RICE UNIVERSITY

# Visual Cues to Reduce Error in Computer-based Routine Procedural Tasks

by

**Phillip H. Chung**

A THESIS SUBMITTED
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE

**Masters of Arts**

APPROVED, THESIS COMMITTEE:

Michael D. Byrne, Assistant Professor
Psychology

David M. Lane, Associate Professor
Psychology

Kenneth R. Laughery, Emeritus Professor
Psychology

HOUSTON, TEXAS
FEBRUARY, 2004

ABSTRACT

**Visual Cues to Reduce Error in Computer-based Routine Procedural Tasks**

by

Phillip Hyungmok Chung

Research has shown that one type of common procedural error, postcompletion error, occurs systematically under high working memory load. Studying the effects of different interventions on this reproducible and well-explained error type may help extend our understanding of the underlying psychological mechanisms behind human error and interactive task behavior. Experiment 1 was an investigation of the error-reducing efficacy of a simple visual cue and a separate downstream error cost condition. While neither was found to be reliably effective, this inquiry provided valuable insight that led to a follow up study. In Experiment 2, a cue based on design guidelines and a mode indicator were implemented to explore possible reasons for why the previous interventions failed. Only the cue had a reliable effect, demonstrating the difficulty of designing a successful intervention. Finally, a computational model based in ACT-R was developed to provide theoretical demonstration of this finding.

# ACKNOWLEDGEMENTS

This thesis would have undoubtedly been impossible without the steadfast guidance and support of my commitee chair and mentor, Mike Byrne. I am also thoroughly obliged to my other committee members David Lane and Ken Laughery, whose vast experience and knowledge were crucial in the direction and progress of this work. Finally, I must thank the other department professors and chair for their teaching and backing in this endeavour.

Outside the university, Jiajie Zhang and Todd Johnson at the University of Texas School of Health Information Sciences supported me during an exciting second year research assistantship in medical informatics. Additionally, the folks at the Usability Testing and Analysis Facility at Johnson Space Center have so kindly introduced me to the realm of space human factors.

Finally, I am forever indebted to my Creator, family, friends, teachers, and advisors of the past who have guided me in this journey through life both before and during my sojourn here in Houston. As much as our behavior and development is shaped by the environment, so has their faith and interest in me been vital to the culmination of this most recent chapter of my life.

TABLE OF CONTENTS

# LIST OF TABLES

Page

# LIST OF FIGURES

Page

# 1. INTRODUCTION

The study of human error has held a place within the domain of human factors for quite some time. As early as the 1960s, attempts were being made to classify human error in industrial settings (Rook, 1962; Swain, 1963). With the introduction of automation and computers, an outstanding domain for human error was established: human–computer interaction. Subsequently, there has been some work to categorize errors occurring in such situations, yet for the most part understanding of this very human phenomenon remains fairly nebulous. As John and Kieras state (1996), "No methodology for predicting when and what errors users will make as a function of interface design has yet been developed and recognized as satisfactory…even the theoretical analysis of human error is still in its infancy."

In recent years, several elaborate models and taxonomies of human error have been developed for the purpose of qualitative diagnosis (e.g., GEMS, Reason, 1990; SRK, Rasmussen, 1987). While useful for post hoc explanations, the predictive power of these is quite limited. Moreover, automation and systems initially developed to reduce human error and support human work often end up introducing further task complexities and opportunities for error. The expectation that errors will occur seems prudent no matter how stringent the preventative measures. However, it does not seems so farfetched to think that a deeper understanding of why they arise may help us not only evaluate but design safer interfaces. In-depth analyses of various error interventions and their relative effects on human performance may provide hints as to why errors occur in instances of human-machine interaction. This should in turn give us ideas about what can be done, from a design perspective, to reduce their frequency.

Beginning with a general introduction to human error, the scope of this paper narrows to focus on one specific procedural error relevant to this work, postcompletion error. A general review of the ground laying work by Byrne and Bovair (1997) and Serig

(2001) will lead into the first experiment. Finally, the follow up study and computational model will be described.

## 1.1. Human Error

Reason (1990) defines human error as, "all those occasions in which a planned sequence of *mental* or *physical* activities fails to achieve its intended outcome." Personal experience and history tell us that humans are inevitably fallible both mentally and physically. "To err is human," goes the saying. Designers, thus, must take this into consideration when developing systems that require interaction with people. Historical catastrophes, including the infamous nuclear plant accident at Three-mile Island, the 747 collision at Tenerife in 1977, and eye-opening reports, like the Institute of Medicine study on medical errors (1999), have brought to attention the reality of human error and the serious consequences they can carry. It was events such as these that catalyzed early efforts to study human error within their respective domains.

### *1.1.1. Error Taxonomies*

According to Sheridan (1997), there are at least three potential categories of human error taxonomies in the study of human-machine interaction. The first of these relates to the locus of behavior at which an error can occur: sensory, memory, decision, or motor. A second major distinction has been frequently made from the behavioral viewpoint between omission and commission errors (Swain, 1963). Omission errors arise in instances of human-machine interaction when uninformed decision makers do not take an appropriate action despite non-automated indications of problems. On the other hand, errors of commission can occur when decision makers follow automated information or directives in light of more valid or reliable indicators hinting that the automation is not proposing a proper course of action. When considered in the domain of human-computer interaction, they are thought to often be the result of not looking for confirmatory or non-

confirmatory information, or discounting other sources of information in the presence of computer generated cues (Mosier & Skitka, 1996).

Omission errors have been further divided into mistakes, violations, slips and lapses, as seen in Table 1 below, each occurring at a different stage of action control (Reason, 1997). Most relevant to the current work, slips refer to instances where correct intentions or plans are not well-executed and a necessary item is omitted from the sequence (Reason, 1990). They are defined by Senders and Moray (1991, p.27) as actions "not in accord with the actor's intention, the result of a good plan but a poor execution." Conversely, lapses largely involve failures of memory that do not necessarily manifest themselves in actual behavior. As Reason (1990) states, a lapse can simply be described as having something "slip your mind."

Table 1

*Omission error types*

| Failure | Nature |
|---|---|
| Mistake | A necessary item is unwittingly overlooked. |
| Violation | The item is deliberately left out of the action plan. |
| Lapse | The intention to carry out the action(s) is not recalled at the appropriate time. |
| Slip | The actions do not proceed as intended and a necessary item is unwittingly omitted from the sequence. |
| Slip or Violation | The actor neither detects nor corrects the prior omission. |

Note. From "*Managing the Risks of Organizational Accidents*," by J. Reason, 1997, Aldershot: Ashgate Publishing Company.

### *1.1.2. Contrasting Perspectives*

Reason's (1984) basic law of error states that "whenever our thoughts, words, or deeds depart from their planned course, they will do so in the direction of producing

something that is more familiar, more expected and more in keeping with our existing knowledge structures and immediate surroundings, than that which was actually intended," (p.184). Our sensitivity to the environment and proclivity to err has been well demonstrated in both research and in the real world. Nevertheless, the conventional perspective has taken this viewpoint to an extreme: when accidents occur they are immediately attributed to human inattention, laziness, carelessness, and negligence (Van Cott, 1994). Blame is directed at the human rather than the machine or system, a response Reason (1994) calls the "blame trap." As an example, physicians in medical school and residency, driven to strive for error-free practice and perfection, consequently develop a tendency to lay blame on themselves and others in the workplace when things go wrong (Leape, 1994).

However, in keeping with a human factors perspective that considers man and machine as a coupled, interactive system operating within a given environment, the injustice of the "blame trap" approach has been frequently demonstrated. The nuclear power plant accident at Three-mile Island is a classic example of poor design leading to failure within a human-machine system. An operator from the plant stated in his testimony at the 1979 Congressional hearing: "If you go beyond what the designers think might happen, then the indications are insufficient, and they may lead you to make the wrong inferences…hardly any of the measurements that we have are direct indications of what is going on in the system," (Testimony, 1979). As Norman suggests, "change the people without changing the system and the problems will continue" (Bogner, 1994).

### 1.2. Psychological Underpinnings

The same psychological processes producing successful performance can also be pointed to as the source of human error (Baars, 1992; Reason, 1990). Reason (1990) describes this idea in terms of a "cognitive balance sheet," where correct performance and systematic errors are placed on opposing sides. For example, the development of

automatic behavior through the delegation of control to lower level processes introduces the opportunity for behavioral slips to occur. To further explore this notion, two related models from the broader concept of human performance should be considered.

Rasmussen's (1987) "Skill-Rule-Knowledge" (SRK) model of task performance provides a general framework of cognitive control mechanisms. It identifies three separate levels of cognitive control displayed during task performance: skill-based, rule-based, and knowledge-based. Each corresponds to a different degree of familiarity with the task and environment, with knowledge-based behavior representing the least degree of control and familiarity and skill-based the highest. With experience the person can proceed sequentially through the three stages of the model, moving from lowest (knowledge-based) to highest (skill-based). At the rule-based level, failure modes stem from either the application of bad rules or misapplication of good rules due to incorrect rule selection. These rules may be active simultaneously, with several competing for instantiation. From the mental model that the operator has constructed in their mind about a system, rules for behavior are selected based on:

1. *Match* to salient features of the environment or internally generated messages.
2. *Strength* or the number of times a rule has performed successfully in the past.
3. *Specificity* to which a rule describes the current situation.
4. *Support* or the degree of compatibility a rule has with currently active information.

These same selection criteria also control the occurrence of errors at the rule-based level, in keeping with Reason's (1990) idea of a "cognitive balance sheet."

A closely tied and widely cited taxonomy of error is Reason's (1990) "Generic Error-Modeling System" (GEMS), which focuses on cognitive error modes and is context-free – i.e., the cognitive factors, rather than the environmental or contextual, are emphasized, allowing error classification in a variety of settings. Following Rasmussen's (1987) SRK model of task performance, errors are thought to occur differently at each of

the three performance levels due to the distinct cognitive processes operating there. The classification system divides errors at the skill-based level into slips and lapses, mistakes at the rule-based level, and mistakes at the knowledge-based level. Since SRK allows behavior to move between the levels of performance, GEMS also accounts for errors at any or all of the three levels on a given task.

These are only two major examples of theoretical constructs devised to classify and help explain the occurence of human errors. Others include the application of GOMS (Goals, Operators, Methods, and Selection Rules) by Wood and Kieras (2002) to predict human error, statistical methods based on task analysis such as that of Baber and Stanton (1996), and the Cognitive Reliability and Error Analysis Method (CREAM; Hollnagel, 1998), the second generation Human Reliability Analysis (HRA).

### 1.3. Postcompletion Error

Noting the lack of specificity in the existing theories of human error, Byrne and Bovair (1997) moved to develop a computational theory for one widely cited (e.g., Rasmussen, 1982; Thimbleby, 1990; Young, 1994) omission error, postcompletion error. Postcompletion errors can be broadly defined as errors that occur when the task structure demands "that some action…is required after the main goal of the task…has been satisfied or completed," (p. 32). With this particular class of error, the actor possesses the correct knowledge necessary to execute the task, usually performed frequently and correctly. Yet, even for operators who have developed high levels of familiarity with the task, the isolation of a postcompletion step within the task structure makes omissions there likely. This is particularly true when the actor is further taxed by external factors such as a working memory load or fatigue.

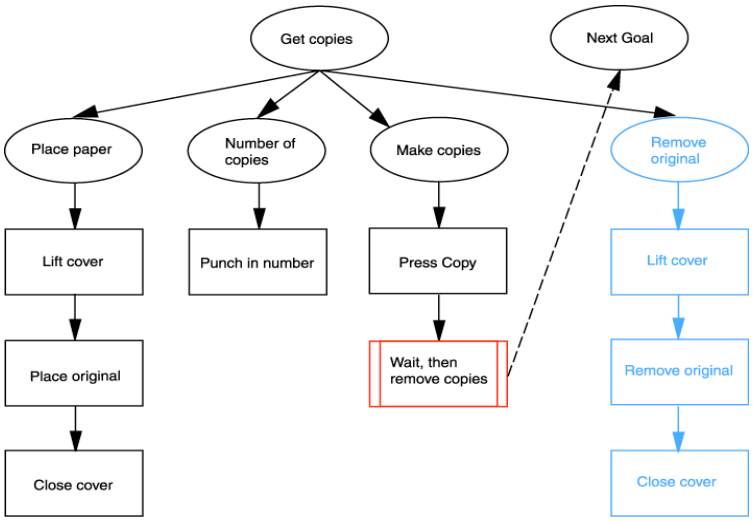
Some commonplace examples of postcompletion errors include forgetting to remove the original after making a photocopy, leaving a card in the ATM after withdrawing cash, and failing to replace the gas cap after filling up a car. Generally, these

errors arise when completion of the main task goal requires that an initial state change to the system be made. This state change generates a related subgoal of returning the system to its original state after completing the main task goal: the postcompletion step. With the example of the gas cap, the state change made to the car by its removal requires that it be later returned to its original state (state maintenance) after completing the main task goal of filling the car up with gasoline. Hence, forgetting to replace the gas cap after a fill-up is considered a postcompletion error.

Byrne and Bovair (1997) in hypothesized that these errors were due to excessive working memory load leading to goal loss, or an omission of a step from the task at hand. Since with postcompletion errors the actor omits a specific subgoal rather than forgetting what to do altogether (the overlying main task goal), the source of the error was thought to more likely be working memory than long-term memory. A more recent study by Reason (2002) examined the photocopy example in detail (Figure 1), finding postcompletion errors to be the most common type of omission in that task. Three high-level explanatory observations were provided:

1. The emergence of the last copy generates a strong but *false completion signal* since the main goal of copying is achieved before all necessary steps (subgoals) are complete.
2. The *proximity* of this false signal to the end of the main task allows for the attention to be increasingly diverted to the subsequent task.
3. The emergence of the last copy indicates that it is no longer necessary to put in another original leaving it *functionally isolated*.

*Figure 1*. Photocopy task structure.




Byrne and Bovair (1997) explain postcompletion errors simply as a "goal loss from working memory," (p.38). As a computational theory it can be more precisely described as occuring when the current goal supplies activation to its subgoals concurrently resident in working memory. When a parent goal is eliminated from memory, any subgoal of the satisfied parent goal also loses activation. In cases where working memory load is high, the parent goal falls below threshold in working memory too soon, leaving unsatisfied subgoals without enough activation to reach threshold. This is attributed to the assumption that higher levels of working memory load lead to faster rates of decay of information from working memory. On the other hand if the parent goal remains active for some time, the associated subgoals will eventually reach threshold and be executed.

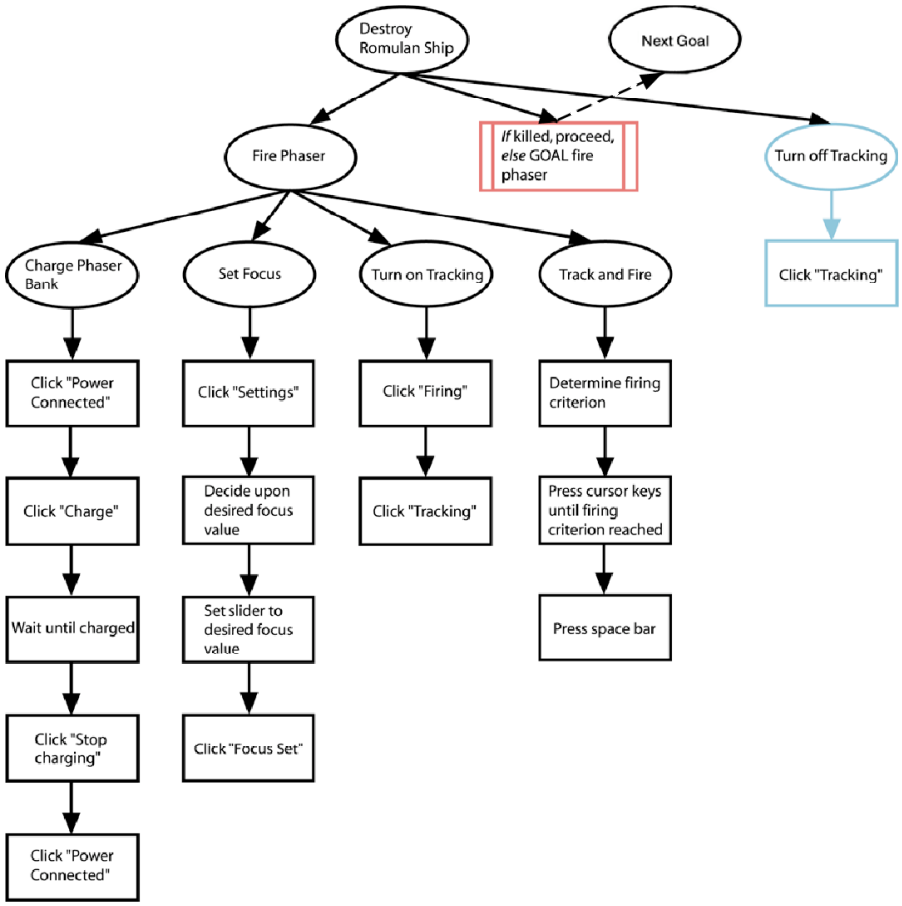
Using the Collaborative Activation-based Production System (CAPS) cognitive architecture, which contains both a long term and working memory system, Byrne and Bovair (1997) were able to test their theory of postcompletion error and quantify its predictions. They compared findings from their model to experiment results produced by

participants on the same computer based task. As hypothesized, postcompletion errors were found to systematically increase in frequency as working memory demands were increased through task complexity and external load. The model demonstrated this, as the error never occurred under high activation ceilings (high working memory capacity, low load), whereas with low activation ceilings (low working memory capacity, high load) the error always arose in the postcompletion version of the task.

## 2. PREVIOUS INQUIRIES

The more recent study by Serig (2001) verified Byrne and Bovair's (1997) account of the strong influence of high working memory load on postcompletion error commission. As in the work of Byrne and Bovair, working memory load was imposed by an auditory letter-tracking task. Consequently, the resulting postcompletion error counts exceeded figures from traditional stochastic theory (which would predict postcompletion errors to account for 1/12 steps or 8.3% of the errors made) as predicted. Participants were also trained and tested on computer-based tasks similar to those used by Byrne and Bovair. The target postcompletion task and three similar dummy tasks were administered under the theme of a Star Trek Bridge Officer qualification course. Only the Tactical task, however, incorporated a postcompletion task structure (see Figure 2) similar to the photocopy example shown in Figure 1.

*Figure 2.* Tactical task structure.

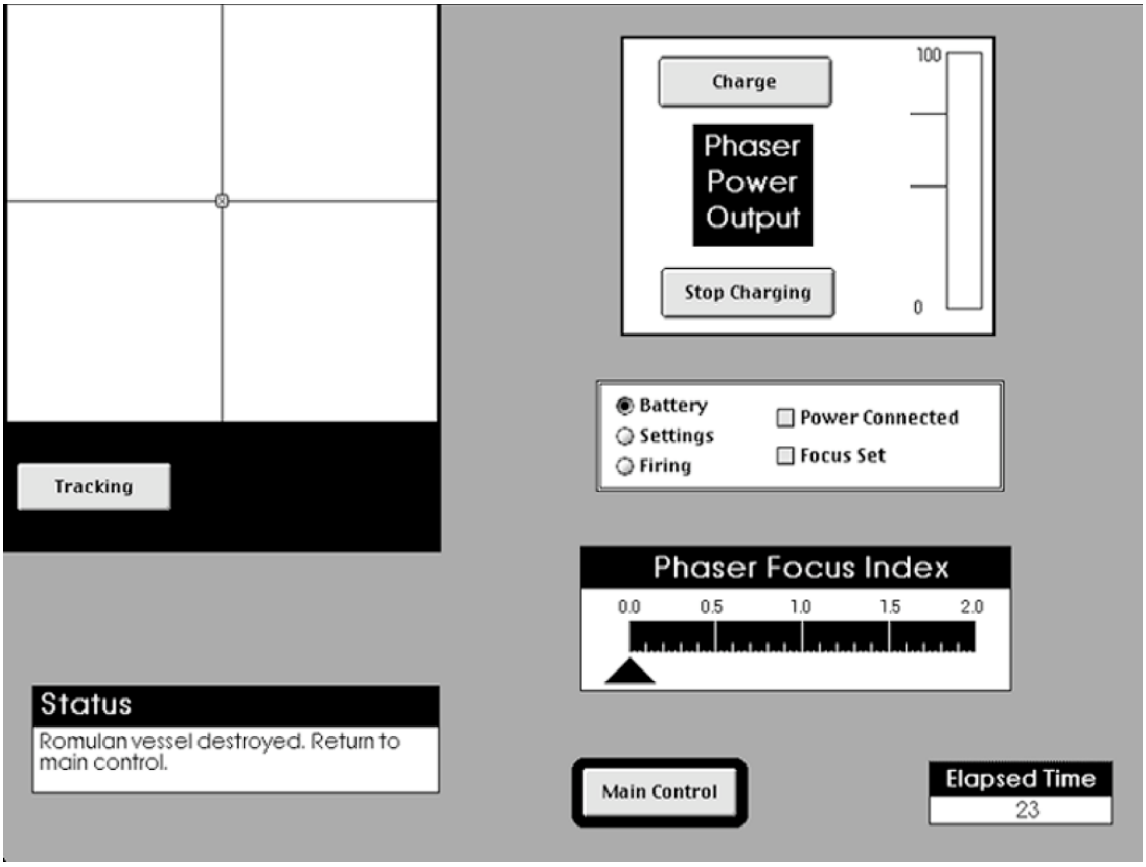



Note. From Serig, 2001).

In brief, the Tactical task entails operating and firing a starship "phaser" at a select target. The main goal of the task, as seen in Figure 2, is to destroy a Romulan starship. Participants are made aware of their success or failure by means of auditory and visual feedback presented on the Tactical console (see Figure 3). Most of the steps related to the postcompletion step take place in the white window at the upper left hand corner of the console. During a trial, participants track and aim at a moving object appearing in that window using the arrow keys on the keyboard and fire the phaser by hitting the spacebar

when the object is aligned with the crosshairs. Depending on whether the outcome is a hit or miss, there is either an exploding sound or a whoosh. If it is a miss, the participant is required to restart the task from beginning. If the target is hit and successfully destroyed, the participant next completes the postcompletion step and exits to the main console where they may proceed to the next task.

*Figure 3.* Tactical console.




The work of Serig (2001) showed a decrease in errors with over time, in accordance to the popular theories of human error and task performance. As participants proceeded through the three days of training, frequency of error decreased reliably across tasks. By the final day of the experiment, Serig (2001) found that the participants were performing the tasks with a medium to high level of skill. A point-based system and timer

implemented in the computer task encouraged individuals to be wary of their pace, possibly generating a speed-accuracy tradeoff. Serig (2001) suggests that, in the absence of such time and performance pressures, operators who are "far enough out on the skill learning curve" would be unlikely to commit errors even with the added working memory load (p.120).

According to Baars (1992), slips seem to "reflect an evolutionary tradeoff between the risks of unintended error and the benefits of increased speed and flexibility," (p.19). In the case of an animal escaping its predator, the animal risks error for the more vital benefit of speed when it leaps rapidly to escape. In a less pressing situation, the animal can take more time to act, greatly reducing the probability of error. Baars states that, "in the most general case, a slip represents a loss of voluntary control rather than a rule violation," (p.21).

At another level it seems that the additional demands imposed in the Serig (2001) experiment of either the working memory load or the time and performance pressures still afford some volition and control for the operator. Just as an animal is fundamentally still able to decide whether or not to make that speed-accuracy tradeoff in its escape, intuition tells us that a person may also slow their behavior to a point where errors are few if they wish. This is what Serig (2001) found, as participants' step completion times increased at the postcompletion step with corresponding declines in error. Nevertheless, in many cases the preference for speed seems natural and the sacrifice of absolute control is made, opening the door for human error.

### 2.1. Hillclimbing and Least-effort

Often this preference for speed is so robust that even additional effort generated by external praise or reprimand is limited in its effect on reducing the frequency of error (Serig, 2001). Previous research suggests that people tend to be "cognitive misers," in that they take the road of least cognitive effort and use decision rules of thumb or

heuristics rather than systematically analyzing each decision (e.g., Fiske & Taylor, 1994). Furthermore, studies of problem solving have demonstrated the pervasive tendency of humans to hillclimb (difference-reduction), or take the shortest perceived route to complete a task or solve a problem. With hillclimbing, past states do not have to be retained and planning more than the next step is not required, thus reducing the required level of cognitive effort (Anderson, 1995).

Evidence for this very human inclination can be readily observed almost anywhere on a university campus. Numerous well-trodden paths diverge from sidewalks across lawns, formed over time by students cutting corners to get to class. Curiously, the "shortcut" often offers little or no real benefit at all. Polson and Lewis (1990) uncovered a similar propensity in computer-based tasks where perceptual similarity alone is often used to select actions appearing to offer the greatest progress towards the goal. This idea has taken on several iterations such as the label-following heuristic proposed by Englebeck (as cited in Polson & Lewis, 1990) which describes the tendency of novice users to select actions in computer-based tasks by comparing the descriptions of available actions with a description of the goal.

Gray (2000) has applied the hillclimbing principle to the interactive behavior involved in programming a VCR. He states that people adhere to a "least-effort principle" in operating the device, mapping prior knowledge to the device and using place-keeping. For the task of programming a VCR, people progress using both global and local place-keeping, which "entails knowing what parts of the task have been completed and what parts remain to be accomplished," (p.221). Task-based and device-specific goal completion is tracked at the global level while at the local level progress on the current goal is followed.

Least-effort in place-keeping leads to what Gray (2000) calls display-based difference-reduction, where differences between the current state of the world and the final goal state are progressively reduced using perceptual information rather than

knowledge in memory. In this manner, local place-keeping becomes a primarily perceptual rather than cognitive task, as Gray found with VCR programming. This is similar to the idea of distributed representations (Zhang & Norman, 1994) or the dual-space nature of human-device interaction (Rieman, Young, & Howes, 1996). Zhang and Norman demonstrated that external representations, where the information is present on the device itself rather than in the head of the operator (internal), drastically decrease and even eliminate human error. By placing constraints on the problem space and exploiting more efficient perceptual processes and information in lieu of memory, they effectively reduce cognitive load.

### 2.2. Postcompletion Error Factors

Serig (2001) and Byrne and Bovair (1997) demonstrated that high working memory load and time and performance pressures lead to postcompletion and other general errors. Working memory itself has been shown in several studies to constrain performance in such activities as arithmetic calculation (Hitch, 1978), concept formation (Bruner et al., 1956), reasoning (Johnson-Laird, 1983) and diagnostic 'trouble-shooting' by electronic technicians (Rasmussen, 1982). Additionally, the various theories of attention generated in the fifties and sixties all concur on the idea of a general 'bottleneck' in cognition and limited resources in the human information processing system. In situations of multi-tasking, particularly, the limitations of attention become markedly conspicuous. Broadbent (1982) states: "The main interference between two tasks occurs at the point where they compete most for the same function."

According to Miller and Swain (1986), stress and inexperience can increase operator error probability by as much as a factor of ten; stress alone by a factor of five. Unfortunately, considering the job characteristics of a surgeon or air traffic controller, for instance, eliminating external pressure seems highly improbable. In instances where the demands on attention may be less, still other factors come into play, such as a lack of

arousal and the many idiosyncrasies of human behavior previously discussed. This is distressing considering that the necessary condition for the occurrence of a slip is in fact the presence of attentional capture related to either distraction or preoccupation (Reason, 1979; 1984). It is thus prudent to presume that humans will tend to err under these conditions and build systems that accomodate for this rather than placing people in the "blame trap."

### 2.3. Hierarchical Control Structures and Goal Management

Many of the assumptions behind the current theory of postcompletion error reside on the foundational concept of hierarchical control structures and their retention by skilled operators. Evidence from cognitive modeling by Kieras, Wood, and Meyer (1997) has revealed that even well practiced experts, such as telephone assistance operators, do not abandon hierarchical control structures. In the previous work by Byrne and Bovair (1997) and Serig (2001), participants reliably generated errors at the appropriate (postcompletion) step, following the theory of a hierarchical postcompletion task structure.

As Altmann and Trafton (1999) suggest, this ability to break down complex tasks and problems into hierarchies and subgoals, "may be to complex cognition what the opposable thumb is to complex action." Traditionally, these types of goal-based processing strategies have relied solely on a "task-goal" stack that essentially predicts perfect memory for old goals. However, an activation-based model of memory for goals (MAGS; Altmann, 2002, Altmann & Trafton, 1999) offers an alternative account to this approach that provides a more straightforward account for the types of errors found in human behavior. In essence memory and the environment (i.e., dual-space, Rieman & Young, 1996; internal and external representations, Zhang & Norman, 1994) are substituted for a goal stack, and task goals are considered as ordinary memory elements with encoding and retrieval processes that must overcome noise and decay. Retrieval

cues from the environment dictate the reactivation of suspended goals (such as the postcompletion step; see Figure 2) with perceptual heuristics acting as a substitute for the stack-native last in, first out rule. This model makes several predictions about postcompletion errors:

1. Any *salient* cue (e.g., a loud beep) should be sufficient to prime a postcompletion action (suspended goal).
2. It should *not* be necessary to put the postcompletion action on the critical path.
3. Reminders at the start will *not* help a PCE at the end (masked by other goals).
4. Just-in-time priming from environmental cues are the *only* reliable reminder.

## 3. EXPERIMENT 1

While designers may opt to place the postcompletion action "on the critical path" to reduce or eliminate their occurrence, such as with many ATMs, this is often impossible or too expensive given the existing system and should not be necessary (Altmann, 2002). In light of these theories' projections, it should be viable to reduce postcompletion errors or omissions in an interactive task by simply cueing or priming a suspended goal. If the necessary condition for slips is attentional capture of some form, reminders or environmental cues similarly exploiting attentional capture should be able to reactivate the suspended postcompletion action subgoal at the appropriate time. Similarly functioning cognitive aids are frequently used in industrial settings and aviation to reduce human error. By prompting the actor to make sure that all steps have been completed in the task, they fundamentally augment the limited capacity of working memory that is the root of many errors. Furthermore, many existing everyday devices and computer applications have indicator lights, beeps, alarms, etc. that act as reminders, although their efficacy may sometimes be questionable.

What type of cue best generates the necessary attentional capture? A great deal of physiological and psychophysical research has shown that the visual system is

particularly sensitive to abrupt stimulus onset. Work by Yantis and Jonides (1984) indicates that there may be a direct link between the mechanism specialized for the detection of onsets and the natural properties of the visual attention system. Other findings, also from Yantis and Jonides (1988), have shown that visual onsets produce attentional capture, whereas color and intensity, which produce powerful effects in other visual tasks, cannot. Based on this evidence and the pervasiveness of simple visual cues implemented as reminders in real-world settings (Reason, 1997) and computer systems, a single red visual onset was chosen as a reminder to appear at the postcompletion step next to the relevant button at the appropriate time.

### 3.1. Predictions

Using modified versions of the computer-based tasks employed by Serig (2001) and Byrne and Bovair (1997), the objective of Experiment 1 was foremost to study the effects of: (1) a simple automated visual cue and (2) a downstream error cost on postcompletion error. Previous research on human error had been plagued by the difficulty of experimentation (Wood & Kieras, 2002), particularly that of devising realistic tasks that produce errors at a high enough frequency to be studied. However, postcompletion errors are well accessible for the study of error interventions, having been shown to be robust and reproducible in previous inquiries and observations from real-world settings (e.g., Byrne & Bovair, 1997; Reason, 2002; Serig, 2001).

The idea of humans as “cognitive misers”, their tendency to hillclimb, evidence for display-based difference reduction (Gray, 2000), and the predictions of MAGS (Altmann & Trafton, 2000) all suggest that humans will use whatever means is made available to them and cut corners to reduce cognitive work. Thus, it was predicted that participants would exploit the simple visual cue as a reminder to complete the postcompletion step. Since participants were given extensive training on the system and the cue, they should have formulated an adequate mental model of the system allowing

them to pay attention to the cue at the appropriate time. The cue adhered to recommendations given by Reason (2000) to be conspicuous (at the critical time) and contiguous (proximal), research by Yantis and Jonides (1988) showing onsets to capture attention better than color or intensity, predictions by Altmann and Trafton (2000) regarding just-in time priming from environmental cues, and real-world prominence on existing devices and applications.

It was also hypothesized that feedback in the mode error condition, generated by a downstream error "cost," would help participants remember to complete the postcompletion step on subsequent trials. Mode errors occur when an action is executed in a way appropriate for one mode (or status), when the system is actually in another mode. They can be commonly experienced when leaving a multi-function remote control in the VCR mode and attempting to use it to control the television at a later occasion. With the Tactical task this condition was instantiated by leaving the console at the postcompletion step if it was omitted on a previous trial. In such a case, when the participant returns to the system, it fails to respond until the formerly omitted postcompletion step is first executed. Feedback from the cost in time and points incurred by having to reorient oneself with the awkward state of the system and remember to complete the postcompletion step was expected to cause participants to devote increased attention at the postcompletion step on subsequent trials.

Detailed analyses of practical countermeasures to error and their corresponding human response will lead to a deeper comprehension of human error within human-computer systems and what can realistically be done to offset them. While the most effective solution would presumably be to introduce a forcing function or elimination of the isolated step altogether, as Byrne and Bovair suggest (1997), this is often impossible given the nature of the task. Moreover, there are regularly tradeoffs to be considered, such as the high costs of system redesign and automation bias (Parasuraman et al, 1993; Skitka et al, 1999). Experiment 1 was designed to investigate the effect of one particular

type of common error intervention (reminder) and report its efficacy in reducing postcompletion errors specifically. The additional downstream error cost condition was to assess the effect of negative feedback to the operator on error commission. A combined condition was added to gauge the collective effect of the two interventions.

## 3.2. Method

### *3.2.1. Participants*

Forty-nine undergraduate students from Rice University participated for course credit in a psychology course and additional cash prizes.

### *3.2.2. Materials*

The materials for this experiment consisted of a paper instruction manual (see Appendix A) for each of the three tasks (Navigation, Transporter, and Tactical), Apple iMac computers running the Bridge Officer Qualification application written in Macintosh Common Lisp.

### *3.2.3. Design*

This experiment used a two-factor mixed within and between participants design consisting of the following factors: task assignment and trial at testing. Task assignment consisted of four conditions: control (no intervention version of the Tactical task), cued (cued version of the Tactical task), mode error (mode error version of the Tactical task), and a combined cued/mode error condition. Participants were randomly assigned to one of these four conditions. The second factor, trial, was a within-participants factor that accounted for the period of testing and provided a measurement of error frequency and learning effects over the period of testing.

The primary dependent variable was the frequency of postcompletion errors made during the Tactical task (out of the total number of opportunities). Other measures of

interest included reaction times at the postcompletion step and postcompletion proportion. Postcompletion proportion was a measure of the number of errors at the postcompletion step out of total errors.

### *3.2.4. Procedure*

Participants were run in two sessions, spaced one week apart. The first session served as a training session using written documentation for each of the tasks: Navigation, Tactical, and Transporter. Order of training on the three bridge station tasks was randomized for every participant, as was group assignment. Only the Tactical task contributed to the measure of postcompletion error. Once they successfully completed the training trial and logged three subsequent error-free trials, they were allowed to move on to the next task. Errors resulted in warning beeps and messages, ejected the operator to the main control, and restarted the task. This prevented participants from completing training without having gone through each of the tasks at least four times with all steps done correctly and completely. When training was complete for all three tasks, they were reminded that they would be competing for prizes in one week.

The second session consisted of the test trials for the three bridge station tasks. Participants completed thirteen trials of each task in random order, for a total of thirty-nine trials for the test day. During the second session, the experiment program emitted warning beeps on error commission to warn individuals but did not eject them to the main control as in training. Moreover, warning messages and reminders were removed and trials were continued until the task goal was met.

Participants were encouraged to work both accurately and quickly by means of a scoring system, an onscreen timer, and prizes The scoring system incremented twenty-five points for each correctly executed step and decremented fifty points for each incorrect. Up to 100 points were awarded for task completion within a set time (see Appendix B). For every incorrect working memory recall trial, the score was

decremented 200 points. No points were accumulated for successfully completing a recall trial. The large weight placed on this task was due to an observation made during the previous experiment (Serig, 2001) of participants tending to neglect the working memory task. At the end of each trial, participants were informed of their task completion time, number of errors committed, and score. Accumulated points were used in competition for prizes.

The concurrent working memory letter task was also introduced on the day of testing. As in the studies by Serig (2001) and Byrne and Bovair (1997), its function was to increase working memory load during task performance. Participants were presented with auditory stimuli in the form of randomly ordered letters spoken through the headphones at a rate of one letter every three seconds. A tone was presented randomly at intervals ranging from nine to forty-five seconds upon which the participants were directed to recall the last three letters in order and type them into the text box that appeared on the screen. The text box appeared at random times to control for the increased tendency of omission errors upon task interruption. This was the same for all three conditions.

*Figure 4*. Visual cue at the postcompletion step.


Since the current work is focused on the effect of an automated cue and downstream error cost on postcompletion error, the experiment program allowed participants to complete a trial at testing without executing the postcompletion step (Tactical task), although a warning beep was emitted. The cued version of the same task featured a simple red visual cue appearing adjacent to the “Tracking” button at the

postcompletion step on every trial (see Figure 4). In the mode error condition, the Tactical console stayed at the postcompletion step if it was forgotten on a previous trial and prevented the operator from proceeding until it was first complete. Finally, in the combined mode error and cued condition, the system combined both the downstream error cost of the mode error condition with the visual cue.

### 3.3. Results

Table 2

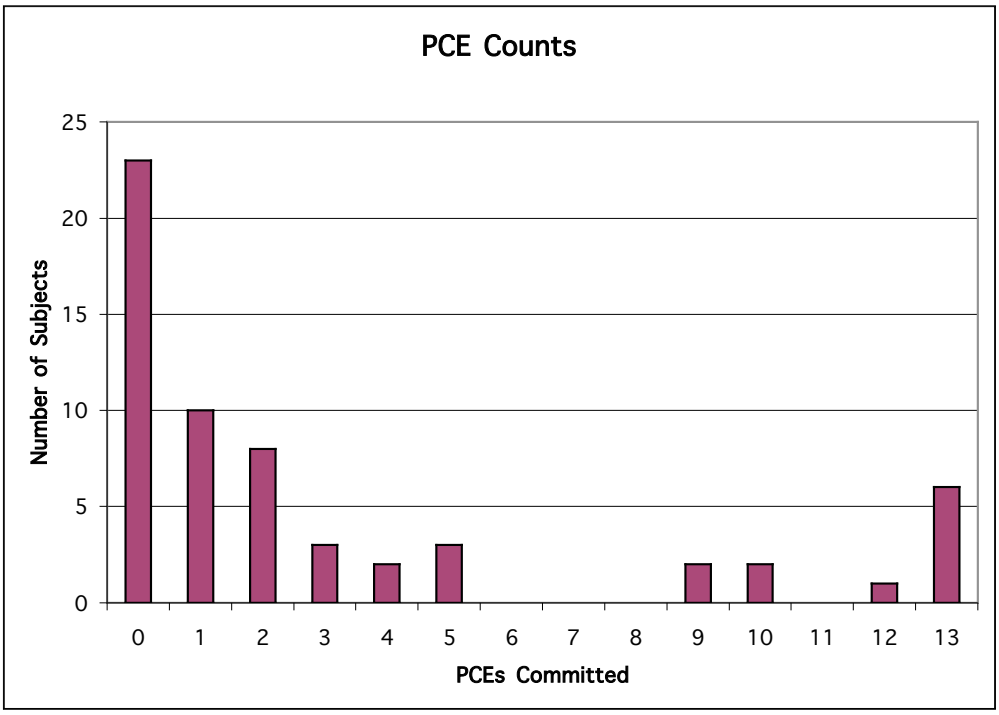
*Participants per condition.*

| Condition | n |
|---|---|
| Control | 11 |
| Cued | 13 |
| Mode | 13 |
| Cue and Mode | 12 |

Analysis of the results focused on the data collected from the testing day. Of primary interest was the mean frequency (out of all trials) of postcompletion errors across groups as well as their proportion (out of all errors). Participants who made an omission at the postcompletion step at more than 50% frequency were removed. This was due to the assumption that postcompletion errors occur when the operator has the correct plan. If errors at this step were shown to occur more than 50% of the trials, it is likely that the participant had not correctly remembered the task completely from training. Fifty percent was chosen as the point of delineation as participants tended to pool into two groups: those with postcompletion errors ranging from 38% frequency and below and those with postcompletion frequency at 70% frequency and above. Thirteen participants were found in the second group and removed: five from the control group, three from the mode error

group, two from the cued group and three from the combined group. This left thirteen participants for the mode error and cued groups, twelve for the combined, and eleven for the control, combining for a total of forty-nine participants included in the final sample (see Table 2 and Figure 5).

*Figure 5.* Postcompletion errors by number of participant across all conditions.

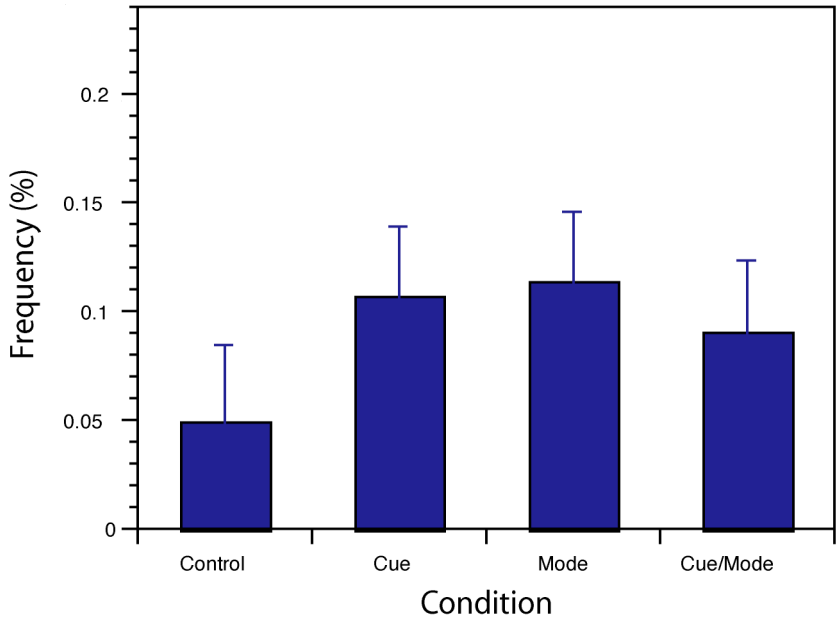



*3.3.1. Postcompletion Frequency*

Postcompletion frequency, or the number of postcompletion errors committed out of the total number of opportune steps (Frequency = Number of Errors at a Step $X_i$ / Total Number of Opportunities for Error at Step $X_i$) was also compared by condition (Figure 6). Between-participants analysis revealed no reliable effect of group on postcompletion error frequency, $F(3, 45) = 1.22$, $p = 0.31$ contrary to the hypothesis. The low means across groups should be noted, as there were likely "floor effects." This may be due to the low number of postcompletion trials and/or the removal of the participants with greater

than 50% postcompletion error frequency. The obtained figures for postcompletion errors were slightly low (e.g., 0.05 versus 0.11 for the Control condition) in contrast to the findings of Serig (2001). A power calculation for the ANOVA shows the effect size to be 0.25, giving us a relatively low power of 0.34 at the .05 level. Power to detect a medium sized effect (i.e., 0.4), however, was 0.61.

*Figure 6*. Postcompletion error frequencies by condition (std. error bars).



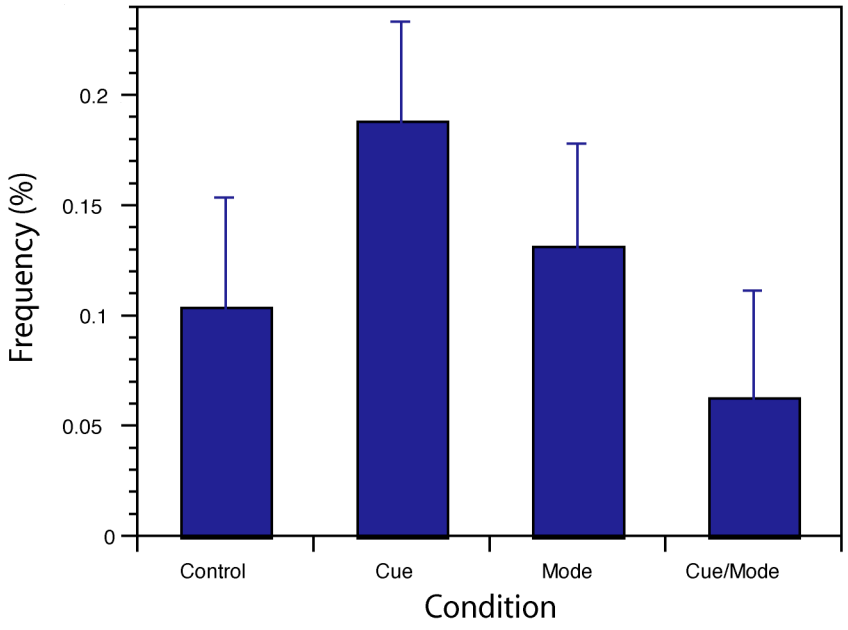

*3.3.2. Postcompletion Proportion*

Postcompletion proportion was a measure of the number of postcompletion errors committed out of the total number of errors at all steps made (Proportion = Number of Errors at a Step $X_i$ / Total Number of Errors in Task X) during the 13 trials of the Tactical task on test day (see Figure 7). Serig (2001) used this measure to show that postcompletion errors occurred at rates higher than predicted by stochastic theory (.083).

However, in contrast to the findings of Serig (2001), the present means are slightly low (e.g., 0.10 versus 0.16 for the Control condition). Not surprisingly, differences between conditions for this measure were also non-significant, $F(3, 45) = .71$,

$p = 0.55$. There were several participants who made a large number of postcompletion errors relative to other errors, particularly in the Control condition. This suggests that, based on the task structure and its isolation within, the postcompletion step was in fact harder to remember.

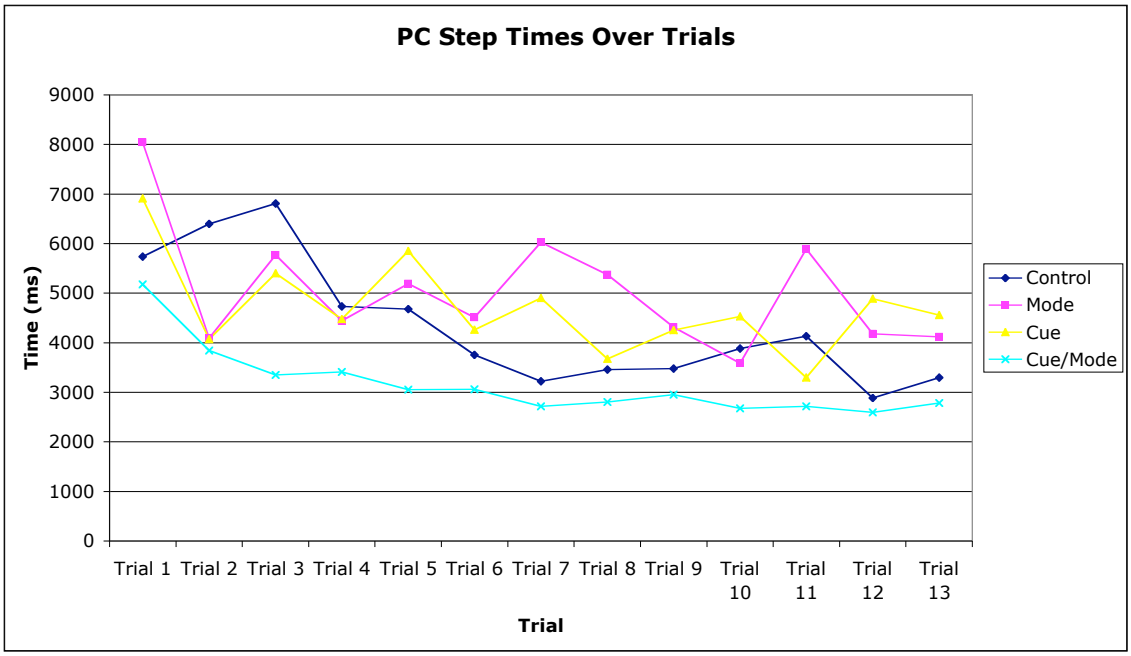
*Figure 7.* Postcompletion error proportions by condition (std. error bars).




### *3.3.3. Postcompletion Step Times*

Participants were run for thirteen trials on each of the three tasks in random order. In debriefing reports, they generally stated that it was not very difficult to recall how to perform the tasks after the first trial. The average postcompletion step times shown in Figure 8 reflect the initial difficulty faced by most of the participants: 5736 ms (Control), 6911 ms (Cued), 8040 ms (Mode), and 5174 ms (Combined). Times declined overall for the rest of the trials, falling within the range found by Serig (~4500 ms, Day 2a; 2001).

*Figure 8.* Mean postcompletion step times over trials by condition.




Due to the substantial variance and skew in the data, a log transformation was applied. A repeated-measures ANOVA revealed a reliable main effect of trial, $F(12, 504) = 7.323$, $p < .01$ and linear trend, $F(1, 31) = 28.15$, $p < .01$. There was also a reliable main effect of condition, $F(3, 42) = 8.23$, $p < .01$, although Tukey's HSD test showed none of the intervention conditions to be significantly different from the control, $p > .05$. Finally, the trial by condition interaction was not significant, $F(36, 504) = 1.23$, $p = .169$.

### 3.4. Discussion

Overall, the results did not support the proposed hypotheses concerning the two treatments. Neither task completion time nor the number of postcompletion errors was reliably different for the cued, mode error, or combined treatments. The expected learning effects were found, however, as time decreased significantly with subsequent trials. Differences in overall task completion times (Tactical) were not reliable between conditions. This suggests that neither the cue nor the downstream error cost or a combination of the two had any substantial effect on task speed. However, given that the

structure of the Tactical task itself, apart from what occurred at the postcompletion step, was identical for all three conditions, it does not seem so out of order that the overall trend was similar across groups.

It should be noted, however, that the mode error condition incurred additional time costs as participants were often slowed at the initial step on subsequent trials following a postcompletion error. Participants were often initially puzzled by the state of the system upon return after postcompletion error commission on a previous Tactical trial. Since, in that condition the Tactical system was left at the point of the postcompletion step if omitted, they were required to complete the step of hitting the "Tracking" button before proceeding back to the first step of the task. Nonetheless, this feedback did not seem to change participants' behavior or decrease their rate of error.

### *3.4.1. Failed Postcompletion Step Recall*

Consideration must be given to the ten participants who were unable to recall the postcompletion step at above 50% frequency. The fact that ten of the initial forty-nine participants were unable to recall this task step is somewhat remarkable considering that they had all completed the extensive training session successfully. Nevertheless, despite a slightly positive relationship between the total number postcompletion errors and other errors, several of these participants with high numbers of postcompletion errors made relatively few other errors. This supports the notion that the postcompletion step is particularly difficult to remember due to the task characteristics provided by Reason earlier in this document.

In subjective reports, participants, including those who committed the error at over 50% frequency, reported that the task as a whole was generally not difficult to remember. However, some of those who did commit postcompletion errors at over 50% frequency stated that the task seemed to change or that they didn't understand why the

program kept beeping at the last step. Again this suggests that they had particular problems recalling the postcompletion step.

Although not markedly different possibly due to the sample size, there was a difference in the number of above 50% frequency participants omitted by condition: five from the control group, three from the mode error group, two from the cued group and three from the combined group. The control condition, without any particular feedback at the postcompletion step, generally left more participants who did not remember the postcompletion step (at above 50% frequency) at testing followed by the mode error, combined, and cued conditions. While the data from the current work is inconclusive, it seems possible that further testing on this measure may reveal general differences in task retention between conditions.

*3.4.2. Review and Explanations*

The work presented here was intended to examine the effects of a simple visual cue and a downstream error cost on postcompletion error commission. Although human error within man-machine systems has been studied for quite some time, the best that the literature seems to offer are general taxonomies of human error and task performance with little predictive value. Various aids, reminders, and interventions have been suggested and implemented in real-world settings (Reason, 1990; 1997), but low level psychological inquiries of how these work and comparisons of which work better in particular situations are lacking. The earlier study by Byrne and Bovair (1997) demonstrated that human error could be more critically assessed at the cognitive level, finding the influence of working memory load on one common procedural error, postcompletion error. The current inquiry was an attempt to take this critical approach further in a comparative study of two types of error interventions.

Although the effects of both conditions in the current study were not found to cause significant change in reaction times or error commission at the postcompletion

step, the findings do present some valuable implications. First, the fact that the visual cue did not significantly reduce the number of postcompletion errors committed by the participants suggests that, at least in this type of computer-based task, the cue was not salient or informative enough to be of significant aid for the sample group tested. While for the experimenter, the sudden onset of a large red dot next to the button that needed to be pressed on a black and white console at the postcompletion step seemed intuitive and informative enough, participants made omissions regardless. Seemingly, they were overlooking the cue or forgetting its correlation with the action of pressing the "Tracking" button at the postcompletion step.

Initially, it was considered that this might be attributed to participants' lack of understanding about the system or scenario (Starfleet Control). However, Rouse (as cited in Sheridan, 1997) states that the evidence does not support the idea "that diagnosis of the unfamiliar requires theory and understanding system principles." Moreover, participants could not have made it to testing without first passing an extensive training session where the system was described to them in full detail. Most likely, for participants who exhibited correct knowledge of the plan (sub-50% postcompletion frequency), their attention was simply distracted at the postcompletion step. Critical observation of the system's state should have caused them to wonder about the novel appearance of the visual cue next to the "Tracking" button. However, a speed-accuracy tradeoff, coupled with the motivation generated by the time and performance pressures, may have allowed omissions to take place despite the cue.

Neither did the downstream error cost present a significant change in behavior at the postcompletion step for participants. While it was expected that the cue would be more effective between the two treatments, it was also hypothesized that this downstream error cost would generate negative feedback to the user to bring about a change in behavior. However, as shown in the results, the number of postcompletion errors in this condition was not significantly different from the control group, suggesting that it did not

provide any significant advantage (or disadvantage) for the participants. This perhaps follows findings by Serig (2001) that demonstrated participants' error commission to be relatively independent of negative or positive feedback.

There are some qualifications that go along with the findings. It is quite possible that the lack of significant differences is due to the few number of postcompletion trials or opportunities for error (thirteen). Additionally, it should be noted that several participants failed to recall the postcompletion step altogether. Since the system no longer offered error messages telling the participant what the correct action was at each step, it is likely that as trials progressed with no strong negative or informative feedback given by the system, the participants continued through the trials uncaring or unaware of their mistake. Driven by the performance and time pressures, it is likely that they failed (or chose not to) to stop and consider their incorrect actions or try to recall the meaning of the interventions. As participants proceeded through the thirteen Tactical trials with the incorrect plan (a mistake) and that particular rule gained strength, it would have become increasingly difficult for the cue or the mode error to generate any influence on the participant's behavior.

One final possible explanation is that the interventions lacked salience or participants forgot the association or rule. However, literature on task training and transfer indicates that frequent hands-on experience with an accurate simulator is the best way for an operator to develop a strong mental model of the process (e.g., Detterman, 1999; Sheridan, 1997). Participants were required to read a manual detailing the association of the cue to the postcompletion step (see Appendix A) and were not allowed to quit the training session until they could complete three further trials on their own without the manual. Evidence from research on warnings and reminders, however, suggests that they should be novel and constantly updated to capture attention sufficiently (Laughery & Wogalter, 1997; Reason, 1997). Whether this is a problem of training, interface design, or cue remains to be seen.

### *3.4.3. Implications*

According to Byrne and Bovair (1997), the commission of postcompletion error is so reliable under high working memory load that they suggest the only real way around is to design it out. Other possibilities that have been commonly suggested and implemented include forcing functions that "force" the user to complete an otherwise potentially omitted step in order to achieve the main task goal. Serig (2001) found that participants given a "forced" Tactical version of the task, where the postcompletion step must be completed to obtain a completion signal (target destroyed cue), made postcompletion errors at close to zero frequency. Automated teller machines initially faced the same sort of problem as the initial auditory and visual reminders implemented in early models failed to successfully remind customers to withdraw their card. As a result, many models today now feature a forcing function that prevents the user from proceeding with a transaction before the card is withdrawn. For the initial gas cap example, newer cars now have caps that are somehow tied to the car.

Reason (2000) offers several different suggestions for reminders as alternatives to forcing functions and more expensive system redesigns that, with some tasks, are not quite possible. In a short survey based study, comparisons were made between reminder strategies, similar to how remediation techniques were evaluated in the current experiment, using a constructed postcompletion error. In contrast, however, the reports provided by Reason (2002) were based on subjective ratings and were not undertaken with computer-based tasks.

### *3.4.4. Further Inquiry*

According to supporting theories and suggestions (e.g., Altmann & Trafton, 2000; Reason, 1997), quick and practical interventions may be found for computer-based tasks that are effective at reducing postcompletion error and errors in general. Further inquiry

was again made using tasks with constructed postcompletion steps, since they generate omission errors predictability and can be accounted for theoretically. To consider the effects across tasks, a new computer-based task was designed and compared with the original used in Experiment 1 to pinpoint differences. Only through such controlled, empirical comparisons and probing of human performance using a known procedural error will predictive, specific, and useful information for error intervention be revealed.

## 4. EXPERIMENT 2

The purpose of Experiment 2 was twofold: first, correct the shortcomings of Experiment 1 and second, expand the breadth of inquiry. Issues with training, train-test delay length, salience of our interventions and strength of association, and the number of trials at testing were considered in devising a new study on routine procedural errors and their possible interventions.

### 4.1. Mode Awareness and Training Issues

It has been hypothesized for quite some time that people use mental models to predict system behavior and guide actions (Norman, 1983). Consequently, training in the previous experiment and in the work by Serig (2001) was thorough and detailed. However, as noted, there were still problems in Experiment 1 with several participants who failed to recall the postcompletion step at greater than 70% frequency at testing. Perhaps due to inadequate training or too long of a train-test delay, it is possible that participants had an inadequate mental model of the system (or at least the Tracking system) and hence the postcompletion step was not immediately apparent.

With a gas cap on a car, despite lack of experience, the physical presence and design of the cap may act as a cue to remind a person to later replace it (system maintenance) after completion of the main goal (filling up the car). The same may be true with the removal of originals from a photocopier or a banking card from an ATM.

However, with the task used in Experiment 1 there were several participants who showed dramatically incomplete recall at testing suggesting a lack of knowledge and not a simple slip of mind or behavior. To them it was not obvious that the "Tracking" button needed to be pressed a second time for system maintenance. Hence, the association between the system change and maintenance (postcompletion) steps was further emphasized in Experiment 2. To do this, four main changes were made:

1. The training manuals were revised to promote a stronger mental model of the system for participants.
2. The delay between training and testing was reduced to two days.
3. Participants were quizzed at the end of training to ensure that they associated the cue or mode indicator and the postcompletion step (e.g., If you see the red blinking arrows…?).
4. The system reminded participants of the postcompletion step on the first trial at testing if they forgot.

Finally, in selecting cues and revising the training materials, the issue of inert knowledge was addressed: just because people possess the relevant knowledge (through training in this case), it does not guarantee that it will be activated when needed, given the conditions in which the task is performed (Woods & Cook, 1999). Even additional effort may be futile as Serig (2001) found that adding additional social incentives at training had little effect in improving participants' performance at testing. Findings from the problem solving literature show that people fail to apply a recently learned problem solving strategy to an isomorphic problem unless explicitly prompted (Gick & Holyoak, 1980). The importance of having cues and information available "just-in-time" is well documented and accepted in other domains as well. Thus, with the working memory load, time, and performance pressures, a successful system interface should provide salient cues to successfully prompt the required action and information at the right time and place in the given context.

## 4.2. Intervention Implementation

Experiment 2 introduced two interventions: an enhanced visual cue and a mode indicator. These were developed specifically to address issues brought up by the findings from Experiment 1. Two separate and somewhat different interventions were employed to help pinpoint the characteristics of a successful intervention, should one or both have a significant effect.

### *4.2.1. An Enhanced Visual Cue*

It is well known that the selection or noting of visual cues or features is automatic (e.g., Treisman, 1988). Automatic processing of novel peripheral cues regardless of whether or not they are informative has been well documented in the literature (e.g., Jonides, 1981; Remington, Johnston, & Yantis, 1992). Other research has also revealed that even the addition of a simple visual cue (e.g., an orange dot) can bring about changes in how people interact with physical objects such as doors (Wallace & Huffman, 1990). Colorful visual cues are known to be effective and necessary in guiding individuals to select points of activity, such as the push plate on a door (Norman, 1988). The simple red onset used in Experiment 1, however, was not an effective reminder.

Hollnagel (1993, p.299) explains that the strength of a cue is relative to its specificity and thus it is the relative not absolute strength that is important when assessing a cue's potential as a reminder. This assertion is governed by the fact that when a task is considered trivial, attention is more easily diverted. Performance becomes controlled by more error-prone generic functions such as "look for cue which indicates a turn," rather than exact intentions such as "look for cue-X, then turn to the right." For this reason, it was important to design and train participants to visually specific cues demanding explicit actions rather than generic ones affording a wider range of meaning. Such generic cues can potentially lead to description errors, or errors caused by multiple cues with ambiguous specification of the required action or state, if used within complex

systems (Norman, 1983). This may explain the ineffectiveness of the cue in the previous experiment.

In a study by Monk (1986), auditory cues were used to drastically reduce the occurrence of mode errors. Keying-contingent sounds were used on a keyboard-based computer game to enhance feedback and draw the user's attention to a change in the system's mode. This worked well because the nature of mode errors is such that they generally occur when the user is unaware of the system's current mode and its consequences. Monk (1986) observes that display changes, however, are effective when the user is required to look at the relevant parts of the display at the appropriate moment in the dialogue. Mechanical pointing devices such as the mouse force users to focus on the screen, making small visual changes or cues, which may go unnoticed with other types of interaction, more likely to be effective.

### *4.2.2. Cue Attributes*

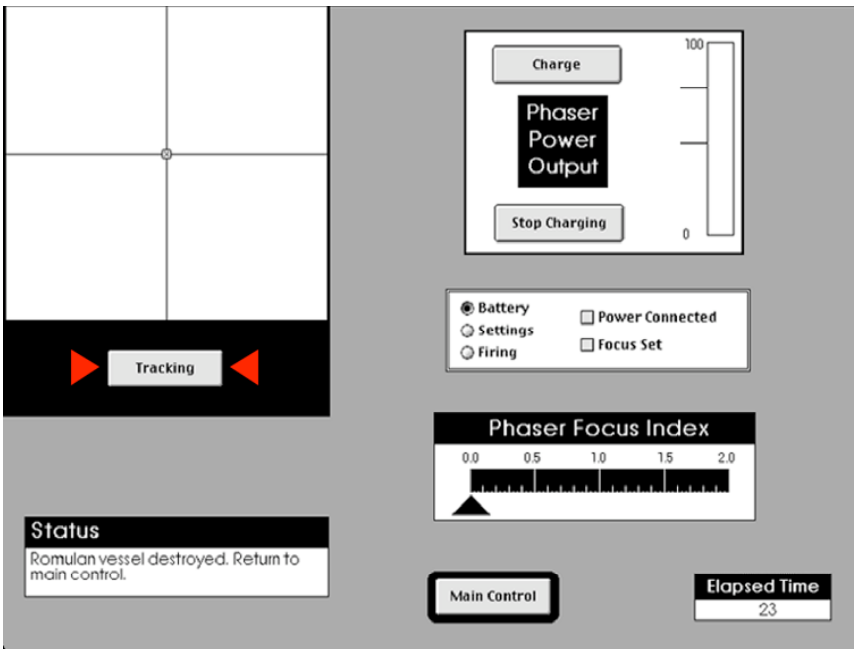
There is much research (Sutcliffe, 1995) suggesting that the visual attributes most effective for attracting attention (warnings and indicators) on a computer interface in order are as follows:

1. Movement (blinking or change of position)
2. Shape and Size (character font, shape of symbols, text size, size of symbols)
3. Color
4. Brightness
5. Shading and Texture (different texture or pattern)
6. Surroundings (borders, background color)

Sutcliffe (1995) suggests that care be taken to ensure that the user population interprets the warning icon as the designer expects. Furthermore, such techniques should only be used sparingly, as the presence of many conflicting stimuli can essentially dull their individual effectiveness.

For color, red, green, and yellow are recommended as status indicators, each corresponding to its meaning on a traffic light. To draw attention, white, yellow, and red are most effective, although yellow offers the best visibility. Based on these recommendations from the literature and the failure of the cue in Experiment 1 to reduce postcompletion errors against the control condition, alternating red and yellow blinking arrows (see Figure 10) were used in both the Tactical task and the new Medical task. Exact placement of the cue was determined through pilot studies to ensure that people associated the cue with the required step.

*Figure 10.* Two blinking (red and yellow) arrows.

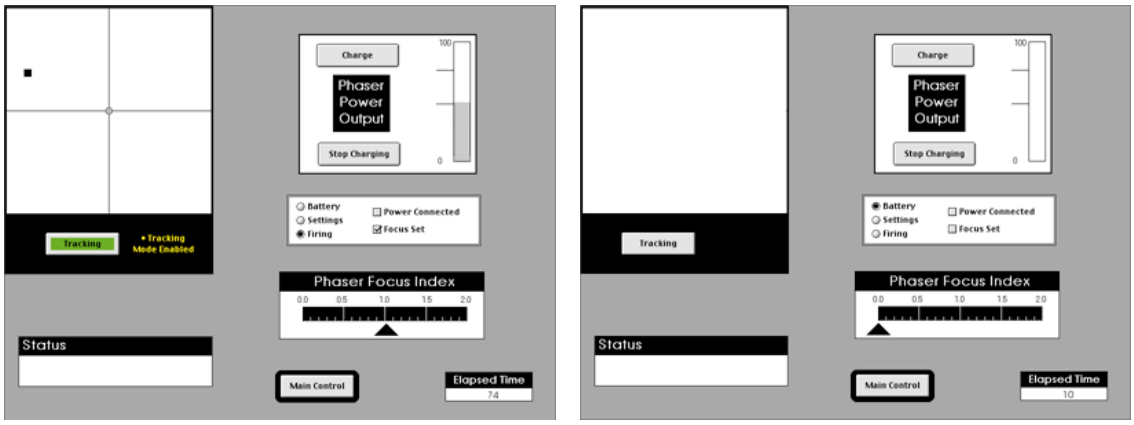



### *4.2.3. A Mode Indicator*

The previously used Tactical interface of the Bridge Officer Qualification program was redesigned for a separate mode indicator condition in which visibility of the state change was enhanced. As shown in Figure 11, the mode indicator consisted of a

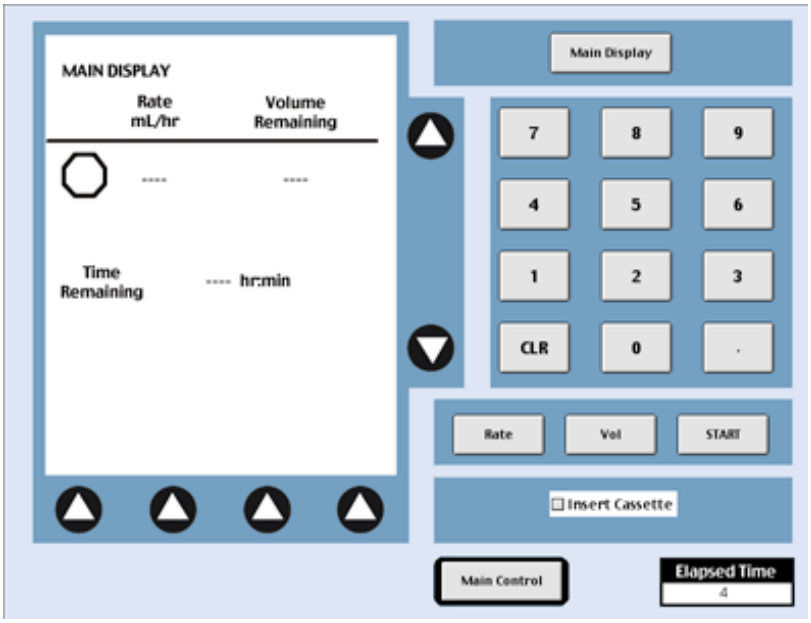
green light on the “Tracking” button, the appearance of crosshairs in the targeting window, and the message “Tracking Mode Enabled.” It was thought that the combination of these three novel features would be sufficient to indicate to the user that the system was in a separate Tracking mode.

*Figure 11.* Mode Indicator in the Tactical task in on (left) and off (right) states.




The Chief Medical Officer Qualification program (Figure 12) was also similarly developed in all three conditions (control, cued, and mode) with the cue and mode indicator appearing at the “Main Display” button instead of at “Tracking”.

*Figure 12.* Medical console.



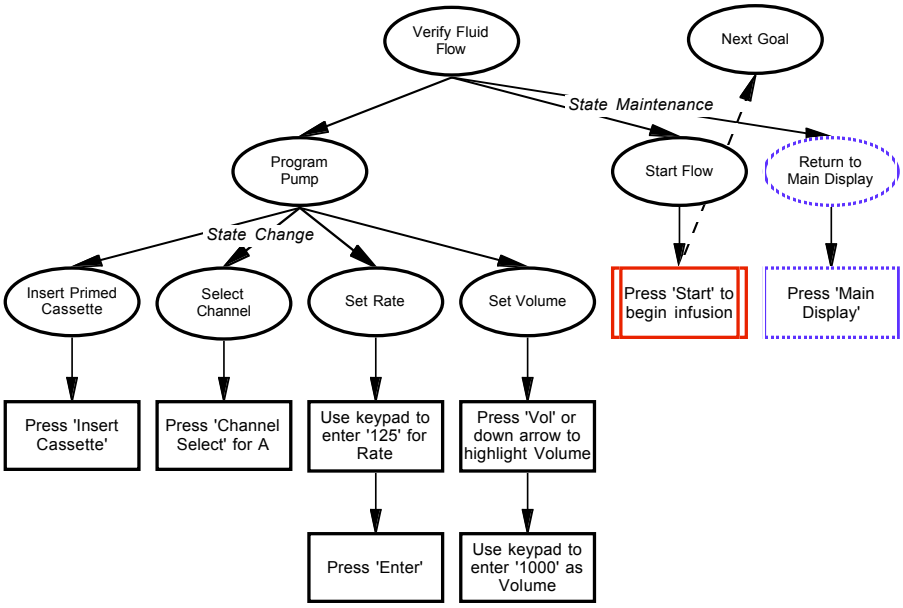

The intention of the mode indicator condition was to provide contextual cues on the interface that convey system that the Tracking system is turned on. When combined with the given if-then rule at training (i.e., "If you see a mode indicator light the system is on"), the mode indicator implies the necessary corresponding action of turning off the Tracking system (system maintenance). Once the participant finishes the intermediary steps and hits the "Tracking" button a second time, the green blinking turns off to indicate that the Tracking mode has ended. This type of mode indicator is quite common in real-world devices. As Monk found (1986) in his study, enhancing feedback with an auditory cue to signal a change in the system's mode decreased the frequency of errors made by participants.

### *4.2.4. Two Postcompletion Tasks*

In Experiment 2 the same two interventions were tested both on the Tactical system used in Experiment 1 and a new (postcompletion) Medical system designed for the purposes of this study. This was to look at any differences in the efficacy of the

interventions across interfaces and tasks. The new system was tied in with the old scenario of the Star Trek Bridge Officer Qualification course under the title: Chief Medical Officer Qualification course. The system and task have real-world applicability, having been taken from a real-world medical infusion pump scenario and interface design. The hierarchical structure of the new task is shown in Figure 13.

*Figure 13.* Medical task hierarchy (postcompletion).




## 4.3. Method

### *4.3.1. Participants*

Ninety-one undergraduate students from Rice University aged 18 to 35 participated for course credit in a psychology course and additional cash prizes ranging from $10 to $40.

### *4.3.2. Materials*

The materials for this experiment consisted of a paper instruction manual for each of the four tasks (Navigation, Transporter, Tactical, and Medical), paper quizzes for the

first day, Apple iMac computers running the Bridge Officer Qualification and Chief Medical Officer Qualification applications written in Macintosh Common Lisp, and a web-based general questionnaire.

### *4.3.3. Design*

Experiment 2 used a two-factor between participants design with two independent variables, task and intervention. Task consisted of two conditions: Bridge Officer and Chief Medical Officer. This was to look at differences in the effectiveness of the interventions across tasks. Intervention consisted of three conditions: control (no intervention), visual cue (alternating red and yellow blinking arrows), and mode indicator (mode indication for the system state change). Participants were randomly assigned to one of the six groups.

The primary dependent measure was the number of postcompletion errors made during the Tactical and Medical tasks. Other dependent measures of interest included response times at the postcompletion step and the overall number of errors per task.

### *4.3.4. Procedure*

Similar to Experiment 1, participants were run in two sessions, although the spacing between them was reduced to two days to help address problems with recall of the tasks. The first session served as a training session using written documentation for each of the tasks (Navigation, Medical or Tactical, and Transporter; Appendix D). Order of training was randomized for every participant. Once participants successfully completed the training trial with the manual and logged three subsequent error-free trials, they were asked to move on to the next task. Errors resulted in warning beeps and messages and participants were returned to the main control to restart the task. This was to prevent participants from completing training without having gone through each of the tasks at least four times with all steps done correctly and completely. When training was

complete, they were reminded that they would be tested for prizes in two days and given a short quiz (Appendix C) to ensure that they had an accurate mental model.

The second session consisted of the test trials for both the Tactical and Medical tasks. In random order, participants completed seventeen trials of their assigned postcompletion task (Tactical or Medical) and 11 trials for each of the two dummy tasks (Navigation and Transporter) for a total of 39 trials for the test day. The number of trials for the postcompletion task was increased from 13 in the first experiment to provide greater power. At testing, the experiment program emitted warning beeps on error commission to warn individuals but did not immediately return them to the main control as in training. Participants were encouraged to work both accurately and quickly by means of a scoring system (Appendix E), prizes, and an onscreen timer as in the first experiment (see section 3.1 for details). The same auditory working memory task from the previous experiments was also used in the current work for all task conditions at testing.

## 4.4. Results

A total of 91 participants began this experiment, but data from only 82 were kept in the final analysis. The primary reason for this loss of data was participant failure to show up at their assigned testing date, but there were also a few cases of lost and incorrectly saved data files and one participant was removed as an outlier (Medical, cue condition). Groups broke down as shown in Table 3 below.

Table 3

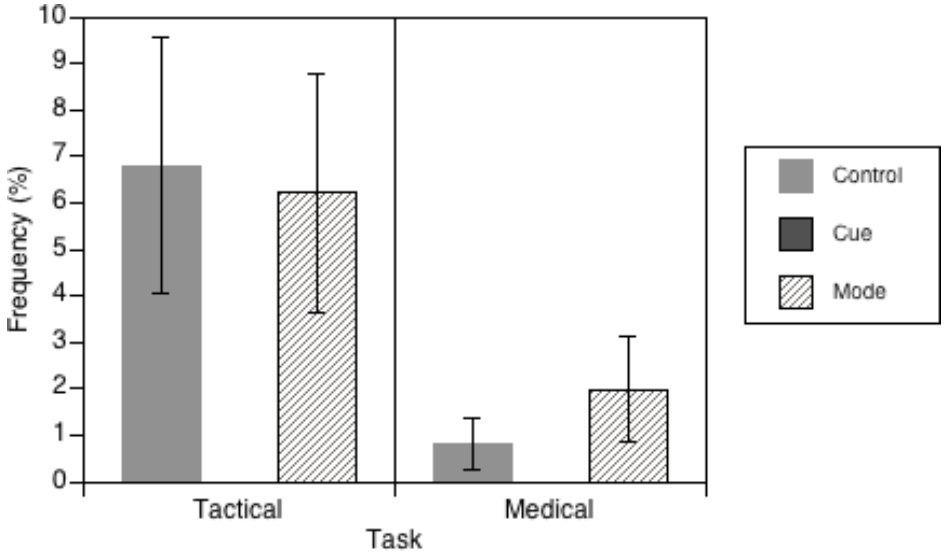
*Experiment 2 descriptives.*

| | | *Tactical* | | | *Medical* | | |
|---|---|---|---|---|---|---|---|
| | | **Control** | **Cue** | **Mode** | **Control** | **Cue** | **Mode** |
| | **N** | 14 | 16 | 13 | 14 | 12 | 13 |
| *PCE frequency* | **M** | 6.81% | 0% | 6.21% | 0.82% | 0% | 1.99% |
| | **SD** | 10.34 | 0 | 9.22 | 2.08 | 0 | 4.14 |
| *PC step times* | **M** | 4365.46 | 3512.23 | 4627.68 | 1077.67 | 984.98 | 1096.58 |
| | **SD** | 1075.95 | 540.78 | 1520.78 | 225.9 | 221.72 | 245.7 |
| *Total errors* | **M** | 0.54 | 0.65 | 0.83 | 0.18 | 0.27 | 0.4 |
| | **SD** | 0.39 | 0.75 | 0.58 | 0.14 | 0.24 | 0.22 |
| *WM accuracy* | **M** | 49.90% | 50.36% | 45.25% | 38.95% | 45.92% | 33.22% |
| | **SD** | 25.39 | 23.07 | 23.14 | 18.01 | 22.56 | 19.63 |

### *4.4.1. Postcompletion Errors*

Our primary measure of interest was the frequency of errors at the postcompletion step in both tasks. This is the step immediately following completion of the main task goal. In contrast to Experiment 1, there were no participants with greater than 50% postcompletion error frequency, suggesting that the correct knowledge was imparted and carried over to testing day.

*Figure 14.* PCE frequency by condition and task (std. error bars).




Note. Cue is 0% in both tasks.

For the Tactical task, mean postcompletion error frequencies were 6.81%, 0%, and 6.21% for the control, cued, and mode indicator conditions, respectively (Figure 14). Analysis of variance showed the effect of intervention to be reliable, $F(2, 76) = 4.061$, $p = .021$, but not the interaction of intervention by task, $F(2, 76) = 1.86$, $p = .162$. Planned comparisons confirmed our hypothesis, as participants made significantly less errors in the cued condition versus the control, $t(76) = 3.14$, p = .002, and even versus the mode indicator group, $t(76) = 2.81$, $p = .006$. In comparison, the mode indicator failed to produce reliable differences with the control group, $t(76) = .263$, $p = .793$.

In the simpler Medical task, mean errors at the postcompletion step were very low: 0.82%, 0%, and 1.99% for the control, cue and mode indicator conditions, respectively. Again, none of the twelve participants in the Medical cued condition made a single postcompletion error in all seventeen of their trials. The same planned comparisons done on the Tactical task revealed no reliable differences across intervention and task.

*4.4.2. Postcompletion Step Times*

Outliers greater or less than three standard deviations from each participant's mean were removed and replaced with their mean. Whether due to the number or nature of the steps, the mean reaction time at the postcompletion step was drastically shorter in the Medical task. An ANOVA showed this effect of task to be reliable, $F(1, 76) = 305.41$, $p < .001$. The effect of condition was also reliable, $F(2, 76) = 4.32$, $p = .017$. However, the condition by task interaction was not, $F(2, 76) = 2.86$, $p = .63$.

*4.4.3. Total Errors*

The average number of total errors (out of all possible steps) was found to be higher for the Tactical task than the Medical: 0.67 in the Tactical versus 0.28 in the simpler Medical task, $F(1, 76) = 14.60$, $p < .001$. Differences across intervention were not reliable, $F(2, 76) = 2.24$, $p = .113$, although it should be noted that the total number of errors was slightly higher for both the cue and mode indicator conditions in both tasks.

*4.4.4. Working Memory Task*

Participants showed no reliable differences in working memory task performance regardless of task $F(1, 76) = 3.47$, $p = .07$ or intervention, $F(2, 76) = 1.09$, $p = .342$.

## 4.5. Discussion

Our findings generally corroborate our hypothesis for the visual cue, but not for the mode indicator. As reported, all 16 participants in the cued condition of the Tactical task exhibited error-free performance at the postcompletion step. In contrast, the control and mode indicator groups showed mean postcompletion error frequencies between six and seven percent Given the lack of reliable differences across intervention for overall

error rates and performance on the working memory task, there seems to be no reason not to attribute the difference in postcompletion error frequency to the success of the intervention.

Nevertheless, our expectations for the mode indicator were not met. In fact, the mode indicator condition in the Medical task showed the highest postcompletion error rate. Whether this means mode indicators in general are ineffective as reminders in the real world is unclear. However, in the given conditions of the experiment, the "just-in-time" visual cue reduced the postcompletion error mean to nil, whereas the mode indicator had hardly any effect relative to the control. This was despite the fact that all participants were given equal training. To ensure visibility, the mode indicator was made green and as large on the screen, if not larger, than the flashing arrows in the cued condition. With the additional novel appearance of the crosshairs (Tactical task) and display information (Medical task), the state change should have been noticeable to the participant at the right time. Thus, its failure does not seem attributable to a lack of knowledge or relative visibility.

It is significant to note that our findings from Experiment 2 and Experiment 1 fell in line with the predictions of Altmann and Trafton's MAGS (2002). This is assuming that the cue in the first experiment failed because it was not salient or seen. Just as they claimed, a salient cue (Experiment 2) was sufficient to prime the postcompletion step, making it unnecessary to place the postcompletion action on the critical path. Moreover, the mode indicator ("reminders at the start") was not helpful for the postcompletion step which followed. As suggested by their explanation, it was likely "masked" by intermediate goals. Finally, our just-in-time visual cue (priming) was the only reliable reminder, consistent with their last prediction.

### *4.5.1. Medical Task*

The Medical task was not effective as a parallel of the Tactical task, possibly due to its substantially shorter length. It took participants nearly one quarter of the time taken to finish the Tactical task and simply failed to generate sufficient error rates to prove useful for comparing the effects of the interventions. Interestingly, though, the same cue that completely eliminated errors in the Tactical task at the postcompletion step also eliminated errors in the Medical task (versus .82% and 1.99% for the control and mode indicator, respectively).

There are several possible explanations for the disparity in error rates. First, due to intrinsic differences in the nature of the task, the assumed postcompletion step may not have been a postcompletion step at all. In fact, it was found that the *last* step of the task (return to Main Control) generated more errors than the supposed postcompletion step in the control condition. This may be tied to a second possibility, which is that the shorter Medical task had a much simpler goal structure. The longer and more complex Tactical task requires the application of additional If-Then operators to complete the task and has more transitions between related "clusters" of buttons and steps on the interface.

The Medical task also clearly illustrates the difficulty of designing a task that can elicit sufficient error rates from participants to study human error in the laboratory. The Tactical task, in contrast, has been successful at generating routine procedural errors in the lab for Serig (2001), Byrne and Bovair (1997), and the present work. This demonstrates the difficulty of predicting what specific factors in a situation cause postcompletion errors to occur. Although the Medical task included a task step (press "Main Display") after the main goal of the task (program and run an infusion), this supposed postcompletion step failed to generate significant error rates. There are implications here for those doing formal evaluations of systems (e.g., heuristic evaluation), since it demonstrates the complexity of identifying potential error inducing steps or features of an interface. Simply conducting a task analysis or classifying errors using a taxonomy will not always be enough.

### *4.5.2. Further Questions*

The most obvious question that surfaces from our findings is why the cue (flashing arrows) worked in this experiment, whereas the mode indicator (text and graphical change) and the cue in the previous experiment (red onset) did not. Was it simply more visually prominent? The mode indicator seems to be equally, if not more, visible, with changes being made on the interface in two areas. Was it the positioning or timing? The cue in the Experiment 1 came in at the same time and place in the task (adjacent to the "Tracking" button).

If Sutcliffe (1995) is correct, then movement is the strongest visual attribute in determining whether or not a cue or warning is seen on the computer interface followed by shape and size. Blinking arrows pointing at the correct button added both rapid movement and an obvious directional meaning to the cue used in Experiment 1: a simple red dot appearing next to the button. However, the visual systems of cognitive architectures such as ACT-R are not currently hardwired to make accurate predictions on their own of whether or not a reminder will be reliably noticed simply based on such characteristics. Neither can such differences be accounted for by current textbook taxonomies, heuristics, or task analyses often used to evaluate interfaces. These issues will have implications later in modeling this task and in further study.

## 5. A COMPUTATIONAL MODEL

The interaction between the related theories of error and cognitive components can be well expressed using a computer-based architecture that embodies many of the relevant constraints on human cognition. These may effectively prove more useful than textbook taxonomies or heuristics when it comes to accurate error prediction. One such

architecture is ACT-R (Anderson & Lebiére, 1998; Anderson et al., in press), which has previously been employed to model human error (e.g., Lebiére, Anderson, & Reder, 1994). Although both Experiment 1 and Experiment 2 had two separate manipulations, the present model will be focused on the cued and control conditions from Experiment 2.

The main independent variable in this line of work has been the intervention. With the task under consideration, the working memory load imposed by the digit span task has also been found to affect performance and increase postcompletion error frequency. Byrne and Bovair's (1997) conjecture and results are also in keeping with general findings of task performance degradation under situations of high cognitive load (e.g., Ruffel-Smith, 1979). The model takes these conditions into consideration as well. It does not, however, account for skill learning, which occured at the beginning of the experiments. At this stage, errors of commission due to lack of knowledge were far more prevalent (knowledge-based mistakes).

### 5.1. Error Modeling Traditions

Traditionally, *symbolic* systems have modeled consistent errors and errors of commission by assuming certain rules fail to apply (Van Lehn, 1989). Symbolic systems have more difficulty, however, with occasional slips or intrusions (Norman, 1981). On the other hand, *connectionist* models, with their holistic computation style, can reproduce human-like errors and graceful degradation of performance under noise or component failures. However, they have been unable to scale up to computer-based tasks like that used here. ACT-R, being a *hybrid* system, is better suited than traditional symbolic systems in this case, because activation of chunks is spread through a connection network (e.g., Altmann & Trafton, 2000).

Since the ACT-R theory proposes a limit on source activation that is divided between tasks, it is well-suited to model the paradigm at hand. By taking advantage of this construct, potential errors at each step in the task structure can be introduced. As

shown in previous work (Byrne & Bovair, 1997; Serig, 2001), postcompletion errors occur reliably with increased working memory load, which, in its model representation, "steal" activation required to make a retrieval of the postcompletion subgoal.

Anderson, Lebiére, & Reder (1994) have already demonstrated ACT-R's capacity to model human error, traditionally considered a domain restricted to connectionist models. In their study, participants were asked to perform a high-level cognitive task of solving simple linear algebra problems while memorizing a digit span, in similar fashion to the work at hand. Using ACT-R they reproduced errors of omission by introducing a cutoff on the latency of memory retrievals – retrievals fail if a chunk does not have sufficient activation. By adding Gaussian noise to chunk activation, they were able to generate a pattern of error quantitatively similar to participant data. With the current paradigm, a similar method was utilized to model errors of omission occurring at the post-completion step.

## 5.2. Model Specifications

The model was built only to perform the Tactical task. Although it has many abstractions, particularly in the non-postcompletion steps, the focus was the postcompletion step itself. There is an assumption here of medium to high skill level and familiarity with the task. Yet it seems rather unlikely that, no matter how familiar the person is with the task, the exact location of the required button would be remembered down to the pixel level. Thus, visual attention was directed to the button at each step using a range rather than specific coordinates with little lag for visual search. The model in its current form also does not explicitly account for the hierarchical nature of the task (goal structure). Hence, the longer delays found in participant data where the task demands movements across "clusters" of widgets on the interface are absent.

There are two key productions in this model related to the postcompletion step. The first of these initiates a retrieval of the postcompletion step information, which

includes the visual coordinates for the next step. Using partial-matching in ACT-R, the level of "similarity" between the two knowledge chunks for the postcompletion step and the last step can be adjusted. The additional working memory load is simulated using dummy chunks representing state information placed in the goal buffer. These chunks, which can be considered as the digits in the digit span task, "steal" activation available for the retrieval of the chunk that produces the postcompletion action. By adding activation noise, random retrievals of the incorrect (last) step are generated, leading to postcompletion errors. This leads to simulated postcompletion errors, where any retrieval of a functionally isolated step (Reason, 1997) is less likely to be made. In the cue condition, a second production automatically "stuffs" the cue into the visual system on appearance which, in turn, triggers the retrieval of the correct knowledge chunk for the postcompletion step. In this case the automatic capture of visual attention by the cue eliminates potential errors.

The postcompletion frequency found in Experiment 1 for the control condition was 4.9%. However, this was only after participants who committed the error with over 50% frequency were removed in adherance to the definition of postcompletion errors as errors occurring at high skill levels. Postcompletion frequency was at nearly 25% with their data included. In Experiment 2 the baseline postcompletion frequency was slightly lower, although this time participants with 25%+ frequency were absent. Thus, as a compromise, 5-15% was the target postcompletion error frequency for this model. This seems reasonable considering that the means for the cued and downstream error groups' participants exhibited postcompletion errors at around 10% and the trend in the data showed these groups to be better overall, although not significantly.

Running the model for 200 trials generated a 14.1% postcompletion error frequency in the control condition. In the cued condition the model responded correctly every time, taking advantage of ACT-R's ability to respond to a prespecified color (red). Once visual attention was "captured" by the novel appearance of the cue at the step,

retrieval of the correct knowledge chunk for the postcompletion step was guaranteed. This behavior was generated by an IF-THEN type rule in the model, dictating retrieval of the postcompletion goal in response to the red cue. This followed instructions in the training manuals, which asked subjects to complete the postcompletion step when the "red indicator" lit up. Hence, the model demonstrated the correct application of this rule with training, similar to performance by participants.

### 5.3. Further Work

The current model is a work in progress. For example, this model fails to recognize failed firing attempts ('target not destroyed'). In further iterations of the model, step times may be fit to data collected from Experiment 1 and 2. This will require that some current abstractions of the model be removed, particularly its retrieval of set coordinates for the location of a visual object in lieu of a visual search. This may also open the possibility of modeling non-postcompletion errors at each individual step as well as generating a plausible account for human reaction to the cue, unaccounted for by the current model. The ultimate goal of this work would undoubtedly be a model that is able to predict errors and the effects of various interventions and redesigns given an interface and task.

## 6. CONCLUSION

Several suggestions for the design of interfaces used in routine procedural tasks can be gleaned from this work. First, this work has demonstrated that timing is key to the success of a reminder. Both the mode indicator (which appears before the postcompletion step) and the downstream error cost had no reliable effect, in contrast to the just-in-time cue introduced in Experiment 2. Furthermore, when implementing visual cues as reminders for users, it seems that movement and/or shape are strong determinates of their effectiveness. Our cue in Experiment 1 appeared at the same exact location as the cue

used in the second experiment, yet generated no reliable differences from the no cue control condition. The mode indicator, which relied on static contextual cues on the interface, also did not reduce the number of errors at the postcompletion step. Even negative feedback from the system (Experiment 1) in the form of a downstream error cost and as reprimands from an "overseer" (Serig, 2001) fail to generate any significant reduction in the frequency of errors. Postcompletion errors cannot simply be willed away.

However, combatting potential errors may not be so simple as merely adding attention-grabbing cues to the interface as reminders. As shown by the Medical task, differences in task (e.g., length) and interface (e.g., background color) characteristics may also attenuate the effectiveness of these cues. For example, the color of a cue may be affected by its relative contrast to a background color. As Sutcliffe (1995) notes, the presence of more than one stimulus in conflict with the others can reduce individual effectiveness. Additionally, the fact that our participants had explicit training on the meaning of the cue should be considered. Simply placing blinking arrows or other novel cues on the interface would affect new users differently from those who had been trained.

One of the problems with the cue used in Experiment 1 may be the speed at which our attention shifts in routine procedural tasks with medium to high level of skill and external pressures. Hence, while the cue used in the first experiment appeared in temporal conjunction with the completion of the previous step and in spacial proximity to the targeting window, it was still overlooked. In contrast, the successful blinking cue continued to generate attention-capturing movement until the postcompletion step was completed. Moreover, it offered immediate information (arrows pointing to the correct button) about its meaning. Consider automobiles that use auditory alarms (a high-pitched beep) to remind drivers that they left the headlights on as they get out. Often, even this noise is insufficient (particularly without training or previous experience) to remind the driver, particularly if he/she is in a rush to exit. Although not evaluated here, placement of the reminder in an advance position along the sequence of task steps is known to help,

as is the case in common daily tasks. Reason (2003) gives a simple example of leaving something in the doorway, as a reminder to take it when leaving.

Finally, since the graphical change (and intention formation by the participant) occurs prior to when the postcompletion step should take place, the mode indicator condition in Experiment 1 may have failed due to its demands on prospective memory. This is the remembering and execution of delayed plans with no additional prompts at the time of intended retrieval (Guynn, McDaniel, & Einstein, 1998). Similar to the popular Einstein-McDaniel paradigm, where a participant must press a key when they encounter a particular word during a separate task of rating and memorizing word lists, the participant in the mode condition was also required to act on the mode indicator after completing the main goal of the task (delay). According to Marsh and Hicks (1998), prospective memory performance decreases with increasing load on the executive resources, such as working memory. Hence, mode indicators, which are used as aids and reminders, are in fact susceptible to the same stressors they are meant to alleviate.

In conclusion, this work is a first step towards extending our knowledge of successful visual cue-based reminders in computer-based tasks. By examining how people make use of visual cues in computer-based procedural tasks to produce correct behavior, understanding may also be gained as to what factors, alternatively, lead to error. While general human performance data, physiological knowledge, and guidelines exist to suggest (e.g., Sutcliffe, 1995) the visual properties of effective cues and reminders, they are vague and have not been elaborately explained at the cognitive level in relation to computer-based tasks. With the tendency to err being innate in humans, computer interfaces must be designed to both reduce the frequency of human error and remediate their effects.

Additionally, these studies have shown that the traditional task analysis can be insufficient to predict errors, particularly those low in frequency, in computer-based tasks. A more extensive method that can account for the reported observations is

necessary for the reliable prediction of errors and thorough interface evaluation. Such an approach would undoubtedly be unwieldy to implement manually and, hence, modeling is suggested as the eventual solution. Those studying eye movements in reading (e.g., Rayner, 1998) have similarly turned to computational modeling as a means of turning their data into something that can be iteratively tested and validated. For a valid modeling approach, however, it will be necessary to determine how specific visual properties affect our visual attention, since our actions on an interface are largely guided by visual cues (e.g., Gray, 2000). Only then can an evaluation of human error be truly useful – for what good is pinpointing a potential user error if no good design solution can be suggested?

# 7. REFERENCES

Altmann, E. M. (2002). Functional decay of memory for tasks. *Psychological Research, 66,* 287-297.

Altmann, E. M. & Trafton, J. G. (1999). Memory for goals: An architectural perspective. *Proceedings of the twenty-first annual conference of the Cognitive Science Society* (pp. 19-24). Hillsdale, NJ: Erlbaum.

Anderson, J. R., & Lebiére, C. (1998). *The Atomic Components of Thought*. Mahway, NJ: Erlbaum.

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Quin, Y. (2004, in press). An integrated theory of the mind. To appear in *Psychological Review*.

Baars, B. J. (1992). The many uses of error: Twelve steps to a unified framework. In B. J. Baars (Ed.), *Experimental Slips and Human Error: Exploring the Architecture of Volition*. New York: Plenum.

Bogner, M. S. (1994). *Human Error in Medicine*. Hillsdale, NJ: Lawrence Erlbaum.

Brennan, T. A., Leape, L. L., Laird, N. M., Hebert, L., Localio, A. R., et al. (1991). Incidence of adverse events and negligence in hospitalized patients: Results from the Harvard Medical Practice Study I. *New England Journal of Medicine, 324*, 370-376.

Bruner, J. S., Goodnow, J. J., & Austin, G. A. (1956). *A Study of Thinking*. New York: Science Editions.

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

Detterman, D. K. (1993). The case for the prosecution: Transfer as epiphenomenon. In Detterman & Sternberg (Eds.), *Transfer on Trial: Intelligence, Cognition, and Instruction* (pp. 1-38). Ablex, Norwood, NJ.

Ericsson K. A., & Kintsch W. (1995). Long-term working memory. *Psychological Review, 102*, 211-45.

Fiske, S. T., & Taylor, S. E. (1994). *Social Cognition* (2nd ed.). New York: McGraw-Hill.

Fitts, P. M., & Posner, M. I. (1967). *Learning and skilled performance in human performance*. Belmont, CA: Brooks/Cole Publishing Company.

Gick, M. L., & Holyoak, K. J. (1980). Analogical problem solving. *Cognitive Psychology, 12*, 306-355.

Gray, W. D. (2000). The nature and processing of errors in interactive behavior. *Cognitive Science. 24(2)*, 205-248.

Guynn, M. J., McDaniel, M. A., & Einstein, G. O. (1998). Prospective memory: When reminders fail. *Memory & Cognition*, 26(2), 287-298.

Hitch, G. J. (1980). Developing the concept of working memory. In G. Claxton (Ed.), *Cognitive Psychology: New Directions*. London: Routledge & Kegan Paul.

Hollnagel, E. (1993). *Human Reliability Analysis, Context and Control*. Academic Press, London.

Hollnagel, E. (1998). *Cognitive reliability and error analysis method (CREAM)*. Oxford: Elsevier Science Ltd.

Johnson-Laird, P. N. (1983). Thinking as a skill. In J. Evans (Ed.), *Thinking and Reasoning: Psychological Approaches*. London: Routledge & Kegan Paul.

Kieras, D. E., Wood, S. D., & Meyer, D. E. (1997). Predictive engineering models based on the EPIC architecture for multimodal high-performance human-computer interaction task. *ACM Transactions on Computer-Human Interaction, 4(3)*, 230-275.

Laughery, K.R., & Wogalter, M..S. (1997). Warnings and Risk Perception. In G. Salvendy (Ed.), *Handbook of Human Factors and Ergonomics* (2nd ed.). New York: John Wiley.

Lebiére, C., Anderson, J. R., & Reder, L. M. (1994). Error modeling in the ACT-R production system. *Proceedings of the Sixteenth Annual Meeting of the Cognitive Science Society* (pp. 555-559). Hillsdale, NJ: Erlbaum.

Marsh, R. L., & Hicks, J. L. (1998). Event-based prospective memory and executive control of working memory. Journal of Experimental Psychology: Learning, Memory, and Cognition, 24(2), 336-349.

Miller, D., & Swain, A. (1986). Human error and human reliability. In G. Salvendy (Ed.), *Handbook of Human Factors/Ergonomics*. New York: John Wiley.

Monk, A. (1986). Mode errors: a user-centered analysis and some preventative measures using keying-contingent sound. *International Journal of Man-Machine Studies, 24*, 313-327.

Mosier, K. L., & Skitka, L. J. (1996). Human decision-makers and automated decision aids: Made for each other? In R. Parasuraman & M. Mouloua (Eds.), *Automation and Human Performance: Theory and Applications* (pp. 201-220). Mahway, NJ: Lawrence Erlbaum Associates.

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

Norman, D. A. (1988). *The Psychology of Everyday Things*. New York: Basic Books.

Park. K. S. (1997). Human Error. In G. Salvendy (Ed.), *Handbook of Human Factors and Ergonomics* (2nd ed.). New York: John Wiley.

Polson, P. G., & Lewis, C. H. (1990). Theory-based design for easily learned interfaces. *Human-Computer Interaction, 5*, 191-220.

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

Rasmussen, J. (1987). The definition of human error and a taxonomy for technical system design. In J. Rasmussen, K. Duncan, & J. Leplat (Eds.), *New Technology and Human Error* (pp. 53–62). Chichester: John Wiley.

Reason, J. (1984). Lapses of attention in everyday life. In R. Parasuraman & D. R. Davies (Eds.), *Varieties of Attention* (pp. 515-549). New York: Academic Press.

Reason, J. (1990). *Human Error*. Cambridge, UK: Cambridge University Press.

Reason, J. (1997). *Managing the Risks of Organizational Accidents*. Aldershot: Ashgate Publishing Company.

Reason, J. (2002). Combating omission errors through task analysis and good reminders. *Quality and Safety in Healthcare, 11*, 40-44.

Remington, R.W., Johnston, J.C., & Yantis, S. (1992). Involuntary attentional capture by abrupt onsets. *Perception & Psychophysics, 51,* 279-290.

Rieman, J., Young, R. M., & Howes, A. (1996). A dual-space model of iteratively deepening exploratory learning. *International Journal of Human-Computer Studies, 44,* 743-775.

Senders, J., & Moray, N. (1991). *Human Error: Cause, Prediction, and Reduction*. Mahwah, NJ: Lawrence Erlbaum.

Serig, E. M. (2001). *Evaluating Organizational Response to a Cognitive Problem: A Human Factors Approach.* Doctoral dissertation, Rice University, Houston, TX.

Sheridan, T. B. (1997). Supervisory Control. In G. Salvendy (Ed.), *Handbook of Human Factors and Ergonomics* (2nd ed.). New York: John Wiley.

Skitka, L. J., Mosier, K. L. & Burdick, M. (1999). Does automation bias decision-making? *International Journal of Human-Computer Studies, 51*, 991-1006.

Stanton, N. A., & Baber, C. (1996). A systems approach to human error identification. *Safety Science, 22(1-3),* 215-228.

Sutcliffe, A.G. (1995). *Human-Computer Interface Design*. London: Macmillan Press Ltd.

Testimony of the Three Mile Island Operators. (1979). *United States President's Commission on the Accident at Three Mile Island, Vol 1*. Washington, DC: U.S. Government Printing Office.

Thimbleby, H. (1990). *User Interface Design*, ACM Press Frontier Series, Addison-Wesley.

Treisman, A. (1986) Features and objects in visual processing. *Scientific American, 254* . 114-124

Van Cott, H. (1994). Human errors: Their causes and reduction. In M. S. Bogner (Ed.), *Human Error in Medicine* (pp. 53-65). Hillsdale, NJ: Lawrence Erlbaum Associates, Publishers.

Wallace, D. F., & Huffman, D. (1990). Use of a visual cue to reduce errors in exiting a crash-bar type door. *Proceedings of the Human Factors Society 34th Annual Meeting* (pp. 567-569).

Woods, D. D., & Cook, R. I. (1999). Perspectives on Human Error: Hindsight Bias and Local Rationality. In F. Durso (Ed.), *Handbook of Applied Cognitive Psychology* (pp. 141-171). New York: John Wiley.

Yantis, S., & Jonides, J. (1984). Abrupt visual onsets and selective attention: Evidence from visual search. *Journal of Experimental Psychology: Human Perception & Performance, 10*, 601-621.

Yantis, S., & Jonides, J. (1988). Uniqueness of abrupt visual onset in capturing attention. *Perception & Psychophysics, 43(4)*, 346-354.

Young, R. M. (1994). The unselected window scenario: analysis based on the SOAR cognitive architecture. *Proceedings of the Computer Human Interaction (CHI) Conference, 1994*.

Zhang, J., & Norman, D. A. (1994). Representations in distributed cognitive tasks. *Cognitive Science, 18*, 87-122.

# 8. APPENDIX A

*Tactical Task Manuals*

# 9. APPENDIX B

*Experiment 1 Point Table*

Prizes:

First Place - $25 Amazon.com gift certificate
Second Place - $15 Amazon.com gift certificate
Third Place - $10 Amazon.com gift certificate

Time Bonus:

Conn
< 10 sec = +100 points
10-20 sec = +50 points

Transporter
< 10 sec = +100 points
10-20 sec = +50 points

Tactical
< 15 sec = +100 points
15-25 sec = +50 points

Task Bonus:

Correct step = +25 points
Incorrect step = -50 points

Letter Task Penalty:

Incorrect = -200 points

# 10. APPENDIX C

*Post-training quiz questions (Tactical, all conditions)*

The phaser should be fired when the target:

a. is directly aligned with the crosshairs.
b. is within the small, donut-shaped area directly around the crosshairs.
c. disappears.
d. appears.

The last step on the phaser console, before returning to main control, is to click the:

a. ‘Power Connected’ box.
b. ‘Charge’ button.
c. ‘Tracking’ button.
d. ‘Firing’ button.

The transporter beam is engaged by moving the cursor over the target and pressing the:

a. spacebar.
b. mouse button.
c. shift key.
d. enter.

When at the navigation console, the ‘Course Correction’ equals:

a. the current – programmed heading.
b. the current + programmed heading.
c. The programmed – current heading.
d. The programmed + current heading.

*The appearance of the blinking arrows in the tracking window next to the ‘Tracking’ button means you must:

a. return to ‘Main Control’.
b. click ‘Tracking’ to turn off the Tracking system.
c. press fire (spacebar) again.
d. Reconnect power and begin the task again.

*If you destroyed the target and the ‘Tracking’ button is lit green with ‘Tracking mode enabled’ indicator next to it, you must:

a. return to ‘Main Control’.
b. click ‘Tracking’ to exit Tracking mode.
c. press fire (spacebar) again.
d. reconnect power and begin the task again.

The letter recall task demands that, when you hear the tone, you type in the last:

a. letter you heard.
b. four letters you heard.
c. five letters you heard, in reverse order.
d. three letters you heard.

*Post-training quiz questions (Medical only, all conditions)*

The correct rate to program for the pump exercise is:

a. “150”.
b. “200”
c. “1000”
d. “125”

Before moving to Main Control and the next task, you must exit the (Infusion console) Programming mode, as indicated by the ‘Programming Mode Indicator’, by pressing the:

a. ‘Main Display’ button.
b. ‘Rate’ button.
c. ‘Vol’ button.
d. ‘Insert Cassette’ button.

Pressing the ‘Rate’ button puts the pump in:

a. Main Display.
b. Programming mode.
c. Infusing mode.
d. none of the above.

The last step on the infusion console, before returning to Main Control, is to click the:

a. ‘Main Display’ button.
b. ‘Rate’ button.
c. ‘Vol’ button.
d. ‘Insert Cassette’ button.

With the last step on the infusion console, before returning to Main Control, the ‘Infusion Flowing’ message and blinking arrows signify that you must click the:

a. ‘Main Display’ button.
b. ‘Rate’ button.
c. ‘Vol’ button.
d. ‘Insert Cassette’ button.

# 11. APPENDIX D

*Sample Tactical Manual (Mode)*

## Starfleet Operations Manual

Model MB-X30.1(2) Phaser Control Bank

This manual describes how to operate the MB-x30.1 Starfleet standard phaser control bank, the primary weapon on current Starfleet vessels. Understanding how to operate this system is critical for any Starfleet officer.

Figure 1 below shows the phaser control bank interface.

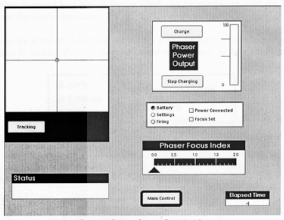



Figure 1. Phaser Control Bank interface

There are four essential steps involved in operating the MTS:

1. Charging the phaser.
2. Setting the focus of the phaser beam.
3. Tracking the target.
4. Firing the phaser.

Each step will be further described in the following pages of the manual.

## Step 1. Charging the Phaser

Overview of steps:

1. Click 'Power Connected'
2. Click 'Charge'
3. Wait until phaser charges above line
4. Click 'Stop charging'
5. Click 'Power Connected'

The x30 class phaser requires more energy than can be generated by a standard power plant. This problem is solved by a virtual battery which can be charged to yield the high output required (see diagram in Appendix).

Several steps are involved in charging the battery. First, the battery must be connected to the power source by clicking 'Power Connected' on the control panel, as shown in Figure 2.

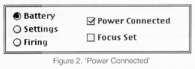



Figure 2. 'Power Connected'

Once the battery is connected, the phaser may be charged by clicking the 'Charge' button (Figure 3) and waiting for the meter to fall within the safe range marked by the horizontal lines. At this point, 'Stop Charging' should be clicked.

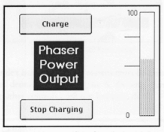



Figure 3. 'Charge', 'Stop Charging', and meter

*Warning: It is crucial that the battery charge falls within the allowable range, as overcharging can damage the unit and undercharging will make firing fail altogether.*

Once the phaser has charged, it is necessary to disconnect the battery from the power source by unchecking the 'Power Connected' box (Figure 2). Unless the power has been disconnected, it will not be possible to operate other phaser controls.

## Step 2. Setting Phaser Beam Focus

Overview of steps:

1. Click 'Settings'
2. Adjust location of slider to desired focus
3. Click 'Focus Set'

The x30 class phaser beam must be focused in order to be effective against a target. Higher dispersion translates to a larger perpendicular cross-section of the beam, making it easier to hit a target. However, the beam also becomes less damaging at higher settings, so proper adjustment must be learned. The Phaser Focus Index is found on the Phaser Control Bank.

The first step in setting the Focus Index is to enable the alteration of current settings. This is done by clicking 'Settings', as shown in Figure 4.

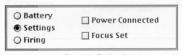



Figure 4. 'Settings'

The focus index must be set using the Phaser Focus Index slider. Click on the triangular indicator and drag it to the desired setting (Figure 5). Setting the focus to approximately 2/3 level is acceptable for most targets. As you progress through training you will get a better feel.

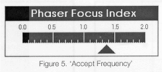



Figure 5. 'Accept Frequency'

Once the Phaser Focus Index has been set, the system must be locked. This is done by clicking the 'Focus Set' button, as demonstrated in Figure 6. As with charging the battery, it will be impossible to operate the other controls until this has been done.

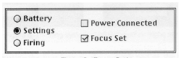



Figure 6. 'Focus Set'

Step 2. Setting Phaser Beam Focus

Overview of steps:

1. Click 'Settings'
2. Adjust location of slider to desired focus
3. Click 'Focus Set'

The x30 class phaser beam must be focused in order to be effective against a target. Higher dispersion translates to a larger perpendicular cross-section of the beam, making it easier to hit a target. However, the beam also becomes less damaging at higher settings, so proper adjustment must be learned. The Phaser Focus Index is found on the Phaser Control Bank.

The first step in setting the Focus Index is to enable the alteration of current settings. This is done by clicking 'Settings', as shown in Figure 4.

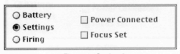



Figure 4. 'Settings'

The focus index must be set using the Phaser Focus Index slider. Click on the triangular indicator and drag it to the desired setting (Figure 5). Setting the focus to approximately 2/3 level is acceptable for most targets. As you progress through training you will get a better feel.

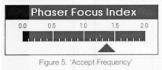



Figure 5. 'Accept Frequency'

Once the Phaser Focus Index has been set, the system must be locked. This is done by clicking the 'Focus Set' button, as demonstrated in Figure 6. As with charging the battery, it will be impossible to operate the other controls until this has been done.

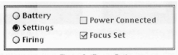



Figure 6. 'Focus Set'

## Step 3. Tracking the Target

Overview of steps:

1. Click 'Firing'
2. Click 'Tracking'
3. Use the number keys to adjust the location of the target indicator

The x30 class phaser contains a sophistical tracking system that is capable of bringing a target into firing range. Once the system has done its job, however, it is up to the operator to manually operate the phaser and fire.

The first step involved in the manual portion of tracking the target is to enable phaser firing. This is done by clicking the 'Firing' button, as shown in Figure 7.

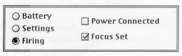



Figure 7. 'Firing' button engaged

The tracking system is next turned on by clicking on the 'Tracking' button, as shown in Figure 8. It is possible to tell that the tracking system is active by noting the presence of the target indicator (blip), 'Tracking Mode' indicator (next to the'Tracking button; Figure 8, right), and the crosshairs (Figure 8, right).


Figure 8. Tracking system in off (left) and on (right) states

Adjustments are made using the four keys on the numeric pad of the keyboard. These keys will bring the crosshairs closer to the target in the direction you press. Hence, if you press up, the target will move downward towards the crosshairs. For the target in Figure 8 (right) you would press the left direction key to move the center crosshairs toward the black dot. Use moderation in the rate of keypresses. Since the system works in bursts, holding a key down will not work.

Due to the difficulty of tracking high-speed objects such as incoming enemy vessels, it is not guaranteed that the target will be hit, no matter how well the tracking is adjusted. Aiming straight on is likely to miss, in fact, due ot the donut shape of the phaser beam. *It is thus optimal to aim quickly and fire as soon as you are within range rather than waiting until you are dead on.*

## Step 4. Fire the Phaser

Overview of steps:

1. Press the space bar
2. Determine if the target has been destroyed
3. If so, click 'Tracking'
4. If not, return to Step 1

Once the tracking has been adjusted and the crosshairs are within range of the target blip, the phaser should be fired immediately by pressing the space bar.

This will have several effects. First, you will hear the sound of the power discharge. You will also notice that several things on the control panel will have returned to their "rest" state. The status of the panel will indicate the results of the firing. There are three possibilities:

1. The phaser will miss the target
2. The phaser will hit the target, but will not destroy it
3. The phaser will destroy the target.

In either of the first two cases, it will be necessary to return to Step 1 (Charging the Phaser) in order to fire the phaser again.

In the third case, the task is complete and you must turn off the tracking system, which will still be in the 'on' state. This is imperative, as the tracking system will automatically lock on to targets by itself if left on. Several accidents in the recent past with ships firing on friendly ships have been attributed to a failure to shut off the Tracking system. Clicking the 'Tracking' button will turn off the tracking system causing the crosshairs and 'Tracking Mode' indicator to disappear. Once you have done this, click 'Main Control', at the bottom, to move on to the next task.

## Review. Summary of Steps

Overview of steps:

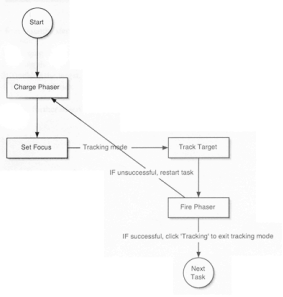
Step 1. Charge the Phaser
1. Click 'Power Connected'
2. Click 'Charge'
3. Wait until phaser charges the appropriate amount
4. Click 'Stop Charging'
5. Click 'Power Connected'

Step 2. Set Phaser Beam Focus
1. Click 'Settings'
2. Adjust location of slider to desired focus
3. Click 'Focus Set'

Step 3. Track the Target
1. Click 'Firing'
2. Click 'Tracking'
3. Use arrow keys to adjust location of the target indicator

Step 4. Fire the Phaser
1. Press the space bar
2. Determine if the target has been destroyed
3. If so, click 'Tracking'
4. If not, return to Step 1

Then click 'Main Control' to move on to the next task



*Postcompletion Step Instructions*

Tactical Manual (Control):

Tactical Manual (Cue):
“The tracking system is next turned on by clicking on the ‘Tracking’ button, as shown in Figure 8. It is possible to tell

Tactical Manual (Mode):

*Main Control Manual*

# Starfleet Operations Manual

Main Control Console

This manual describes how to use the Main Control console during your Operations Officer Qualifying exams.

The Main Control console is your "home station" during this exam. From here you can access the other bridge stations: Pump, Conn, and Transporter. The Main Control will provide you with two important pieces of information: 1. feedback about your performance on the last task and 2. the next task that you must complete.

Figure 1 shows the main control console.

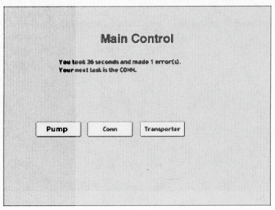



Figure 1. Main Control console

The bridge stations are identified as follows in the Main Control console:

1. Pump
2. Conn
3. Transporter

There is one basic step to operate the Main Control console:

1. Click on the button corresponding to the task identified in the feedback message, as shown in Figure 2 ("Your next task is the Conn").

Qualifying will take part in two phases spaced two days apart. Phase I will consist of training and Phase II will be testing.

## Phase I. Training

Training will commence on your first day of qualification. A manual will be provided for each of the bridge tasks (Pump, Conn, and Transporter). You will have four trials on which to train at each station. The manual may be used on the first trial only.

Instructions:

1. Read the entire manual for the indicated bridge station
2. Click on the button for that station in Main Control to begin the first training trial. YOU MUST READ THE ENTIRE MANUAL BEFORE LEAVING MAIN CONTROL. You may use the manual as an aide to complete this trial.
3. If you successfully complete the trial, you will be prompted to return your manual to the proctor. Please raise your hand to signal him/her.
4. After returning your manual to the proctor, complete the remaining three practice trials for that bridge station.
5. According to the prompt at Main Control, begin training on the next bridge station.

## Phase 2. Testing

You will undergo testing on day 2 of your qualification. Scores will be based your accuracy and time on each task. As your performance on these tasks will influence your standing for Operations Officer School candidacy, please note your performance on each task, as reported in the Main Control console, and seek to improve it on each subsequent trial.

Scores from Phase 2 will be used in contention for prizes. The top 5 scorers on the candidacy exams, judged by the number of points scored, will receive cash bonuses (refer to Prize and Point scale).

You will also be tested on your ability to respond to demands in the environment not directly related to your performance at each station. During the testing phase of the qualifying exam, the hectic environment of the bridge will be simulated. It is crucial for any officer to be able to attend to the task at hand while simultaneously keeping an eye and ear out for critical information "in the background."

Hence, you will hear individual letters broadcast through your headphones. At random intervals, you will be prompted to input the last three letters that were broadcast. Your accuracy in recalling this information will be tracked and may influence your standing for Operations Officer School candidacy. It is necessary to respond to these prompts to continue the task at hand. They are heavily weighted in terms of point value.

Please commence with your qualification exam when you have finished reading this manual.

*Good luck, cadet!*

*Navigation Manual*

# Starfleet Operations Manual

Model RD-x30.2 Navigation System

This manual describes how to operate the RD-x30.2 Starfleet standard navigation system. Understanding how to operate this system is crucial for any Starfleet officer.

Figure 1 below shows the navigation console.

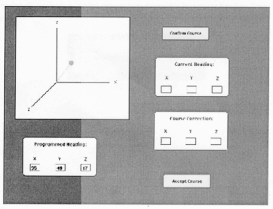



Figure 1. Navigation console

There are three essential steps involved in operating the navigation system:

1. Determine the ship's course heading relative to the programmed heading.
2. Compute course difference if current heading deviates from programmed heading.
3. Enter course correction into the navigation system.

Each step will be further described in the following pages of the manual.

## Step 1. Determine Ship's Course Heading

Overview of steps:

1. Click 'Confirm Course'

During space flight, variations in the atmosphere (i.e., solar winds, stellar dust fields, etc.) can influence a starship's projected course. Thus, it may be necessary to engage in course correction.

The first step of correcting a ship's course is to determine its current heading. This is done by clicking on the 'Confirm Course' button on the control panel, as shown in Figure 2.


Figure 2. 'Confirm Course'.

## Step 2. Compute Difference

Overview of steps:

1. Compare course heading in 'Programmed Heading' with course heading in 'Current Heading'
2. Enter course correction in the 'Course Correction' text boxes

If the course identified in the 'Course Heading' matches the 'Programmed Heading', enter '0' on the X, Y, and Z fields by using the numeric keypad.

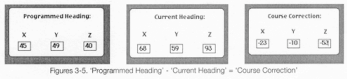
If the course identified in 'Current Heading' does NOT match the 'Programmed Heading', you must compute the difference between the intended (programmed) heading course and the actual (current) course. To compute the difference, subtract the 'Current Heading' values from the 'Programmed Heading' values. For example, if the 'Programmed Heading' is [45, 49, 40], and the 'Current Heading' is [68, 59, 93], you would calculate the course correction as follows:

Axis: Programmed - Current
X: 45 - 68 = -23
Y: 49 - 59 = -10
Z: 40 - 93 = -53

You would thus enter [-23, -20, -53] in the 'Course Correction' field. Note that if 'Programmed Heading' for a value is less than the 'Current Heading', the 'Course Correction' value will be negative (you need to move "down" in space to reach the correct heading).

Figures 3-5 illustrate this process.

*Note: The 'Current Heading' + the 'Course Correction' should equal the 'Programmed Heading'*




Figures 3-5. 'Programmed Heading' - 'Current Heading' = 'Course Correction'

## Step 3. Enter Course Correction

Overview of steps:

1. Click 'Accept Course'

In order to accept the course correction into the navigation computer, click the 'Accept Course' button, as shown in Figure 6.


Figure 6. 'Accept Course' button

Once this is done, the entered correction is processed by the navigation computer, making the correcting the ship's course.

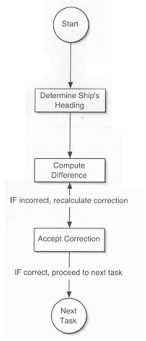
**Review. Summary of Steps**

Overview of steps:

Step 1. Determine Ship's Course Heading
1. Click 'Confirm Course'

Step 2. Compute Difference from the Programmed Heading
1. Compare course heading in 'Programmed Heading' with course heading in 'Current Heading'
2. Calculate difference between 'Programmed Heading' and 'Current Heading' by subtracting 'Current Heading' from 'Programmed Heading' (Value = Programmed - Current)
3. Enter course correction in 'Course Correction' text boxes

Step 3. Enter course correction into the navigation system
1. Click 'Accept Course'



*Transporter Manual*

# Starfleet Operations Manual

Model MB-x30.3 Manual Transporter System (MTS)

This manual describes how to operate the MB-x30.3 Starfleet standard Manual Transporter System (MTS), the primary method of bringing aboard crewmembers in hostile circumstances when automatic transporters are being jammed.

Figure 1 below shows the MTS interface.

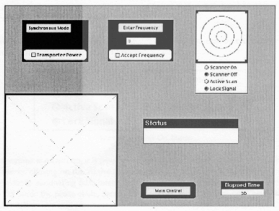



Figure 1. Manual Transporter System interface

There are four essential steps involved in operating the MTS:

1. Lock on to the homing signal.
2. Setting the jamming frequency.
3. Synchronizing the transporter and homing signal.
4. Energizing the transporter.

Each step will be further described in the following pages of the manual.

## Step 1. Lock on to Homing Signal

Overview of steps:

1. Click 'Scanner On'
2. Click 'Active Scan'
3. Wait until scanner homes in on a valid signal
4. Click 'Lock Signal'
5. Click 'Scanner Off'

When crewmembers need to be beamed aboard in hazardous situations, one or more of them will use their communicator to send out a signal broadcasting their location. It is necessary to scan for this signal and then lock the MTS onto it.

Several steps are involved in locking onto a signal. The first is to turn on the scanner by clicking the 'Scanner On' button, as depicted in Figure 2.

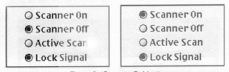



Figure 2. 'Scanner On' button

Once the scanner is turned on, it is possible to activate the scanning system and track the signal. This is done by clicking on the 'Active Scan' button. The scanner will then report a number of signals, gradually eliminating false ones until there is only one remaining. Wait for the unique signal to fall inside the phase circle, then press the 'Lock Signal' to lock on. This is illustrated in Figures 3 and 4.

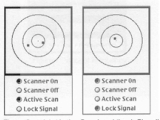



Figure 3 and 4. 'Active Scan' and 'Lock Signal'

It is critical that there be only one active signal in the scanner to prevent anything other than a crewmember from being beamed aboard. It is equally important that the last signal be in-phase, or within the smallest circle, to ensure a reasonable chance of success. Once the signal has been locked in, you must turn off the signal scanner by clicking on the 'Scanner Off' button.

## Step 2. Setting the Jamming Frequency

Overview of steps:

1. Click 'Enter Frequency'
2. Type in the desired scanner frequency
3. Click 'Accept Frequency'

You must next enter a jamming frequency into the MTS, ranging from 1-100. Higher frequencies increase the probability of successfully jamming the enemy's attempts at jamming your own signal. Nonetheless, a higher frequency also lowers the strength of the transporter beam, making it less likely to bring the crew aboard once the beam has been activated. It is your responsibility as the operator to decide on the best tradeoff.

The first step in setting the jamming frequency is enabling keyboard entry of a signal value by clicking the 'Enter Frequency' button. Clicking this button will cause a blinking cursor to appear on the frequency field, as shown in Figure 5.


Figure 5 (left) and 6 (right). 'Enter Frequency'

After text entry is enabled, the frequency value between 1 and 100 should be entered on the keyboard. For this exercise, start in the middle with a value around 50 and adjust as you see fit. Once entered (Figure 6), it is necessary to tell the system to commit to that frequency. You may do this by clicking on the 'Accept Frequency' button, as shown in Figure 7.

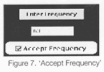



Figure 7. 'Accept Frequency'

## Step 3. Synchronizing Transporter and Signal

Overview of steps:

1. Click 'Transporter Power'
2. Click 'Synchronous Mode'
3. Use the mouse to track the homing signal

The x30 class transporter system must be manually synchronized with the homing signal. This requires that you first connect power to the transporter system, tell the system to allow synchronous tracking by the operator, and then manually track the signal. Turning on the main transporter power is done by clicking on the 'Transporter Power' button, as shown in Figure 8.

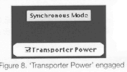



Figure 8. 'Transporter Power' engaged

Second, it is necessary to switch the system into synchronous tracking mode. This is done by clicking on the 'Synchronous Mode' button, which is also displayed in Figure 8. When this button is clicked, the cursor will turn into a targeting circle with crosshairs, and the square target signal will appear in the tracking area of the screen (Figure 9). The target will be in constant motion, reflecting the instability of the homing signal. Move the mouse as close to the target as possible before energizing the transporter.

## Step 4. Energizing the Transporter

Overview of steps:

1. Click the mouse button
2. Click 'Synchronous Mode'
3. Determine if the beam was successful

When the MTS is in synchronous mode, the transporter will energize when the mouse button is clicked. This should be done as soon as the targeting crosshairs are close to the target. Energizing the transporter will bring the MTS display to rest. Next, switch out of the synchronous mode by again clicking 'Synchronous Mode' to determine the outcome of the beam. The status box and audio feedback should indicate if the transportation was successful or not.

If it was unsuccessful, return to Step 1 (Lock on to the homing signal) and try again. If the beam is successful, you may return to the main control screen by clicking on the 'Main Control' button at the bottom of the screen.


Figure 10. 'Main Control'

## Review. Summary of Steps

Overview of steps:

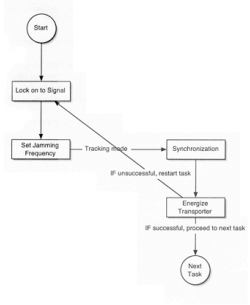
Step 1. Lock on to the Homing Signal
1. Click 'Scanner On'
2. Click 'Active Scan'
3. Wait until scanner homes in on a valid signal
4. Click 'Lock Signal'
5. Click 'Scanner Off'

Step 2. Setting the Jamming Frequency
1. Click 'Enter Frequency'
2. Type in the desired scanner frequency
3. Click 'Accept Frequency'

Step 3. Synchronize the Transporter and Homing Signal
1. Click 'Transporter Power'
2. Click 'Synchronous Mode'
3. Use the mouse to track the homing signal

Step 4. Energizing the Transporter
1. Click the mouse button
2. Click 'Synchronous Mode'
3. Determine if the beam has been successful
4. If not, return to Step 1

Return to Main Control



*Sample Medical Manual (Mode)*

# Starfleet Operations Manual

Model PC-x30.4 Hypospray Infusion Console

This manual describes how to operate the PC-x30.4 Hypospray Infusion console, standard on all starship Biobeds. This console is also on the portable Hypospray devices included in all Starfleet standard medikits, making it crucial for any officer to understand its basic operation.

Figure 1 below shows the Infusion Console interface.

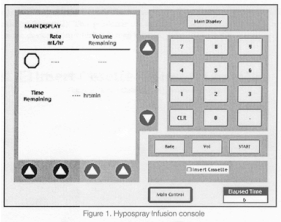



Figure 1. Hypospray Infusion console

There are four essential steps involved in operating the class x30 Infusion console:

1. Insert the cassette
2. Program the rate of infusion
3. Program the volume to be infused
4. Start the infusion flow

Each step will be further described in the following pages of this manual.

## Step 1. Insert the Cassette

Overview of steps:

1. Click 'Insert Cassette'

The x30 class Hypospray device is used for subcutaneous and intravenous administration of carefully-controlled dosages of medication on a subject. The Hypospray injects the subject by means of a pinpoint, high-pressure, microscopic stream. This allows medication to be administered painlessly through the skin or clothing without mechanical penetration.

Quick loading of any standard medication cassette (such as Delactovine) is done by pressing 'Insert Cassette' (Figure 2). This pre-loads the medication in the cassette into a high-pressure chamber in the device, from which it is then infused.


Figure 2. 'Insert Cassette' box before and after

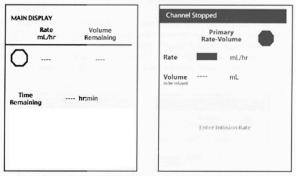
## Step 2. Program the Rate of Infusion

Overview of steps:

1. Click 'Rate'
2. Click numerical buttons to program value (1-2-5) for the rate of infusion

The Hypospray device may be used for controlled infusion of medication over time. This, of course, requires that you input a rate at which the medication should be delivered.

As seen in Figure 3 (left), the console is initially set in the 'MAIN DISPLAY' mode. This is where further functionality may be accessed for advanced presentation of infusion data. However, these options will not be covered in this basic manual.




Figures 3 and 4. Main Display (left) and Programming screen (right)

To begin programming an infusion, click 'Rate' (Figure 5). This will bring up the programming screen (Figure 4, right). The system will now be in programming mode.


Figure 5. 'Rate' button

The black box indicates the current field into which values may be entered. Click the number buttons on the right hand panel to enter the value: "125". This is a common infusion rate used for Delactovine, a frequently administered stimulant found in most Starfleet issue medikits with the Hypospray device. We will maintain this infusion order for all trials of this exercise.

## Step 3. Program the Volume to be Infused

Overview of steps:

1. Click 'Vol'
2. Click number buttons to program the value (1-0-0-0) for the volume to be infused.

It is possible to deliver infusions longer than the typical small volume quick infusions done in the field. For this exercise you will program such an infusion.


Figure 6. 'Vol' button

First, click 'Vol' (Figure 6) to bring the black text box down to the 'Volume to be infused' field (Figure 7, left). As with the rate, the number buttons on the right hand of the interface can now be used to enter values. Enter the value: "1000" (Figure 7, right). Again, this is a commonly administered order for Delactovine and will be used throughout training to preserve the simplicity of the exercise and ensure that you learn the most basic functions of the Hypospray device.

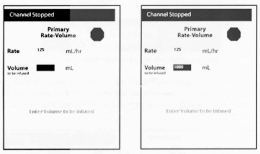



Figure 7. 'Volume to be infused' field selected (left) and entered (right)

## Step 4. Start the Infusion Flow

Overview of steps:

1. Click 'START'
2. Click 'Main Display'

Once the rate and volume have been programmed, the infusion may be initiated by pressing the 'START' button (Figure 8). This should bring up the message 'Infusion Flowing', as seen in Figure 10 (bottom left), and an auditory chime, to signal the beginning of the infusion.


Figure 8. 'START' button

If the infusion is flowing, click 'Main Display' (Figure 9), which will have blinking arrows around it to prompt you. This will cause the device to exit the Programming mode and return to the Main Display, which should now appear as in Figure 11 (right). This last step is *crucial*, as leaving the device in the Programming mode has led to several deaths aboard Starships recently. The device can be accidentally reprogrammed if left in this mode, leading to dangerous over- or underinfusions.

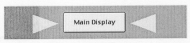



Figure 9. 'Main Display' button with flashing arrows

Furthermore, only the Main Display shows 'Time Remaining' for the infusion, allowing you or anyone on your crew to better assess the status of the infusion in a quick glance. Thus it is essential that you leave the device in this state before clicking 'Main Control' to leave the console.

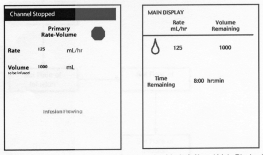



Figure 10 and 11. 'Infusion Flowing' in Programming Mode (left) and Main Display (right)

## Review. Summary of Steps

Overview of steps:

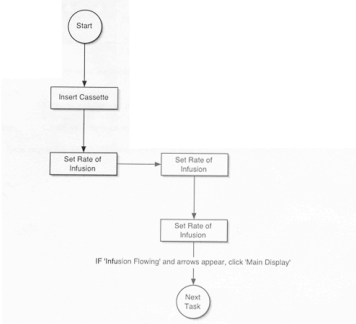
Step 1. Insert the Cassette
1. Click 'Insert Cassette'

Step 2. Program the rate of infusion
1. Click 'Rate' (Programming Mode engaged)
2. Click number buttons to program the value (1-2-5) as the rate of infusion

Step 3. Program the volume to be infused
1. Click 'Vol'
2. Click number buttons to program the value (1-0-0-0) for the volume to be infused

Step 4. Start Infusion Flow
1. Click 'START'
2. If infusion is flowing, click 'Main Display', indicated by flashing arrows, to exit Programming Mode

Return to Main Control



# 12. APPENDIX E

*Experiment 2 Point Table*

Prizes:

First place: $30
Second place: $20
Third place: $15
Fourth place: $10
Fifth place: $10

Time Bonus:

Conn
< 10 sec = +100 points
10 - 20 sec = +50 points

Transporter
< 10 sec = +100 points
10 – 20 sec = +50 points

Tactical
< 15 sec = +100 points
15 – 25 sec = +50 points

Task Bonus:

Correct step = +25 points
Incorrect step = -50 points

Letter Task Penalty:

Incorrect = -200 points

# 13. APPENDIX F

*Model Parameters*

(sgp :v nil :pm t :esc nil :ans 0.2 :rt -0.6)
(pm-set-params :real-time t :visual-movement-tolerance 1 :show-focus nil :randomize-time t)
(setsimilarities (s11 s12 -1.0))