the likelihood of postcompletion errors. Second, participants were interrupted as they performed the primary procedural tasks. Interruptions also increased the likelihood of postcompletion errors (Li et al., 2008).

One concern is that the manipulations used to increase the postcompletion error rate may also be influencing the logistic regression model such that the model represents the cognitive processes under these specific conditions. Thus, the model may not be generalizable outside of these conditions. To address this concern, Ratwani and Trafton (2010a) examined performance of the logistic regression model from Ratwani et al. (2008) on the financial management task with global placekeeping. In that experiment, environmental cues remained on the interface, allowing participants to use this information to keep track of their progress in the task. When the logistic regression model was applied to these data, the model accurately predicted more than 87% of the postcompletion errors with a false-positive rate under 8%. The high true-positive rate and the low false-positive rate suggest that the model can account for postcompletion errors in tasks with global placekeeping.

Examining performance of the logistic model under conditions where there is no interruption immediately preceding the postcompletion step is more difficult because very few errors occur under these circumstances. However, to address this concern, the performance of the logistic regression model was examined on data from four different experiments: Experiments 1 and 2 from this article, Ratwani et al. (2008), and Ratwani and Trafton (2010a). From these four experiments, there were 11 postcompletion errors where an interruption did not precede the postcompletion step. The predictive model accurately predicts nine (82%) of these cases. Although there are few data points, this analysis suggests that the model is generalizable to cases where interruptions are not present.

Although we have demonstrated that the logistic regression model can accurately predict postcompletion errors on two different tasks under various conditions, it is important to note that the model is not likely to be generalizable to other types of procedural errors. Postcompletion errors are a very specific type of procedural error that occurs under specific circumstances. A critical underlying theoretical component of the predictive model is goal forgetting and inattention to the postcompletion action. Postcompletion errors are more susceptible to goal forgetting because the postcompletion step occurs after the main goal of the task is completed and, consequently, is no longer maintained in working memory (Byrne & Bovair, 1997). Other procedural errors, on the other hand, are not necessarily due to goal forgetting. For example, anticipation and perseveration errors (i.e., repeating or skipping a step) occur at other points in the task hierarchy and have a different theoretical explanation (Byrne & Bovair, 1997; Cooper & Shallice, 2000; Trafton, Altmann, & Ratwani, 2009). These errors are associated with not knowing where in the task procedure work should be completed as opposed to a pure goal-forgetting mechanism. Thus, other types of procedural errors will likely require predictive models that better represent the underlying theoretical constructs that account for those specific errors. The eye-movement-based methodology that we have used to predict postcompletion errors may still be fruitful, but the specific predictors will likely be different for different types of procedural errors.

5.3. Application of a Real-Time System

There are several methods for reducing postcompletion errors. Task redesign is the most effective method in that postcompletion errors can be completely eliminated (Byrne & Davis, 2006). However, there are several large-scale, costly systems where redesigning the task to remove error prone steps is simply not possible. An alternative method to task redesign is to constantly present a cue indicating the next correct action every time the action needs to be performed (Byrne & Davis, 2006; Chung & Byrne, 2008). Unfortunately, this constant cue method has several disadvantages. First, users can become frustrated and annoyed with a cue that is always present (Ceaparu, Lazar, Bessiere, Robinson, & Shneiderman, 2004). Second, on interfaces that are rich with information, a constant cue can contribute to visual clutter on the interface (Heiner & Asokan, 2008). Finally, users often become accustomed to information that is always present on an interface, and users begin to ignore this information (Burke, Hornof, Nilsen, & Gorman, 2005). For these reasons, a constant cue may become ineffective over a long period.

The real-time prediction and prevention approach presents a cue only when there is a high likelihood of a user making an error and resolves many of the issues that are present with the constant cue method. Users are less likely to be annoyed with the cue because it will only be presented when it is really needed; the cue will not be constantly present reducing visual clutter; and, it is important to note, the cue will remain effective because the cue will not be seen one every postcompletion action.

Of course, the effectiveness of a real-time prediction and prevention system depends on the accuracy and reliability of the system. If the system is not reliable, the operator will not trust the system. Our system is accurate with an average of 88% of the postcompletion error actions correctly classified in Experiments 1 and 2. However, the ROC analyses that were performed to establish the optimal criterion of intervention assumed an equal weighting for true positives and false negatives. Depending on the setting in which the prediction and prevention system is implemented, the criterion can be shifted to be more liberal or conservative. For example, in high-risk environments where the consequences of a postcompletion error are severe, one might shift the criterion to reduce the number of false negatives (increasing false positives) in an attempt to ensure that a postcompletion error does not occur. Shifting the criterion is going to change the accuracy of the model and will influence operator trust in the system. These trade-offs require further investigation.

An important aspect of the real-time prediction and prevention system is the effectiveness of the cue. If a cue is presented and the user does not detect the cue before the user commits the error, the prediction and prevention system is completely noneffective. Several researchers have examined the properties of visual cues to determine what makes an effective cue (Byrne, 2008; Chung & Byrne, 2008; Czerwinski, Cutrell, & Horovitz, 2000; Trafton et al., 2005). For a cue to be effective, the cue should be blatantly obvious (i.e., highly salient), be very specific, and be presented just prior to the action that is being cued. Byrne (2008) found empirically that presenting a cue just prior to the action that needs to be cued as opposed to having the cue present from the beginning of the task is the most critical aspect of designing an effective cue.

The manner in which the real-time error prediction and prevention system was implemented in Experiment 3 leveraged the just in time aspect of effective cue presentation. In addition, the cue was salient and specific. Although the red arrow cue that was used was effective, this type of visual cue can contribute to visual clutter in more data-rich interfaces. Thus, the effectiveness of more general methods of cueing such as an auditory tone or a screen flash should be examined to alleviate the potential visual clutter problem.

The real-time eye-tracking approach to minimizing postcompletion errors would be the most beneficial in high-risk environments where task redesign is prohibitively expensive or not possible due to the inherent structure of the task. High-risk environments include areas such as aviation and other transportation systems, the medical domain, and military systems to name a few. There are several real-world settings where the real-time postcompletion prediction system could be implemented; two concrete examples are given here.

In the aviation domain, most modern cockpits of commercial airliners use a flight management system, which is a computer-based system for planning a flight. A pilot's duty is to enter information in this computer-based system, such as the origin, destination, and route, before takeoff so that the trip is efficiently planned. Entering information in the flight management system is a procedural task: Experienced pilots have performed the task hundreds to thousands of times. Some flight management systems also have a postcompletion step, specifically, after editing a route that requires several subgoals, one must push the *execute* button for the edited route to be finalized. Pilots occasionally fail to push the *execute* button and the route remains in its original form. The consequences of this error can be severe; for example, if the original route remains and autopilot relies on the entered route, the aircraft may actually be traveling on a route that is different from what the pilot and air traffic control intended. Errors in the cockpit are exacerbated by the fact that the cockpit can be a very busy environment with constant interruption and distraction from air traffic control and flight attendants (Boehm-Davis & Remington, 2009; Chou & Funk, 1990; Loukopoulos, Dismukes, & Barshi, 2001).

To completely eliminate the postcompletion error from the cockpit, the flight management task could be redesigned such that the postcompletion step no longer exists. However, the task redesign approach would require replacing all existing systems, and there would be large time and monetary costs to eliminate the post-completion error. Alternatively, an eye-tracking system could be implemented in the cockpit environment to help reduce the occurrence of postcompletion errors. The eye-tracking system would not require the pilot to change his or her behavior, and the system would not interfere with the pilot's performance; in fact, in field experiments, eye trackers have been embedded in cockpits before (Dehais, Causse, & Pastor, 2008).

The second example of a real-world scenario in which a postcompletion error prediction and prevention system could be used comes from the medical domain. To order medications for patients, physicians use a computerized order entry system. Detailed information about the patient and the required medications are entered into this system, and the medication that is eventually prescribed to the patient is based on this system. Entering an order into this system is a procedural task, and a documented error that occurs with this system is failing to log off the patient's order entry after completing the order details (Koppel et al., 2005). Physicians complete the main goal of entering the patient's order but occasionally make a postcompletion error by failing to log off the system after the main goal of entering the patient information is completed. This postcompletion error can have serious consequences because a physician may later make a new entry without realizing that the previous entry was still logged on; consequently, patients may receive unintended medications (Koppel et al., 2005).

Eliminating the postcompletion error from the physician order entry system by task redesign could be rather difficult because this step cannot be easily moved to a different part of the task structure such that it no longer occurs after the main goal of the task is completed. The postcompletion error can be prevented using the real-time prediction and prevention system. Current eye-tracking technology can detect and capture a user's eye movements whenever the user is in front of the computer monitor without recalibrating each time the user moves away from the monitor. Thus, whenever the physician is in front of the computer terminal entering information in the order entry system, the physician's eye movements can be recorded and analyzed to predict and prevent postcompletion errors.

Implementing an eye tracker in real-world scenarios like the cockpit of an airplane or on medical computers to prevent postcompletion errors may seem like an exorbitant step; however, in these high-risk domains the cost of a postcompletion error can potentially outweigh the cost of the eye tracker. Further, in these domains, an eye tracker can be used for other functions beyond postcompletion error prediction and prevention. For example, in the cockpit of an airplane, eye-tracking algorithms could also be implemented to determine pilot workload and fatigue (Marshall, 2007) as well as pilot situation awareness (Ratwani & Trafton, 2010b). Thus, the eye tracker could be used to measure several cognitive functions in real time to facilitate pilot performance. In addition, it may be possible to develop eye-movement-based algorithms to predict other types of procedural errors based on the methodology we have used to predict postcompletion errors. Once these algorithms are developed, they can be run on the same eye tracker, providing an even greater motivation to include eye trackers in these high-risk systems. Thus, the introduction of eye-tracking technology is not limited to the prevention of postcompletion errors; rather, eye trackers can benefit user interaction in several different ways.

NOTES

Acknowledgments. We thank Malcolm McCurry for designing the experimental tasks and programming the real-time eye-tracking system and Kristina O'Connell and Jenny Sousk for help with data collection. We also thank Richard Pew and Mike Byrne for comments on this manuscript.

Support. This work was supported by project number N0001409WX20173 from the Office of Naval Research to Greg Trafton and by a fellowship from the National Research Council to Raj Ratwani.

Authors' Present Addresses. Raj Ratwani, 4400 University Drive, MS3F5, Fairfax, VA 22030. E-mail: rratwani@gmu.edu. Greg Trafton, Code 5515, 4555 Overlook Avenue, S.W. Washington, DC 20375. E-mail: trafton@itd.nrl.navy.mil

HCI Editorial Record. First received July 8, 2009. Revisions received January 15, 2010 and May 24, 2010. Accepted by Richard Pew. Final manuscript received June 27, 2010. — *Editor*

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). Timecourse of recovery from task interruption: Data and a model. *Psychonomics Bulletin and Review, 14*, 1079–1084.

Baars, B. J. (1992). The many uses of error: Twelve steps to a unified framework. In B. J. Baars (Ed.), *Experimental slips and human errors: Exploring the architecture of volition* (pp. 3–34). New York, NY: Plenum.

Barber, C., & Stanton, N. A. (1996). Human error identification techniques applied to public technology: Predictions copared with observed use. *Applied Ergonomics, 27*, 119–131.

Boehm-Davis, D. A., & Remington, R. (2009). Reducing the disruptive effects of interruption: A cognitive framework for analysing the costs and benefits of intervetion strategies. *Accident Analysis and Prevention, 41*, 1124–1129.

Burke, M., Hornof, A., Nilsen, E., & Gorman, N. (2005). High-cost banner blindness: Ads increase perceived workload, hinder visual search, and are forgotten. *ACM Transactions on Computer-Human Interaction, 12*, 423–445.

Byrne, M. D. (2008). Preventing postcompletion errors: How much cue is enough? In B. C. Love, K. McRae, & V. M. Sloutsky (Eds.), *Proceedings of the Thirtieth Annual Conference of the Cognitive Science Society* (pp. 351–356). Austin, TX: Cognitive Science Society.

Byrne, M. D., & Bovair, S. (1997). A working memory model of a common procedural error. *Cognitive Science, 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*, 627–638.

Ceaparu, I., Lazar, J., Bessiere, K., Robinson, J., & Shneiderman, B. (2004). Determining causes and severity of end-user frustration. *International Journal of Human–Computer Interaction, 17*, 333–356.

Chou, C. D., & Funk, K. (1990). Management of multiple tasks: Cockpit task management errors. In *Proceedings of IEEE International Conference on Systems, Man and Cybernetics* (pp. 470–474). New York, NY: IEEE.

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*, 217–232.

Cooper, R. P., & Shallice, T. (2000). Contention scheduling and the control of routine activities. *Cognitive Neuropsychology, 17*, 297–338.

Czerwinski, M., Cutrell, E., & Horovitz, E. (2000). Instant messaging: Effects of relevance and time. *Proceedings of CHI 2000 Conference*. New York, NY: ACM.

Dehais, F., Causse, M., & Pastor, J. (2008). Embedded eye tracker in real aircraft: new perspectives on pilot/aircraft interaction monitoring. *ICRAT 2008*.

De Santis, A., & Iacoviello, D. (2009). Robust real time eye tracking for computer interface for disabled people. *Computer Methods and Programs in Biomedicine, 96*, 1–11.

Dodhia, R. M., & Dismukes, R. K. (2009). Interruptions create prospective memory tasks. *Applied Cognitive Psychology, 23*, 73–89.

Donmez, B., Boyle, L., & Lee, J. D. (2007). Safety implications of providing real-time feedback to distracted drivers. *Accident Analysis and Prevention, 39*, 581–590.

Eichele, T., Debener, S., Calhoun, V. D., Specht, K., Engel, A. K., Hugdahl, K., . . . Ullsperger, M. (2008). Prediction of human errors by maladapative changes in even-related brain networks. *Proceedings of the National Academy of Sciences, 105*, 6173–6178.

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters, 27*, 861–874.

Gray, W. D. (2000). The nature and processing of errors in interactive behavior. *Cognitive Science, 24*, 205–248.

Gray, W. D. (2004). Errors in interactive behavior. In W. S. Bainbridge (Ed.), *Encyclopedia of human–computer interaction* (pp. 230–235). Great Barrington, MA: Berkshire.

Heiner, A., & Asokan, N. (2008). Using salience differentials to making visual cues noticeable. *Proceedings of the 1st Conference on Usability, Psychology, and Security*. New York, NY: ACM.

Hodgetts, H. M., & Jones, D. M. (2006a). Contextual cues aid recovery from interruption: The role of associative activation. *Journal of Experimental Psychology: Learning, Memory & Cognition, 32*, 1120–1132.

Hodgetts, H. M., & Jones, D. M. (2006b). Interruption of the Tower of London task: Support for a goal activation approach. *Journal of Experimental Psychology: General, 135*, 103–115.

Ji, Q., Zhu, Z., & Lan, P. (2004). Real-time nonintrusive monitoring and prediction of driver fatigue. *IEEE Transactions on Vehicular Technology, 53*, 1052–1068.

Just, M. A., & Carpenter, P. A. (1976). Eye fixations and cognitive processes. *Cognitive Psychology, 8*, 441–480.

Kirwan, B. (1992a). Human error identification in human reliability assessment. Part 1: Overview of approaches. *Applied Ergonomics, 23*, 299–318.

Kirwan, B. (1992b). Human error identification in human reliability assessment. Part 2: Detailed comparison of techniques. *Applied Ergonomics, 23*, 371–381.

Koppel, R., Metlay, J. P., Coehn, A., Abaluck, B., Localio, A. R., Kimmel, S. E., & Strom, B. L. (2005). Role of computerized physician order entry systems in facilitating medication errors. *Journal of American Medical Association*(293), 1197–1203.

Li, S. Y. W., Blandford, A., Cairns, P., & Young, R. M. (2008). The effect of interruptions on postcompletion and other procedural errors: An account based on the activation-based goal memory model. *Journal of Experimental Psychology: Applied, 14*, 314–328.

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 Twenty-Eighth Annual Meeting of the Cognitive Science Society*. Mahwah, NJ: Erlbaum.

Loukopoulos, L. D., Dismukes, R. L., & Barshi, I. (2001). Cockpit interruptions and distractions: A line observation study. In R. Jensen (Ed.), *Proceediings of the 11th International Symposium on Aviation Psychology*. Columbus: Ohio State University Press.

Macmillan, N. A., & Creelman, C. D. (2005). *Detection theory: A user's guide*. Mahwah, NJ: Erlbaum.

Marshall, P. (2007). Indentifying cognitive state from eye metrics. *Aviation, Space and Environmental Medicine, 78*(1), B165–B175.

Monk, C. A., Trafton, J. G., & Boehm-Davis, D. A. (2008). The effect of interruption duration and demand on the resumpting suspended goals. *Journal of Experimental Psychology: Applied, 14*, 299–313.

Norman, D. A. (1981). Categorization of action slips. *Psychological Review, 88*(1), 1–15.

National Transportation and Safety Board. (2005). *Railraod accident report*. Washington, DC: Author.

Peng, C.-Y. J., Lee, K. L., & Ingersoll, G. M. (2002). An introduction to logistic regression analysis and reporting. *The Journal of Educational Research, 96*, 3–14.

Perrow, C. (1999). *Normal accidents: Living with high-risk technologies*. Princeton, NJ: Princeton University Press.

Pompei, F. J., Sharon, T., Buckley, S. J., & Kemp, J. (2002). An automobile-integrated system for assessing and reacting to driver cognitive load. In *Proceedings of Convergence 2002* (pp. 411–416). Piscataway, NJ: IEEE SAE.

Rasmussen, J. (1982). Human errors: A taxonomy for describing human malfunction in industrial installations. *Journal of Occupational Accidents, 4*, 311–335.

Rasmussen, J., & Jensen, A. (1974). Mental procedures in real-life tasks: A case study of electronic troubleshooting. *Ergonomics, 17*, 293–307.

Ratwani, R. M., McCurry, J. M., & Trafton, J. G. (2008). Predicting postcompletion errors using eye movements. In M. Czerwinski, A. M. Lund, & D. S. Tan (Eds.), *Proceedings of the 2008 Conference on Human Factors in Computing Systems, CHI 2008* (pp. 539–542). New York, NY: ACM.

Ratwani, R. M., & Trafton, J. G. (2008). Spatial memory guides task resumption. *Visual Cognition, 16*, 1001–1010.

Ratwani, R. M., & Trafton, J. G. (2009). Developing a predictive model of postcompletion errors. *Proceedings of the 31st Annual Meeting of the Cognitive Science Society*. Red Hook, NY: Curran.

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

Ratwani, R. M., & Trafton, J. G. (2010b). Single operator, multiple robots: An eye movement based theoretic model of operator situation awareness. *Proceedings of the 5th ACM/IEEE Conference on Human Robot Interaction*. New York, NY: ACM.

Rayner, K. (1998). Eye movements in reading and information processing: 20 years of research. *Psychological Bulletin, 85*, 618–660.

Reason, J. (1990). *Human error*. New York, NY: Cambridge University Press.

Schnell, T., & Wu, T. (2000). Applying eye tracking as an alternative approach for activation of controls and functions in aircraft. *Proceedings of the 5th International Conference on Human Interaction with Complex Systems*. New York, NY: ACM.

Sellen, A. J., & Norman, D. A. (1992). The psychology of slips. In B. J. Baars (Ed.), *Experimental slips and human error: Exploring the architecture of volition* (pp. 317–339). New York, NY: Plenum.

Shappell, S. A., & Wiegmann, D. A. (2001). Applying reason: The human factors analysis and classification system (HFACS). *Human Factors and Aerospace Safety, 1*(1), 59–86.

Stanton, N. A., & Stevenage, S. V. (1998). Learning to predict human error: Issues of acceptability, reliability and validity. *Ergonomics, 41*, 1737–1756.

Swets, J. A. (ed.) (1964). *Signal detection and recognition by human observers*. New York, NY: Wiley.

Swets, J. A. (1986a). Form of empirical ROCs in discrimination and diagnostic tasks: Implications for theory and measurement of performance. *Psychological Bulletin, 99*, 181–198.

Swets, J. A. (1986b). Indices of discrimination or diagnostic accuracy: Their ROCs and implied models. *Psychological Bulletin, 99*(1), 100–117.

Swets, J. A. (1992). The science of choosing the right decision threshold in high-stakes diagnostics. *American Psychologist, 47*, 522–532.

Tabachnick, B. G., & Fidell, L. S. (2001). *Using multivariate statistics* (4th ed.). Needham Heights, MA: Allyn & Bacon.

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 49th. Annual Meeting of the Human Factors and Ergonomics Society*. Santa Monica, CA: HFES.

Trafton, J. G., Altmann, E. M., Brock, D. P., & Mintz, F. E. (2003). Preparing to resume an interrupted task: Effects of prospective goal encoding and retrospective rehearsal. *International Journal of Human–Computer Studies, 58*, 583–603.

Trafton, J. G., Altmann, E. M., & Ratwani, R. M. (2009). A memory for goals model of sequence errors. *9th International Conference on Cognitive Modeling – ICCM2009*.

Wilson, G. F., & Russell, C. A. (2003). Operator functional state classification using multiple psychophysiological features in an air traffic control task. *Human Factors, 45*, 381–389.

APPENDIX A. RECEIVER-OPERATING CHARACTERISTIC (ROC) ANALYSIS

A receiver-operating characteristic analysis (ROC) comes from signal detection theory and has been used in numerous domains (i.e., decision sciences, machine learning, medical sciences, etc.). For the purposes of this article, we use ROC analysis as a method of examining the performance of a classifier at distinguishing between two discrete states of the world (Fawcett, 2006; Macmillan & Creelman, 2005). The combination of the two discrete states (e.g., error or no error) and the two predicted classifications (predicted error or no predicted error) results in four possible outcomes: true positives, false negatives, false positives, and true negatives. An ROC curve is a plot of the false positive (*x*-axis) and true positive (*y*-axis) rates based on the classifier function. In ROC space, the upper left-hand corner of the plot represents perfect prediction where the true positive rate is 1 and the false positive rate is 0 (Swets, 1992).

In cases where a classifier function does not result in a binary outcome, such as logistic regression that results in a log odds (which can be converted to predicted probability), a criterion needs to be established to categorize the discrete states of the world. An ROC analysis provides a method for visualizing the performance of the classifier function at each possible criterion level. Each criterion level will have an associated true-positive and false-positive point in ROC space. By selecting the criterion level associated with the ROC curve point that is closest to the upper left-hand corner in ROC space, the optimal criterion can be determined.

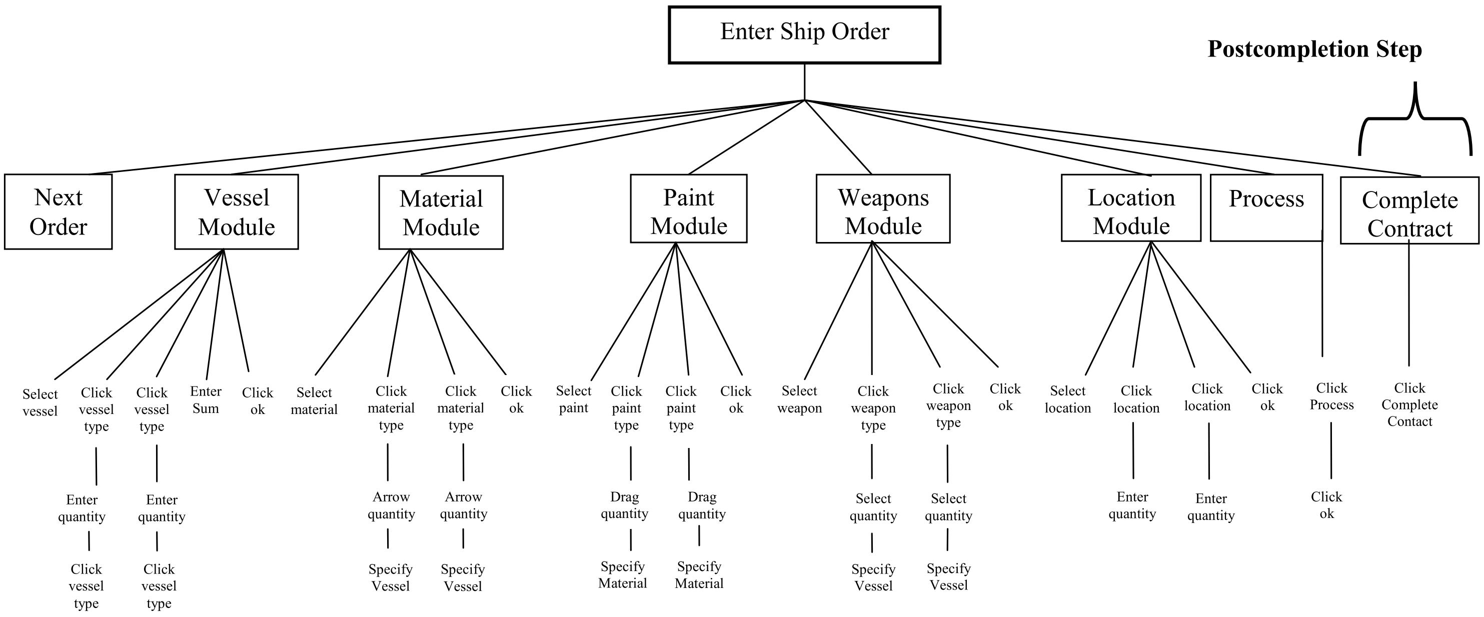
There are several ways of gauging the performance of a classifier using an ROC analysis. One of the most common measures is the area under the curve (AUC). The AUC is equivalent to the probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance (Fawcett, 2006). When applying a classifier with a specified criterion to set of binary instances, one can also examine the four possible states that are represented in a confusion matrix (Fawcett, 2006). In addition, sensitivity is also examined that refers to the separation of signal and noise; the greater the sensitivity, the better the classification. Sensitivity is commonly measured by d' and generally varies between .5 and 2.0 (Swets, 1964).

APPENDIX B. SEA VESSEL TASK

Figure 9 shows the sea vessel production task structure and the actions required to complete a single order. The procedure for performing the sea vessel production task is detailed here:

- Begin the task by clicking the Next Order button. Orders for two different types of navy sea vessels will appear in the Navy Manifest window. The Navy Manifest provides specific information in regard to the Quantity, Vessel type, Material, Weapons, and Paint that should be included in each order.
- The first module to complete is the Vessel Module. Begin by clicking the Vessel Selector button from the Selector window. A message reading "Vessel Activated" will briefly appear, indicating that you have activated the Vessel module. In the Vessel module, click the appropriate vessel types based on the Navy Manifest and enter in the specific quantities as indicated in the Navy Manifest. Click the Total button, calculate the sum of the two vessel types, and enter the sum in the Total field. To complete the module click Ok.

FIGURE 9. Sea vessel task structure.



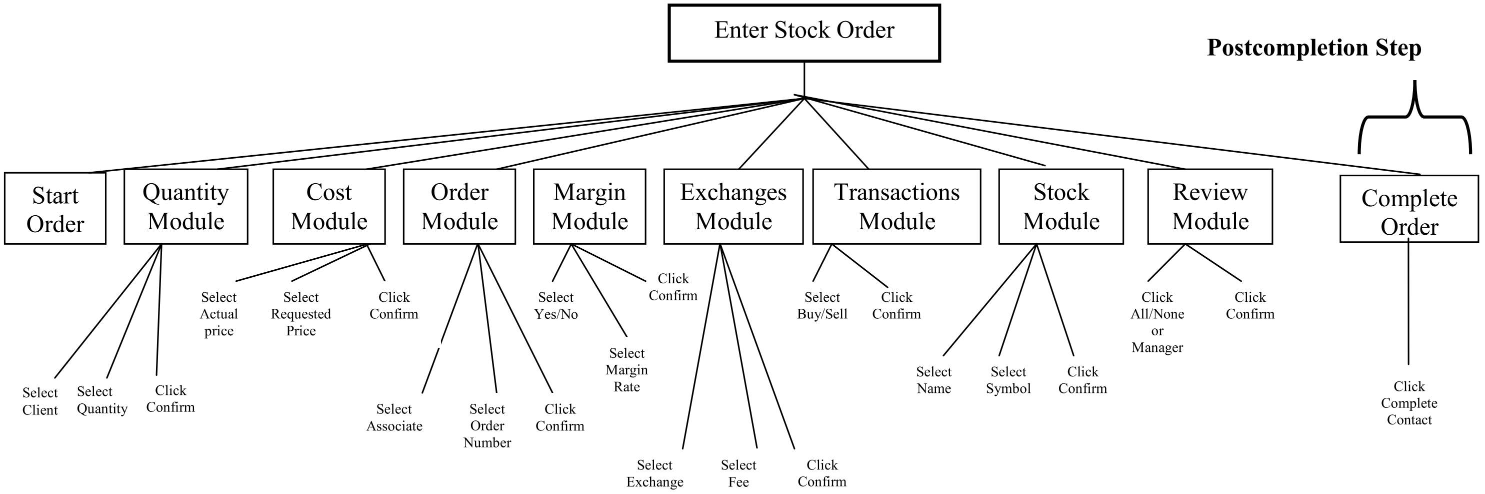
- The second module to complete is the Material Module. Begin by clicking the appropriate Selector button from the Selector window and enter in the details in the Material module based on the information provided in the Navy Manifest. Click Ok to complete the module.
- The third module to complete is the Paint Scheme Module. Begin by clicking the appropriate Selector button and enter in the details in the Paint Scheme module based on the information provided in the Navy Manifest. Click Ok to complete the module.
- The fourth module to complete is the Weapons Module. Begin by clicking the appropriate Selector button and enter in the details in the Weapons module based on the information provided in the Navy Manifest. Click Ok to complete the module.
- The fifth module to complete is the Location Module. Begin by clicking the appropriate Selector button and enter in the details in the Location module based on the information provided in the Navy Manifest. Click Ok to complete the module.
- Click the Process button to process the order. Upon clicking the Process button a pop-up window will appear indicating the number of sea vessels you have ordered. Click the Ok button to acknowledge the message.
- Click the Complete Contract button to finish the order.

APPENDIX C. FINANCIAL MANAGEMENT TASK

Figure 10 shows the financial management task structure and the actions required to complete a single order. The procedure for performing the financial management task is detailed next:

- Begin the task by clicking the Start Order button of an order that is currently active. To determine whether an order is active, compare the current prices of the stocks in the stock information box to the desired buy and sell prices of the orders. If a client wishes to sell a stock and the desired sell price is at or above the current market price of the stock, the order is valid. If the client wishes to buy a stock and the desired buy price is at or below the current market price, the order is valid.
- The first module to complete is the Quantity module. In the Quantity module, select the appropriate client and quantity based on the order and then click the confirm button.
- The second module to complete is the Cost Module. Select the appropriate actual stock price based on the Stock Information box, select the requested price based on the order, and click confirm.
- The third module to complete is the Order Info Module. Select the associate based on the Stock Information box, select the order number based on the order, and click confirm.

FIGURE 10. Financial management task structure.



- The fourth module to complete is the Margin Module. Determine whether the order is a margin order by looking at the client's order. If the order is a margin order, select the appropriate margin rate based on the Stock Information box. After entering the information, click confirm.
- The fifth module to complete is the Stock Exchanges module. Look at the order to determine which exchange the order is on. Click the appropriate exchange and based on the exchange select the appropriate fees. After entering the information, click confirm.
- The sixth module to complete is the Transaction module. Click buy or sell based on the order and then click confirm.
- The seventh module to complete is the Stock Information module. Select the name and symbol of the stock based on the order. After entering these details, click confirm.
- The eighth module to complete is the Review module. If the order if or more than 5000 shares select All/None, otherwise select Alert Manager. After making the selection, click confirm.
- After clicking the confirm button on the eighth module, a window will pop up that displays the stock in the transaction, the shares being bought or sold, and the commission. Click ok to acknowledge the message.
- Click the Complete Order button to finish the order.