RICE UNIVERSITY

Changing the Interface with Minimal Disruption: The Roles of Layout and Labels

by

Phillip H. Chung

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

Doctor of Philosophy

APPROVED, THESIS COMMITTEE:

____

____

Michael D. Byrne,
Assistant Professor, Psychology

Christopher M. Kelty,
Assistant Professor, Anthropology

____

David M. Lane,
Associate Professor, Psychology,
Statistics, and Management

____

Philip T. Kortum,
Professor-in-the-Practice, Psychology

HOUSTON, TEXAS

MAY 2006

ABSTRACT

Changing the Interface with Minimal Disruption: The Roles of Layout and Labels

by

Phillip H. Chung

This dissertation reports findings from two laboratory experiments and a field study demonstrating significant reliance by users on interface layout information in interactive tasks. In Experiment 1, a paradigm was introduced where either the layout of the interface was changed or labels were removed, after participants completed a minimum of eleven trials of a routine computer-based task. Since layout change had a more detrimental effect on performance, in Experiment 2, two methods expected to mediate that effect were explored: the addition of color and a layout based on a simple preexisting rule of top-to-bottom control order. Only the latter was effective, showing that introducing an interface layout that leverages preexisting knowledge can actually improve task performance. In the field study, a methodology was developed to put these findings to the test at a local family medicine clinic using an electronic medical records system. By studying nurses' use of an existing data entry form, a new form was designed to more closely follow their workflow. Similar to the top-to-bottom control order manipulation in Experiment 2, the new form layout seemed to produce better performance and was liked better by the nurses. Thus, in contrast to the vast body of literature in the field that has emphasized the importance of label information (e.g., Polson & Lewis, 1990) and goal structure (e.g., John & Kieras, 1996) in computer-based tasks, these findings reveal that users quickly learn to rely on layout information.

ACKNOWLEDGMENTS

First, I must thank my immediate family, without whose support I would obviously not have arrived at this point in life. Had I gone my own way, I would today be a designer of not-so-user-friendly things. Next, I am forever indebted to my advisor, Mike Byrne, who was always wise and generous in his guidance of my efforts at Rice. Other Rice professors critical to the realization of this work include my dissertation committee members, David Lane, Phil Kortum, Chris Kelty, and Ken Laughery, who aided me with my master's thesis. Thank you also Robert Henning at UConn for your cultivation of my interest in this field and strong recommendation to pursue graduate study.

My internship experiences were invaluable during my time in graduate school, not only helping me in my career after but also deepening my interest in the field. For these opportunities I must thank Jiajie Zhang and Todd Johnson at UTH and Jurine Adolf, Mihriban Whitmore, Kritina Holden, Vicky Byrne, Cindy Hudy, Lena DeSantis, and the rest of my coworkers at JSC. Also, thank you Lisa Spiro for allowing me to work on LESTER and Sarah Edmonson for your help at Baylor.

Finally, I could not have made it through the five years without my fellow students and friends: Mike Fleetwood, Chris Fick, Frank Tamborello, Kristen Greene, Sarah Everett, Jennifer Tsai, Michael O'Connor, Nohsook Park, Minmin Yang, Aniko Sandor, Danielle Paige, Camille Peres, Heejoon Han, Hyunwoo Yang, Daniel Kim, Kagome, Oaxaca HC, Rice KGSA, and anyone else I wanted to include but ran out of space for. Best wishes to all of you in your own endeavors, wherever the winds of life take your ship.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. Introduction

A recent article in *USA Today* (Baig, 2005) chronicled Microsoft's intention to again redesign the user interface for the next version of their Office software, which will display only the tools most frequently used. Such chronic and rapid change is perhaps the hallmark characteristic of human-computer interaction, and by nature we humans are both quick in defining the way technology evolves as well as adapting to its eventual use. Nevertheless, human adaptation is not always flawless, and careful thought by designers and engineers is essential to facilitate transition through each generation of technology. Poor management of this has, in some well-known instances, led to human error and even death (Besnard & Cacitti, 2005). The necessity for human adaptation to changing technology is intrinsic to any application of computer technology, from medical devices to software, and the stakes can often be high. For this reason, a clear understanding of how the visual design of the most salient portion of the system, the user interface, mediates the application of task knowledge is vital to the work of designers and engineers.

Operator error is often the unfortunate result of designing a system without consideration of this issue. In his seminal work on human error, Reason (1990) defines human error most generally as, “all those occasions in which a planned sequence of *mental* or *physical* activities fails to achieve its intended outcome, when these failures cannot be attributed to the intervention of some chance agency.” Byrne (2003) takes this definition further, emphasizing the point that errors are failures to meet a demand imposed by the task or tool. This definition suggests that the internal *and* external context of task performance of any error must be jointly considered in order to have a correct understanding of the root cause. This definition coincides with the assertions of traditional human factors regarding the importance of considering the entire system or context rather than the isolated actions of the operator (Chapanis, 1996).

Byrne's (2003) definition of human error is well-suited to explain a particular accident case occurring in 1990 during a night shift at the ASCOMETAL steelworks factory in France (Besnard & Cacitti, 2005). The operator had been working with eleven different thread drawing machines, one of which had its main operating buttons reversed on the interface. This anomalous machine was in fact the one involved in the accident, as the operator unintentionally opened the pressing wheels at a step in the process where this action was prohibited. Besnard and Cacitti (2005) observed that the error was not completely the fault of the operator, since on any of the other ten machines his actions would have been fine. Rather, it could be traced to the conjunction of several low-probability conditions, which in concert resulted in the tragic outcome of his death.

It is such matters of visual change to an interface and the resulting mismatches occurring between the operator and machine that are of concern in the work reported here. The general literature on transfer of knowledge as a phenomenon in cognitive psychology is longstanding (e.g., Detterman, 1993), but research specific to computer-based tasks is rather lacking. Of particular relevancy in such instances are the perceptual factors that come into play, such as spatial memory and the saliency of certain visual features, as operators develop skill on a computer-based task that primarily operates in the visual modality. How do these factors affect performance when the task structure and users goals remain fairly constant but visual features on the interface are changed?

Traditionally, the literature in the field has focused on the role of labels (e.g., Polson & Lewis, 1990) in computer-based tasks such as browsing the web. However, recent work by Byrne, Maurier, Fick, and Chung (2004) demonstrated that present theories of human error and skilled task performance might be inadequate. In their study, two routine procedural tasks also used in the work here were isomorphic at the abstract structural level but visually dissimilar and were found to produce different error rate profiles and step times. Such findings could not be predicted by extant theories of cognitive control such as GOMS (John & Kieras, 1996), standing for Goals, Operators, Methods, and Selection Rules, and Hierarchical Task Analysis (Annett & Duncan, 1967), which are based on the hierarchical decomposition of goals in a task. In fact the predictions of such theories would be near-identical performance. Computational modeling work further demonstrated a possible coupling of internal representation of task structure and the visual layout of an interface, thus calling for further investigation of how cognitive mechanisms (e.g., goal structures) interact with visual-cognitive mechanisms (e.g., search strategies) in these kinds of scenarios.

1.1. Procedural tasks and expertise

With routine procedural tasks, such as that of the ASCOMETAL factory operator, procedures can become so well practiced that they require little conscious thought beyond the initial step. Rasmussen's (1987) "Skill-Rule-Knowledge" (SRK) theory of task performance provides a general framework of cognitive control mechanisms to help explain this. 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 a person proceeds sequentially through the three stages of the model, moving from lowest (knowledge-based) to highest (skill-based).

At the rule-based level, failure modes may stem from either the application of bad rules or the misapplication of good rules due to incorrect rule selection. These rules may be active simultaneously, with several competing for instantiation. From the abstract model that the operator has constructed in their mind about a system, or mental model, rules for behavior are selected based on certain selection criteria listed below. These also control the occurrence of errors at the rule-based level, in keeping with Reason's (1990) idea of a "cognitive balance sheet," where correct performance and systematic errors are related like two opposing sides on a coin.

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.

Salminen and Tallberg (1996) note that with the 178 fatal occupational accidents reported in Finland between 1985 and 1990, the frequency of errors increased from knowledge-based level (least common) to the skill-based level (most common). Rule-based errors were also relatively frequent in occurrence, particularly among experienced workers as expected. Hence, the increase in performance from initial step-by-step interaction to "skilled" semi-automatic behavior brings with it a subsequent *increase* in the frequency of potentially *serious* human errors. In fact, with the 178 fatal accidents

Salminen and Tallberg (1996) report, more than half of the human errors to blame could be traced to the highest, skill-based level of performance.

These findings corroborate evidence that training cannot completely eliminate such low-frequency errors (e.g., Byrne & Davis, in press). Potentially dangerous industrial machines are designed with safety guards for this reason. For example, modern hydraulic presses used for sheet metal forming have non-tie down, anti-repeat controls requiring two buttons to be depressed together for operation, despite the highly repetitive (skill-based) nature of the work. This prevents the operator from accidentally triggering the machine by hitting a single button while one hand is still under the press. Reason's (1990) idea of the "cognitive balance sheet" seems to hold true here. This concept states that the same processes producing successful human performance are also the source of human error (Baars, 1992; Reason, 1990). Errors are simply the undesirable byproducts of the heuristic or rule-based reasoning we use, so rather than disappearing they change forms as familiarity with a task increases. For example, repetition of a behavior in training leads to automatic behavior as control is delegated to lower-level processes. However, this increases the opportunity for behavioral "slips" to occur, while the chance for knowledge-based errors decreases.

Besnard and Cacitti (2005) argue that the rule-based level is the domain of *transfer errors*, which occur when rules (procedural knowledge) from one task are inappropriately applied to another task. This claim seems acceptable given that schemata (Bartlett, 1932), or reusable high-level knowledge structures, operate at this level rather than at the skill-based level, where sensorimotor control is dominant. Again, this is in line with Reason's (1990) concept of the cognitive balance sheet. The application of schemata spares cognitive resources, as perception and action are directly related (Neisser, 1976); well-learned plans of action are triggered by the perception of salient and meaningful cues in the environment. It is for this reason that interfaces we consider *intuitive* are perhaps better considered as *familiar* (Raskin, 1994).

Incorrect perception of familiarity, however, can lead to transfer errors. In a series of three experiments conducted by Woltz, Gardner, and Bell (2000), transfer errors were found to occur in sequential, multi-step cognitive tasks when new processing sequences resembled old ones. Their task required participants to take four-digit numbers and apply a sequence of three rules (e.g., *different-same-different*) to them. For example, ‘3213’ would be reduced to ‘113’ by first applying the *different* rule (i.e., when two contiguous digits are different, they are reduced to the remaining digit: 32 = 1, 31 = 2; 12 = 3). Next, 113 would be reduced to 13 by applying the *same* rule (i.e., when two contiguous digits are the same, they are reduced to a single digit of that value: 33 = 3, 22 = 2, 11 = 1). Finally, the *different* rule would again be applied in this case to reduce 13 to 2. The performance of individuals trained on this task was compared with that of novice individuals on a similar but different experimental number reduction task. Tasks were made to differ by a change in the sequence (e.g., if they were trained on *same-different-same* and *different-same-different*, then they were tested on *different-same-same* or *same-different-different*).

As in their earlier work (Woltz, Bell, Kyllonen, & Gardner, 1996), they found that the more experienced participants initially committed errors of the *strong-but-wrong* type (Reason, 1990). This error type is synonymous with what Norman (1981) calls a *capture slip*, an error occurring when a strong habitual action sequence is incorrectly substituted for a related, weaker action sequence. Reason (1990) notes that these types of errors can occur at the rule-based and skill-based levels of performance. According to Woltz, et al. (2000), the fact that practiced participants made significantly faster responses (training caused latency data to decrease to a near asymptote, in accordance with the power law of practice) on the transfer task than the novice group but with more errors was evidence of the strong-but-wrong error type. They (Woltz, et al., 2000) also showed that these errors were solely sequential and occurred unconsciously (c.f., Anderson, 1982). Even when asked, the experienced participants were mostly unaware that they were committing these transfer errors when system feedback was removed. Even having explicit knowledge of familiar processing sequences did not facilitate performance on the transfer task.

Woltz, et al., (2000) present these results as empirical support for *negative transfer*, since participants with experience made more errors on the new sequence trials than did the novice participants. In their second experiment, however, they did notice that these negative transfer effects were influenced by the complexity of the task. Complexity, manipulated by an increase in the number of different sequences and number of instances of each sequence, seemed to decrease the frequency of transfer errors committed by the experienced participants. Thus, Woltz, et al. (2000) advise that the demonstrated effects on sequence knowledge may be less likely in complex real-world skills.

1.2. Transfer a function of abstract or surface features?

Interestingly, the results of Woltz, et al. (2000) are in contrast with those of Singley and Anderson (1985), who found transfer to be highly positive across dissimilar procedural tasks, which were in fact more complex and naturalistic (real-world text editors). Singley and Anderson (1989) note that transfer is more likely in procedural tasks when skill transfer conditions are highly overlapped with skill training conditions. Transfer has been proposed to be theoretically indistinguishable from learning, which is ubiquitous, since experts use transfer to acquire their expertise. Since tasks are rarely or never exactly the same, given the inconstancy of the environment, in almost all cases schemata are somehow being reapplied, even without conscious thought (e.g., Lewicki, Czyzewska, & Hoffman, 1987). Thus, returning to Neisser's (1976) explanation of schemata usage in direct exploratory actions, negative transfer or interference should occur when incorrect rules are selected due to perceptual (surface) similarity between two computer interfaces.

In the work of Singley and Anderson (1985), it was similarity between the training and test computer-based tasks at the abstract high-level goal structure responsible for transfer, since the tasks were largely different on the surface (commands). The tasks used in those experiments, however, were command line text editors that offer minimal external information. Hence, their findings may not apply to newer, more visual computer-based tasks (c.f., Byrne, et al., 2004) where visual search is much more important (e.g., Fleetwood & Byrne, in press). The analogical problem solving literature (e.g., Holyoak & Koh, 1987) suggests that people notice and reuse schemata when there are high levels of *surface* similarity in the information content of two problems. According to Sweller (1980), this tendency is justified, as much of the time there is a strong correlation between surface features of problems and their underlying abstract solution structures. This issue is not unrecognized in the field of HCI and design, although perhaps it is understudied.

Similarly, Kieras and Polson's (1985) framework for the formal analysis of Cognitive Complexity (CCT) in computer-based tasks emphasizes proper mapping between the user's task and device representations, which they proposed as the major components of the knowledge required to operate a device successfully. CCT proposes that certain task representations are independent of a device and can thus be transferred to a new device. Knowledge related to the device representation, on the other hand, can be divided into four categories: task-relevant knowledge, device layout knowledge, device behavior knowledge, and how-it-works knowledge. Device layout knowledge concerns the physical layout of the device, such as the location of controls, the format of the display, and the locations of various switches and status indicators and is of primary interest here, since it is knowledge of what is visible on the interface.

Analogical reasoning based on surface features has also been suggested to hold dominant responsibility for creative cognition and the development of expertise. Ball, Ormerod, and Morley (2004) made an inquiry into the use of analogizing in design contexts, by comparing the use of analogy by expert and novice designers. They make a distinction between what they call schema-driven analogizing (i.e., the recognition-primed application of abstract experiential or schema-based knowledge to a new problem), and case-driven analogizing (i.e., the mapping of a specific prior design problem to the current problem). Using think-aloud protocols and a task of designing an automated car-rental facility, they found results to support the claim that expert designers make greater use of schema-driven analogizing, in contrast to novices for whom the opposite was true. Most relevant to the work here, they emphasize the fact that the majority of case-driven analogizing for both experts and novices was dominated by the use of surface-level cues in the target problem, as opposed to the use of abstract cues in the underlying problem structure.

Returning to the definition of transfer errors of the strong-but-wrong type, Reason (1990) explains that they can result from perceptual confusions at the skill-based level. That is, they arise because recognition schemata are incorrectly triggered by perceptual cues appearing similar to or doing a comparable job as the expected object. Salient low-level physical features of an interface that are part of the device layout knowledge, such as the arrangement of buttons on a telephone, are quickly cued by our perceptual system. This can work both to our advantage and disadvantage when interacting with systems. It can facilitate training efforts and help us overcome learning curves, yet also hinder performance by sometimes leading to human errors. As previously discussed, Woltz, et al. (2000) demonstrated that these types of errors often occur without conscious awareness when processing sequences of new multi-step skills resemble old ones, particularly when working at high speeds. Those findings supported Reason's (1990, p. 97) claim that, "when cognitive operations are underspecified, they tend to default to contextually appropriate, high-frequency responses," and perhaps provide some explanation for real-world accidents, such as the ASCOMETAL case.

1.3. "Cognitive misers"

Our reliance on schema-based processing responsible for transfer errors is also largely tied to our general human tendency to be "cognitive misers," (Fiske & Taylor, 1991). That is, our inclination to take the road of least cognitive effort and use decision rules of thumb or heuristics rather than systematically analyzing each decision. Again, the problem solving literature provides substantial evidence to demonstrate 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, such as in traffic on the highway. Curiously, “shortcuts” taken by drivers often offer little or no real benefit, as cars moving to a “faster” lane may end up far behind cars remaining in the initially slower lane. Since the driver’s physical position in the car limits his or her perception and awareness to no more than the few cars in front, he or she is unable to see far enough ahead to plan or execute more efficient strategies of weaving through traffic. Likewise, with unfamiliar computer-based tasks, actions are often planned and executed based on the system state visible through the interface at a given moment in time.

Polson and Lewis (1990) uncovered this 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) demonstrated hillclimbing with 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 information visible on the device for place-keeping, which simply entails “knowing what parts of the task have been completed and what parts remain to be accomplished," (p.221).

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. Gray (2000) found that in this manner, VCR programming becomes a primarily perceptual rather than cognitive task. This finding is supported by the ideas of distributed representations (Zhang & Norman, 1994) and Rieman, Young, and Howes's (1996) dual-space model (external interface and the user's internal memory) of human-device interaction, where information is spread across the internal mind and the external interface. Zhang and Norman (1994) demonstrated that external representations, where the information is present on the device itself rather than in the head of the operator (internal), may drastically decrease and even eliminate human error. By constraining the problem space and exploiting more efficient perceptual processes and information in lieu of memory, as Gray (2000) found in VCR programming, information in the environment effectively reduces cognitive load.

1.4. Visual design to facilitate change

Vicente's (1992) concept of Ecological Interface Design (EID) focuses on enhancing and supporting worker adaptation to change and novelty in complex systems by leveraging the perceptual system. Providing functional information or contextual emergent features in the interface at a high level of abstraction, the EID framework posits, will allow users to make use of the more powerful perception-action capabilities versus resource demanding analytical processes (Hajdukiewicz & Vicente, 2002). Using emergent feature graphics is one means by which EID takes advantage of Rasmussen's (1985) Abstraction Hierarchy (AH), upon which EID is largely based. With AH, several levels of abstraction are used to represent a system's functional properties, from its physical form to its physical processes, and so forth. Functional information is thus presented alongside physical information about the system, leading to better performance in computer-based tasks (e.g., Janzen & Vicente, 1998). Hajdukiewicz and Vicente (2002) demonstrated that eventual disruption of these added higher-level emergent feature graphics had a stronger detrimental effect on user performance than did perturbation of the lower level existing physical information. The benefits of these added graphics, however, come primarily with complex and dynamic systems, where unexpected task disruptions are more likely to occur.

The suggestions of EID (Vicente, 1992) generally agree with Rasmussen's (1987) SRK theory. Both make direct reference to the benefits of using familiar external features in the environment, which allow for either the cueing of schemata or presentation of functional information. However, what are the specific visual features that people use to recognize similarity between computer interfaces and objects on an interface? The design community generally prescribes the use of such techniques as color schemes, button shapes and sizes, functional mapping, etc., which together are thought to provide internal consistency (and in turn, familiarity) for the design of an interface. The Gestalt Law of Prägnanz (Koffka, 1935), which subsumes such principles as grouping of visual elements by proximity, similarity, continuation, etc., may also offer explanation for how familiarity or the so-called usability metric of "intuitiveness" might be enhanced. Indeed, even the industrial design literature (see Ulrich & Eppinger, 1995) recommends the use of Gestalt principles in the design of products to enhance usability.

Nonetheless, the use of such grouping principles and visual features, which primarily relate to the perceptual system, cannot individually be considered to improve the general usability of a system. Chung and Byrne (in press) provide a concrete analytical demonstration that perceptual salience of a cue alone is an insufficient quality to prime correct behavior. What is needed is the activation of the correct memory by the external visual cue. This quality, *meaningfulness*, is one that is frequently discussed in the applied HCI field and in cognitive psychology (e.g., levels of processing; Craik & Lockhart, 1972), and is essentially tied to the same principle as *familiarity* (Raskin, 1994) with an interface. That is, a cue must be able to correctly and quickly trigger existing knowledge structures or schemata. This undoubtedly requires that some previous exposure or learning occurred, in order that these knowledge structures may be retrievable with future appearances of the cue.

Learning of constant external information, such as cues on a computer interface, has been demonstrated to occur gradually with practice (Somber, 1987). Ehret (2002) notes that there are two discrete parts to learning the locations of objects on an interface: the visual search for the objects themselves and the association and retrieval of related information. Location learning was demonstrated to be a relatively automatic process or by-product of visual attention in Ehret's (2002) experiments, much as acquisition of procedural knowledge can be implicit (e.g., Anderson, 1982). Moreover, the degree and precision of interface object location learning was found to vary depending on the cost of learning an interface. This cost of learning was manipulated using a variety of buttons placed in a similar spatial arrangement: colored button labels (least cost), meaningful button labels, arbitrary button labels, and no label (greatest cost) buttons. Essentially, precision, in terms of recalling the location of objects, and speed of acquiring this knowledge increased along with the cost of learning the labels. These results indicate that location is relied upon as a performance cue in direct manipulation interfaces only to the extent that the interface provides no less-effortful alternative. Moreover, they demonstrate the strong influence of surface features in the highly visual computer-based tasks of today.

How much of external information on an interface actually gets used is arbitrated by the amount of time saved, down to the level of milliseconds (Gray & Boehm-Davis, 2000). Recent work by Gray and Fu (2004) demonstrated the effects of least-effort in guiding interactive behavior with respect to how much of the information on the interface is actually used. They demonstrated how a system's *soft constraints* (in contrast to *hard constraints* that limit the possible actions on an interface) determine which patterns of interactive behavior are likely to be chosen and executed, through rational analysis of the interface (Anderson, 1990). Soft constraints in a software program using multiple windows were shown to encourage non-optimal reliance on imperfect, internal knowledge over external information present on the system. Surprisingly, this occurred even when the difference in effort between perceptual-motor and memory retrieval was small and higher error rates and lower performance resulted. This is somewhat unintuitive given the popular cognitive engineering perspective that knowledge-in-the-world is generally preferable to knowledge-in-the-head (e.g., Zhang & Norman, 1994).

Participants in their experiments (Gray & Fu, 2004) were assigned to one of three different conditions: a *Free-Access* condition (all necessary information was clearly visible on the screen), a *Gray-Box* condition (a mouse click was necessary to uncover the information), and a *Memory-Test* condition (information was well-learned and tested prior to the trial). Their prediction, generated through multiple analyses with a cognitive engineering tool that estimates the amount of *time* required for perception, action, and memory retrieval (e.g., CPM-GOMS; John & Kieras, 1996), was that participants in the *Memory-Test* condition would perform the best, followed by *Free-Access*, and then *Gray-Box*. The data for both of the performance measures (trials-to-criterion and goal suspensions) confirmed this prediction. Moreover, participants in the Free-Access and Gray-Box groups made far fewer information accesses than expected, further suggesting that time spent for memory retrievals was weighed nearly the same as time spent for perceptual-motor activity.

Both strategy and performance were thus affected by the perceptual-motor demands of the system interface. Gray and Fu (2004) suggest that this preference for imperfect knowledge-in-the-head would be heightened in more complex systems with greater perceptual-motor demands. These findings again highlight the need for fine-grained cognitive engineering analyses of how people actually use visual information present on a computer interface, particularly in safety critical and complex situations of human-computer interaction.

1.5. Error data banks

Supporting human error prediction across computer-based tasks will require “data banks” of human error (Rasmussen, 1987). Sufficient error data would allow for the development of specific design guidelines to reduce error, such as in instances where a visual interface is being redesigned. Chung, Zhang, Johnson, and Turley (2004) demonstrated that reuse of visual interfaces by device manufacturers even when the underlying task structure is different can negatively affect performance. Fortunately, collecting human error data has been shown to be possible in several of the works described here (e.g., Woltz, et al., 2000; Chung & Byrne, in press). The HCI domain holds great promise for the study of human error, as software programs provide an easily manipulated stimulus to elicit and capture error.

Since computer-based tasks today are primarily visual, by altering only the visual interface of a system it is possible to immediately change users' interactive strategies and patterns of error, as several of the experiments described here have done. By looking at these differences, predictive principles founded on cognitive theory and quantitative data can be generated to arm designers and evaluators. This would not only improve our post hoc evaluations of systems but also enable the design of safer and easier to learn computer interfaces. As emphasized in previous research (e.g., Byrne et al., 2004; Gray & Fu, 2004), such analyses will need to be fine-grained to determine how the visual components of an interface affect user performance. Typical usability quick fixes such as simply loading information on the external interface (knowledge-in-the-world) to reduce workload have been shown to not always generate the intended effect.

2. Lab Experiments

As prior research in our laboratory has demonstrated, the manner in which user behavior may be affected by the visual design is often unintuitive (e.g., Chung & Byrne, in press). Proper understanding of the influence of visual design on user performance requires careful consideration of the task it is being used for, the user's previous knowledge, and the environment it is being used in. This general idea is widely accepted by usability practitioners in industry, as exemplified by established design practices, such as Contextual Design (Beyer & Holtzblatt, 1998) and Scenario-based Design (Carroll, 1995). Failure to consider these factors when redesigning an interface can potentially lead to such disasters as the ASCOMETAL incident, which can be attributed to a transfer of nonoptimal methods or procedures (Singley & Anderson, 1989). While it has been shown that skilled performance overall improves despite any initial interference when learning a similar task, inappropriate transfer caused by visual similarities can lead to catastrophic consequences with even a single occurrence under the wrong conditions.

2.1. Experiment 1

The purpose of this first experiment was to explore what device knowledge users come to rely upon, after developing a certain level of familiarity or skill with a given interface. As described in the introduction, Byrne et al. (2004) have reported differences in user performance attributable to the design of a system beyond the influence of the underlying task structure alone. To gain a better understanding of how this may happen, it is requisite to determine what visual information users are leveraging to complete their tasks. One way to do this is by changing two different sources of information on the interface partway through an experiment session and comparing the resulting detriments in performance. Three computer-based tasks, used in our lab for previous experiments (e.g., Byrne & Davis, in press; Chung & Byrne, in press), provide a well-tested environment in which to conduct this type of study.

The two computer-based tasks used to test the manipulations were the Phaser task detailed in Figure 1 and the Transporter task presented in Figure 2. As can be seen from the figures, the two tasks are similar, with the same number of steps and subgoals and controls arranged in groups. Nonetheless, they are not completely isomorphic, as the results reported by Byrne et al. (2004) showed. During a typical experiment, participants access both tasks and an additional filler task (Navigation) through a Main Control interface, as prompted by the system. Interaction is mainly done with the mouse, although there are some keys to be pressed (e.g., spacebar and arrow keys for the tracking and firing portions of the Phaser task and number keys to enter frequencies in the Transporter).

Goals	Steps
1st Subgoal	1. Power Connected
	2. Charge
	3. Stop Charging
	4. Power Connected
2nd Subgoal	5. Settings
	6. <Slider>
	7. Focus Set
3rd Subgoal	8. Firing
	9. Tracking
	10. <Tracking Task>
	10. Shoot (Spacebar)
4th Subgoal	11. Tracking
	12. Main Control

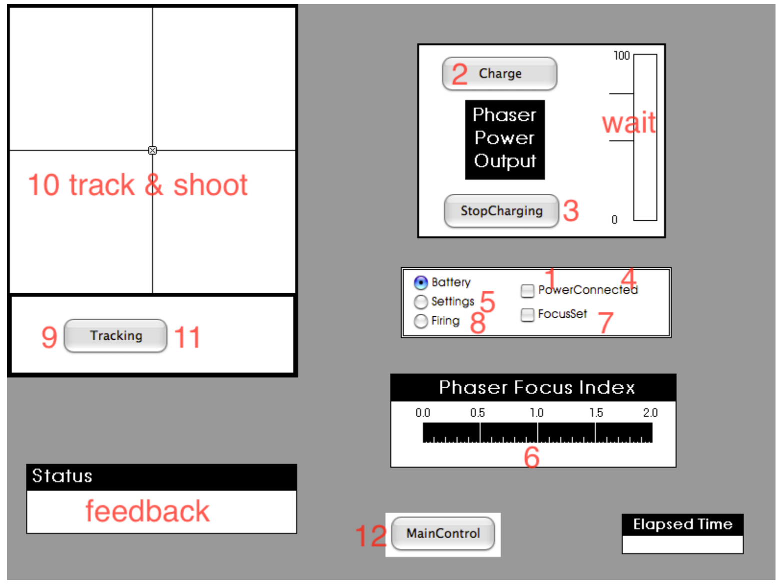


Figure 1. Task goal hierarchy and screenshot of the standard version of the Phaser task. The numbers and labels ("feedback," "wait," and "track & shoot") shown by the controls are not visible on the interface during the task.

Goals	Steps
1st Subgoal	1. Scanner On
	2. Active Scan
	3. Lock Signal
	4. Scanner Off
2nd Subgoal	5. Enter Frequency
	6. <Type>
	7. Accept Frequency
3rd Subgoal	8. Transporter Power
	9. Synchronous Mode
	10. <Tracking Task>
	10. Shoot (Mouse)
4th Subgoal	11. Synchronous Mode
	12. Main Control

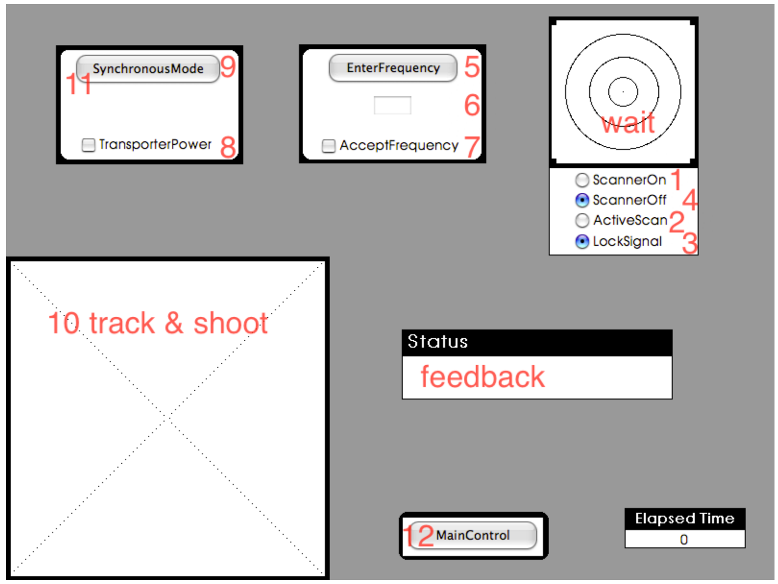


Figure 2. Task goal hierarchy and screenshot of the standard version of the Transporter task. The numbers and labels ("feedback," "wait," and "track & shoot") shown by the controls are not visible on the interface during the task.

Traditional usability evaluations of user interfaces have primarily focused on the most salient component of the user interface for a novice user, the labels (e.g., Polson & Lewis, 1990). As described earlier, Ehret (2002) demonstrated the importance of label quality in determining how much users come to rely on them. Less work, however, has been done to study how the visual design (e.g., the ordering and grouping of controls) of the interface affects user performance, particularly in routine procedural tasks. Unpublished data collected in our lab (Byrne, Chung, & Fick, 2004) has shown that simply changing the ordering or grouping of buttons on an interface can have noticeable effects on performance. Specifically, changing the layout of an interface to meet both global (e.g., top-to-bottom) and local user expectations (e.g., following the ordering of buttons in the previous button grouping) can reduce or virtually eliminate errors occurring at the boundary between two subgoals in a task. Tamborello and Byrne (2005) have also demonstrated that in visual search tasks good labels cannot overcome poor visual cues, and that both factors are at least equally important.

It is thus evident that visual search, based on a user's existing knowledge and expectations (i.e., top-down), can significantly influence performance in routine procedural tasks. The top-down influence has been similarly demonstrated in studies on the optimal placement of links in web design (Oulasvirta, Kärkkäinen, & Laarni, 2004). Thus, the purpose of this first experiment was to answer the question: how much are users guided in routine procedural computer-based tasks by the visual layout of an interface versus label information? To determine how users rely on these two sources of information, two different changes were introduced in a task halfway through the testing session. A layout change condition was used to assess reliance on the local ordering of objects on the interface (Figures 3a and 3b). Conversely, in similar fashion to the work by Ehret (2002), the semantic information was eliminated by crossing out the labels in a second condition, forcing users to continue with the task using only the unchanged graphical information on the interface (Figures 4a and 4b).

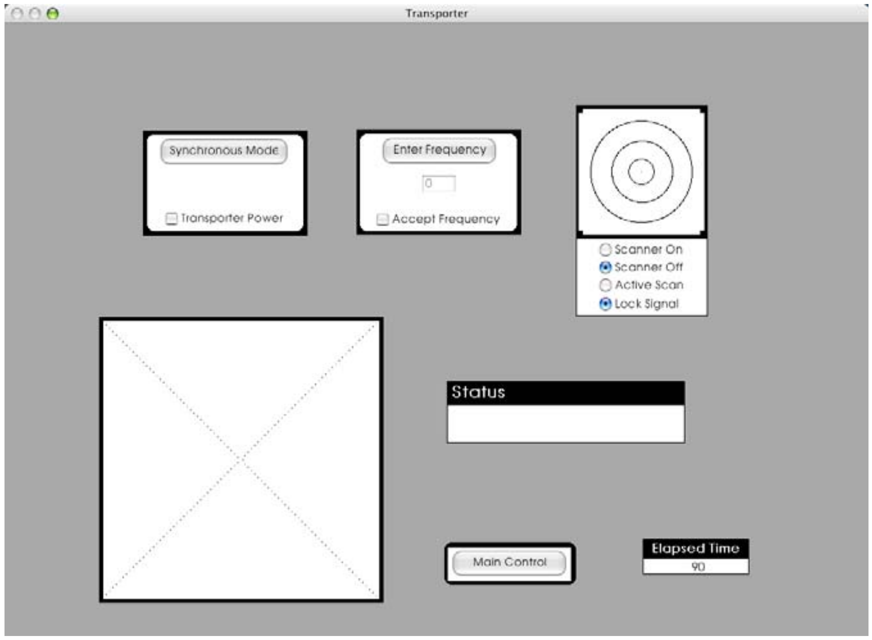


Figure 3a. Layout change: Pre-change Transporter interface.

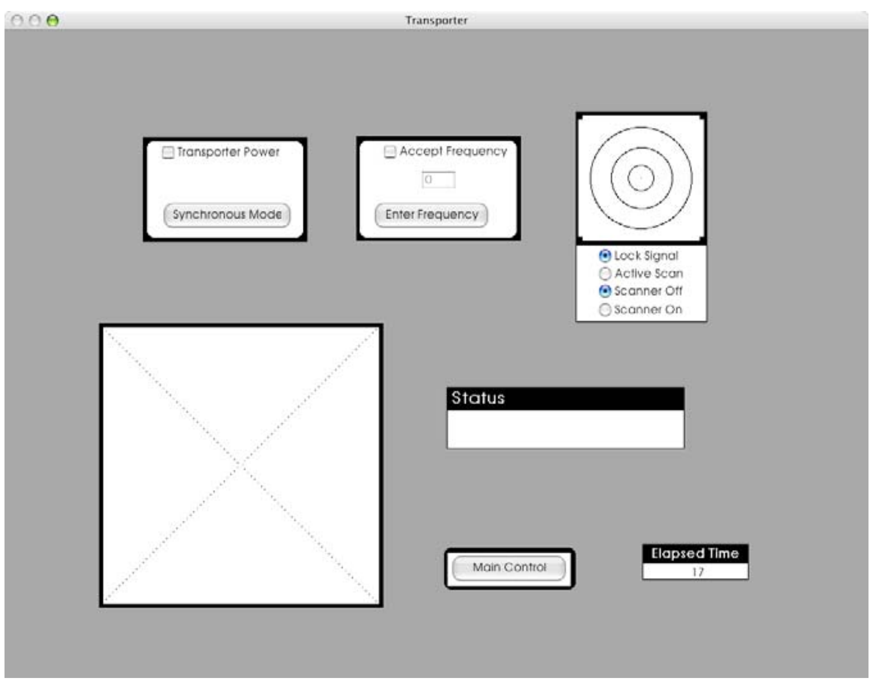


Figure 3b. Layout change: Post-change Transporter interface.

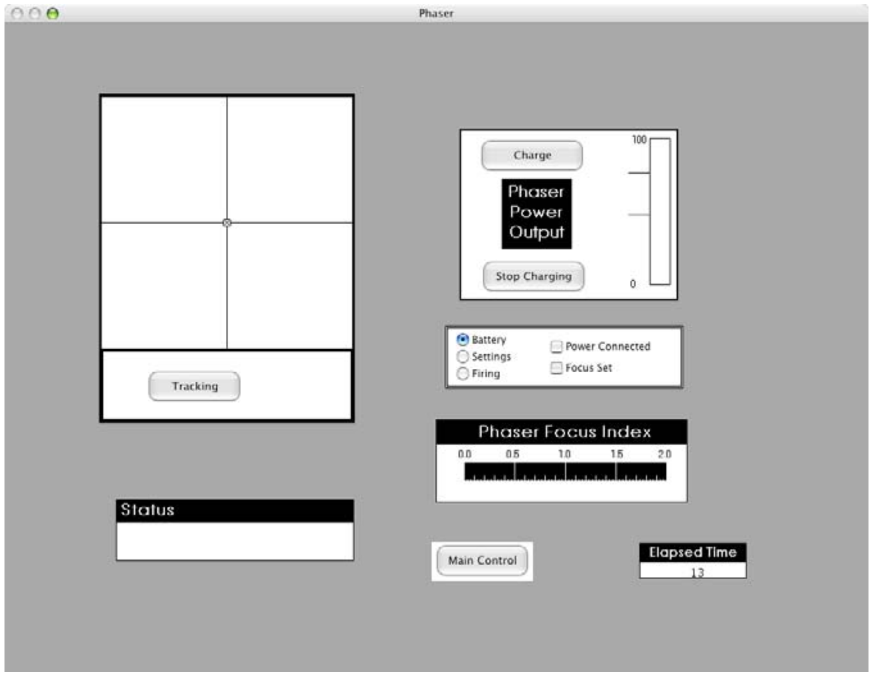


Figure 4a. Label removal: Pre-change Phaser interface.

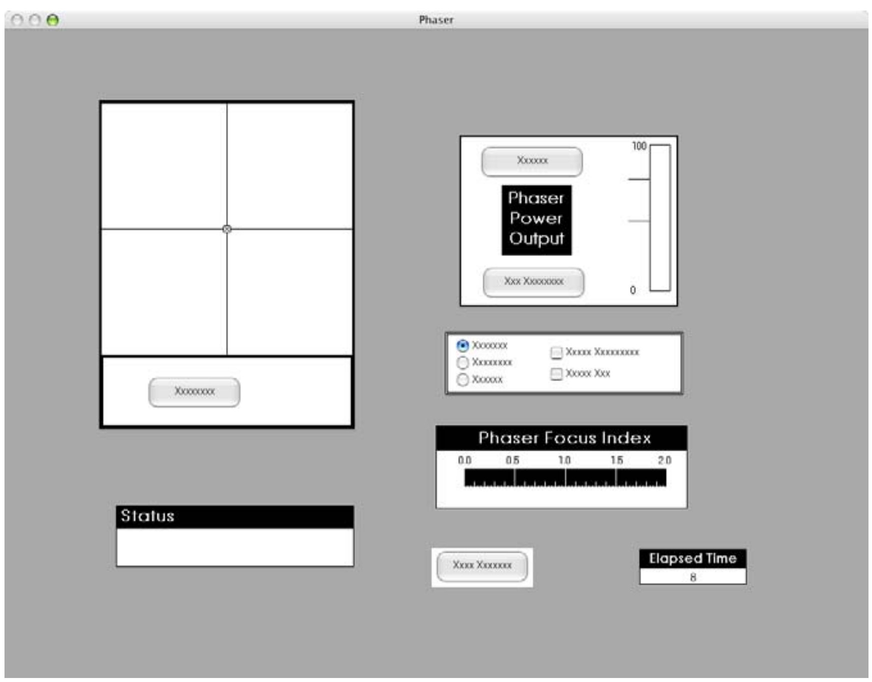


Figure 4b. Label removal: Post-change Phaser interface.

2.1.1. Method

2.1.1.1. Participants

Thirty-one Rice University undergraduate and graduate students (16 female, average age 23.9 years) participated in this experiment for partial course credit or monetary compensation ($10 for the first session, $15 for the second session) and cash prizes ($10, $15 and $25 for top three performers). All participants had normal or corrected vision and were naive to the tasks used in this experiment.

2.1.1.2. Design

Experiment 1 used a 2 x 2 mixed design, with a within-participants variable of time (pre- and post-change), and a between-participants variable of change (label removal or layout change). Participants were randomly assigned one of the two groups. The primary dependent variables were step times and error frequencies. Step time was the time taken to complete each correct step in the task, recorded in milliseconds. Dividing the total number of errors committed by the total number of opportunities for error (or the total number of times the participant executed that step) provided the other primary dependent variable of error frequency for each step. Errors at each step were only counted once per occurrence, with a separate record for multiple errors at the same step, which was not included in this analysis. For steps of interest, errors recorded in the post-layout change condition were classified by what button was incorrectly pressed.

2.1.1.3. Materials

The experiment application written in Macintosh Common Lisp and web-based demographic questionnaire were displayed in Microsoft Internet Explorer running on Apple eMac computers with Mac OS X. Other materials included a standard Apple single-button mouse, standard Apple QWERTY keyboard, and Sony MDR-201 headphones. Paper instruction manuals were provided for each of the tasks (Main Control, Navigation, Transporter, and Phaser), and a printout of the prize and point scale was shown at the second session (Appendix A).

2.1.1.4. Procedure

Participants were run in two sessions spaced four to ten days apart. The first session served as a training session using written documentation for each of the tasks: Navigation, Phaser, and Transporter. Order of presentation and condition were randomized across participants. Once the initial training trial was successfully completed using the manual, participants were required to log three subsequent error-free trials before moving on to the next task. Errors resulted in warning beeps and messages, ejection to the Main Control, and a restart of 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.

The second session consisted of test trials for the three tasks. Participants completed fourteen trials for each of the target tasks (Phaser and Transporter) and 11 trials of the Navigation task, for a total of 39 randomized trials for the test session. 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 continued until the task goal was successfully met. A general reminder about the change condition was presented onscreen at the beginning of the testing session and in the training manuals. After seven trials of the target task, participants were warned more specifically ("labels will be XXXed out" or "system controls in the transporter task will now be rearranged"), immediately before the change occurred. They then completed seven more trials of the target task in the post-change condition, still interspersed with the other tasks.

The concurrent working memory letter task was also introduced on the day of testing. As in previous studies by Byrne and Davis (in press) and Byrne and Bovair (1997), its function was to increase working memory load during task performance and raise error frequencies. 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 at random 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 appearing on the screen. This was the same for both conditions.

Participants were encouraged to work both accurately and quickly by means of an onscreen scoring system, an onscreen timer, and cash prizes for the top three performers, as determined by their accumulated points. 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 specified period of time, as made known to the participant on a prize and point scale presented at the beginning the session. For every incorrect working memory recall trial, a tone sounded and the score was decremented 200 points. No points were added for successfully completing a recall trial. Accumulated points were used in competition for cash prizes of $10, $15, and $25, awarded to the top three performers.

Finally, a web-based demographic questionnaire appeared onscreen after the completion of the last trial at testing. After completion of this questionnaire, the researcher debriefed the participant.

2.1.1.5. Theoretical expectations

Traditionally, the HCI literature (e.g., Polson & Lewis, 1990) has suggested that new users learn to perform computer-based tasks using the labels. The findings of Ehret (2002) provided additional reason to believe that the participants would rely largely on the labels, since they were not obscure and were explicitly described in the manuals at training. Furthermore, the manuals presented the tasks to participants in a hierarchical form (Appendix A), in accordance to the ideas of theories such as GOMS (John & Kieras, 1996). Hence, it was expected that label removal would lead to worse performance.

2.1.2. Results

Of the thirty-three participants in this experiment, three were removed due to technical issues with the experimental program (crash and incorrectly saved data) and another after discovering previous participation in a similar experiment. From the remaining twenty-nine participants, one more was removed as an outlier. This was determined using the criterion of possessing three or more extreme (3+ inter-quartile range) data points on pre- and post- change error frequencies or reaction times at one of the steps in the target task (Phaser or Transporter). This led to the final breakdown of fourteen participants in the Phaser label removal group and fourteen in the Transporter layout change. Nine extreme error frequencies (>70%) and step times (>10,000 ms) were replaced with the grand mean for that step. Finally, median trial data was used for each participant to mediate the effect of outliers.

To assess the influence of the change conditions on task performance, only data from the last three of the seven pre-change trials and first three of the post-change trials were used in the analyses. Pre-change trial data was restricted to the last three trials because it was found in previous experiments that participants generally required a trial or two to remember the task fully and return to the previous skilled level of performance reached at the end of the training session. Since the primary goal of this study was to assess the immediate impact of change on the participants' performance, analysis of the post-change trials was also restricted to the first three.

Only the first ten of twelve steps were considered in this analysis, since only they were truly comparable across tasks. Looking at Figure 5 it is immediately apparent that removing the labels had a negative effect in terms of error frequency to some degree at six of the ten steps in the task. An analysis of variance run with the within-participants variables of change (pre and post) and step (1-10) for error frequencies in the Phaser label removal condition, however, showed the effect of change to be unreliable, $F(1, 13) = 1.29$, $p = .28$. The change by step interaction was also non-significant, $F(3.9, 50.2) = 1.38$, $p = .26$. Nevertheless, participants' mean error frequency at the first step increased more than 10% after the change, perhaps reflecting a greater importance of the label at the first step in a task. Statistically, however, this difference was not reliable, *t*(19.4) = -1.72, *p* = .10. In terms of the specific errors being made at the first step after the labels were removed, the "Charge" button (step 2) was incorrectly pressed seven times by all participants in total, while "Tracking" was incorrectly pressed twice. However, the error data from the last three pre-change trials also shows that "Charge" led to 2 errors and 3 for "Tracking," and in the initial four trials of the testing session, participants had 9 errors with "Charge" and 10 for "Tracking."

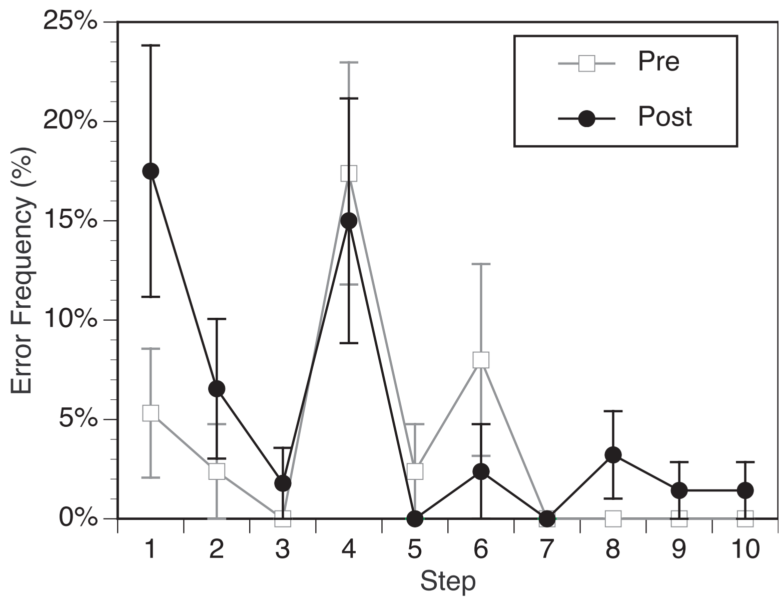


Figure 5. Pre and post label removal error frequencies (%) by step for the Phaser task. Bars represent standard error of the mean.

For step times, or the amount of time (in milliseconds) taken to complete a single step in the task, only steps 1, 2, 4, 5, 7, 8, and 9 were included in the analysis of variance, as the other steps did not completely rely on the participants' performance. That is, the excluded steps required the participant to wait on the system. Again, it seems only the first step was clearly affected by the change. The main effect of change, $F(1, 13) = 1.55$, $p = .24$, was found to be unreliable. However, the change by step interaction was significant, $F(2.1, 27.4) = 3.80$, $p = .03$, primarily driven by the noticeably slower first step in the post-change state, $t(18.6) = -2.31$, $p = .03$, which was again non-significant with a Bonferroni correction for multiple comparisons.

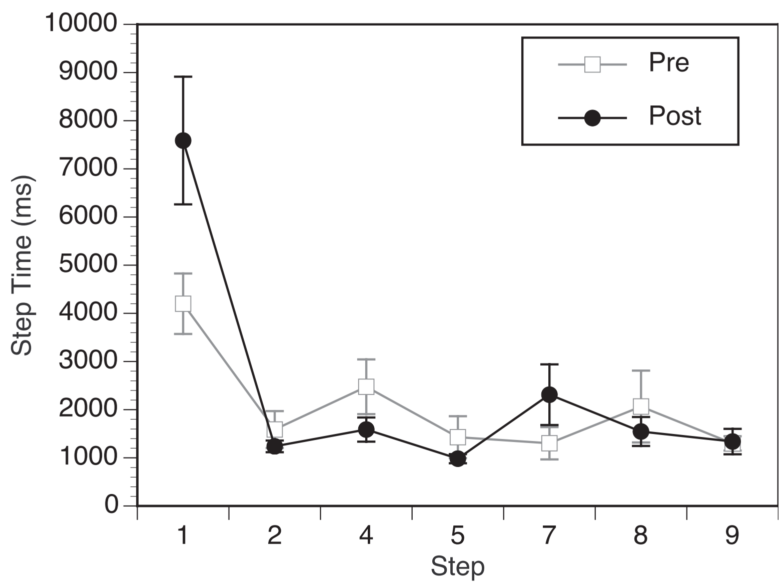


Figure 6. Pre and post label removal step times (ms) by step for the Phaser task. Bars represent standard error of the mean.

For the layout change condition run on the Transporter task, the effect on error frequency was readily apparent (Figure 7). Both the effect of change, $F(1, 13) = 8.25$, $p = .01$, and the change by step interaction, $F(3.7, 47.7) = 3.36$, $p = .02$, were significant. Most strikingly, there was a very large effect (>20% increase) of layout change on error frequency at the first step, $t(13) = -3.68$, $p < .01$, significant with the Bonferroni correction for multiple comparisons. As for the errors being committed at this first step, incorrectly pressing the "Active Scan" button (step 2) led to sixteen of them, and "Lock Signal" (step 3) was pressed once.

The main effect of layout change, however, was non-significant in the analysis of variance for step times (Figure 8), $F(1, 13) = 0.66$, $p = .43$, and the change by step interaction also failed to reach statistical significance, $F(2.4, 30.7) = 2.17$, $p = .12$.

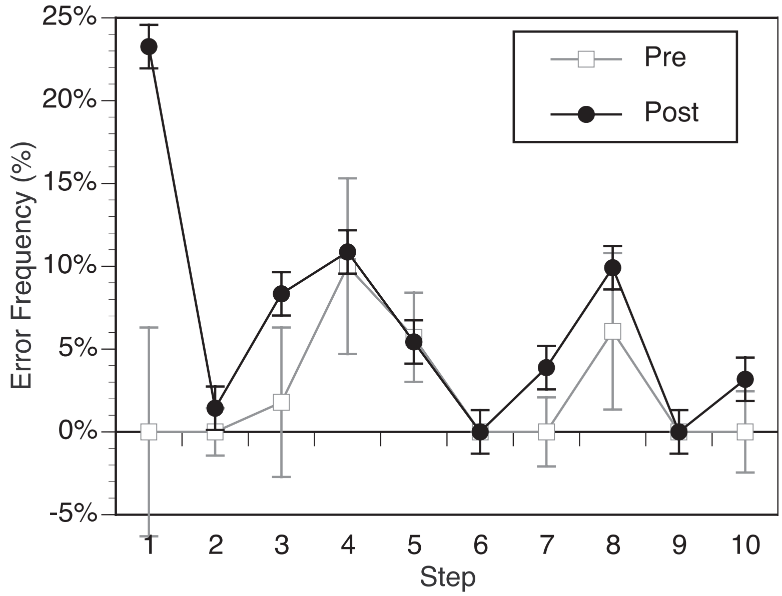


Figure 7. Pre and post layout change error frequencies (%) by step for the Transporter task. Bars represent standard error of the mean.

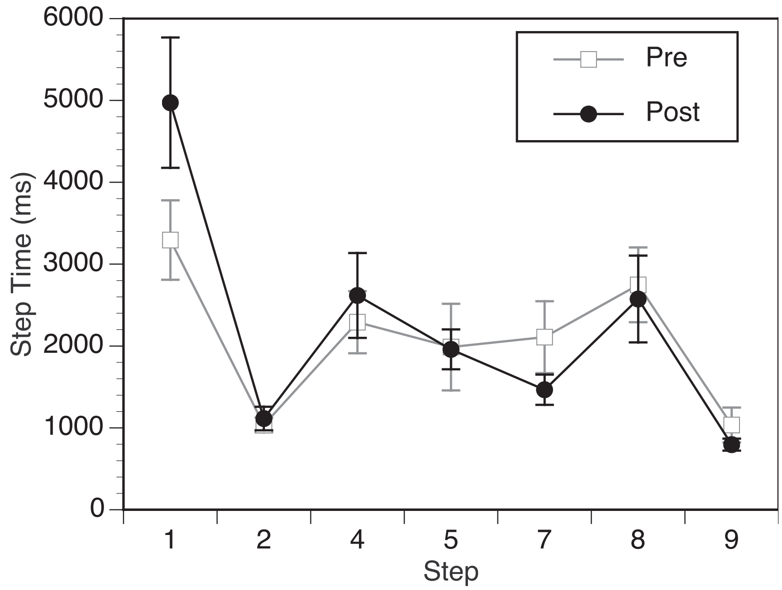


Figure 8. Pre and post layout change step times (ms) by step for the Transporter task. Bars represent standard error of the mean.

Next, difference scores were computed by subtracting post-change data from pre-change data for both error frequency and step times. This was done to allow comparison across the two different manipulations of label removal and layout change, since each was implemented in a similar but different task (i.e., Transporter and Phaser). Since the difference score was calculated by subtracting the pre-change data from the post-change data, a higher error frequency or step time indicates worse performance after the change. Layout change negatively affected error frequency more than did label removal (Figure 9) for seven of the ten steps, while the disparity in step time difference scores was mixed (Figure 10). Analysis of variance for error frequency, $F(1, 26) = 2.25$, $p = .15$, and step time, $F(1, 26) = 2.21$, $p = .58$, showed a non-significant effect of condition. Step 1 was also higher for the layout change condition in terms of post-change error frequency, although for step times, it was less affected.

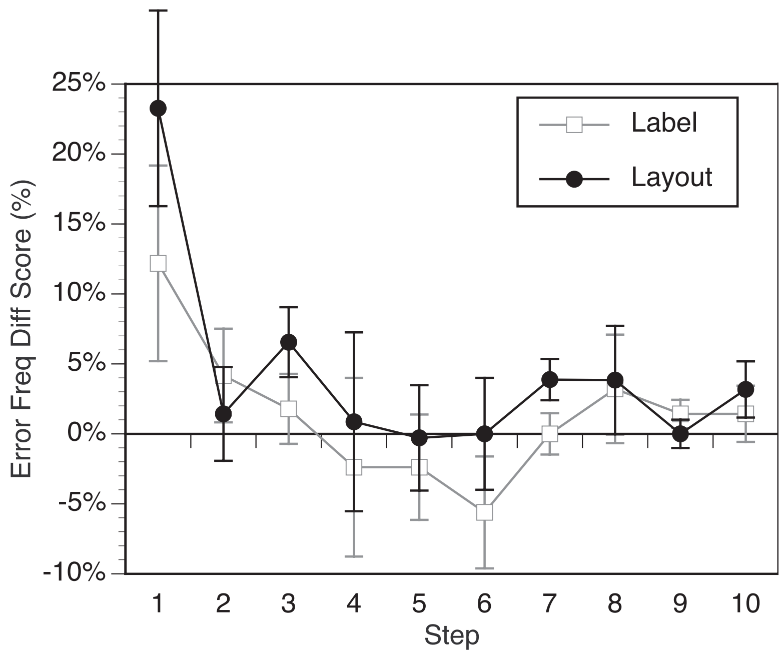


Figure 9. Error frequency difference scores (post - pre) by step for the label removal (Phaser) and layout change (Transporter) conditions. Bars represent standard error of the mean.

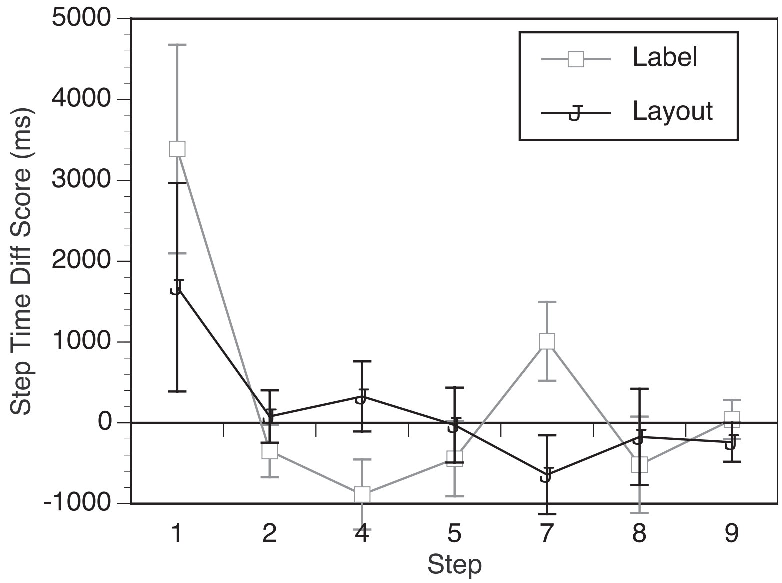


Figure 10. Step time difference scores (post - pre) by step for the label removal (Phaser) and layout change (Transporter) conditions. Bars represent standard error of the mean.

2.1.3. Discussion of Experiment 1

The results indicated that changing the layout of the interface had a more detrimental effect on participant performance than did removing the labels. Given the reliable main effect of change and change by step interaction for error frequency in the Transporter layout change condition, it seems that the layout of an interface has a definite influence on task performance in these types of routine procedural computer-based tasks. That is, with the development of some skill, in this case a minimum of 11 trials (4 for training and 7 for testing, pre-change), memory for the locations of objects on the interface becomes heavily relied upon. As noted, performance on the first step seems particularly susceptible to disruption layout change and even somewhat by the removal of the labels.

The strong effect of label removal on the first step of the Transporter task should be addressed. For most other steps, performance actually continued to improve even after the labels were removed. However, on the first step, participants incorrectly pressed "Charge" (step 2) a total of seven times after the labels were removed. If the greater detrimental effect of layout change on performance can suggest that participants were relying on memory for location more than the labels, then it seems strange to find that the label removal condition led to several errors, where participants were clicking on a button located far from the proper location of the step 1 control. However, as reported, the error data from the previous trials provide an explanation.

The error data at the first step in the layout change condition were more interesting. As noted in the results, 16 of the errors committed at the first step in the post-change state were due to incorrectly clicking the "Active Scan" (step 2) button. It is important to note that this error-prone "Active Scan" (step 2) radio button replaced "Scanner On" (step 1) in the post-change state, as the top-most unselected radio button in the cluster of four radio buttons. "Scanner Off" (step 4), which is already initially selected (i.e., the radio button is filled in) at the beginning of the task, replaced "Scanner On" (step 1) at the top of the cluster. For the last three pre-change trials in the layout change condition, as presented in the results there were no errors made, and "Active Scan" can only be found to have been the cause of an error at the first step once in the first four pre-change trials. Clearly, there is some effect of memory for locations that led participants to click "Active Scan" after the change occurred.

The error frequency difference scores (Figure 9) also fell above or at zero (except for step 6) for every step in the layout change condition, indicating that performance in terms of errors hardly improved and at many steps became worse. This is in contrast to the label removal condition, in which three of the steps showed improvement in terms of error frequency even after the change, suggesting that further development of skilled performance at that point was not affected by the absence of labels. Since preventing errors is the primary focus of the present work, this is important to note.

Finally, the lack of huge effects generated by the change manipulations is in itself quite interesting. Despite the change manipulations, participants were still able to resume and proceed with the task at a high level of skill, as reflected by the lack of significant change in step times. This is reminiscent of the ASCOMETAL case, in which the skilled factory worker was able to operate the anomalous machine for the most part without problems, although he mistakenly pressed the wrong button on the reversed interface (Besnard & Cacitti, 2005). Moreover, it supports the idea of separate task and device representations (Polson & Kieras, 1985). The remapping or transfer of procedural knowledge to a new interface seems to be carried out fairly quickly, even when one has much experience invested in the original interface.

Designers thus need to carefully consider that users who are familiar with a task will not to slow down their interaction despite an interface change. This of course can be dangerous, particularly if a change is applied to the layout. Transfer errors were found by Woltz et al. (2000) to occur without conscious awareness, particularly when working at high speeds. Moreover, as in the ASCOMETAL case, any such low-frequency error can lead to a fatal outcome. The reduction of such errors, caused by changes to an interface, will be welcome in certain scenarios, such as with upgrades to a critical interface used at low frequencies (e.g. an ATM or a medical device). Since the results of this first experiment indicated that layout change had a seemingly greater effect on error frequency than did label removal, it was the focus of more in-depth study in Experiment 2.

2.2. Experiment 2

To better understand how layout information is utilized and to assess how changing it affects performance, two new manipulations were introduced in this follow-up experiment. Unpublished research conducted in our laboratory (see Byrne et al., 2004) has shown errors to increase or decrease in a routine procedural task, as a function of the differential ordering of objects on the interface, for example. Errors seem to decrease when the interface is designed in a way that is in line with the user's existing knowledge of the system, such as by ordering the required buttons within a subgoal group so they are consistent with the ordering of the previous subgoal group. Also, arranging the controls to follow a single universal rule, such as top-to-bottom, has also been shown to reduce error frequencies.

Most common usability guidelines and research assert the importance of preserving consistency in the layout of an interface, such as within a website (e.g., U.S. Dept. of Health and Human Services, 2003). It thus seems possible that the visual design of the interface can be managed to facilitate transfer of a procedural skill, by taking advantage of strong, existing knowledge rules, thereby eliminating the necessity of learning completely new locations. In a largely visual computer-based task, reorienting oneself to a new interface for a constant task would require the remapping of previously acquired procedural knowledge to adjust to new object locations. Having to reorient oneself to the locations of controls (e.g., windshield wiper controls, emergency lights, etc.) in an unfamiliar rental car is a real-world example of this. It was therefore considered that ordering objects on an interface in such a way that is consistent with preexisting knowledge in the form of a simple rule (e.g., top-to-bottom order) would facilitate the transfer of procedural skill. Instead of having to learn entirely new locations for each of the controls, participants could rely on a single rule, thereby shortening the process of remapping procedural task knowledge to the new locations of objects on the interface.

Similar to Experiment 1, participants in Experiment 2 began with the original version of the Transporter task, and a change was introduced to the layout halfway through the experiment. However, in contrast to the layout change manipulation in Experiment 1, the post-change layout here followed the global expectation of top-to-bottom reading order. Buttons within each of the three subgoal groups were ordered from top to bottom, or down-down-down (DDD) in adherence to global expectations (Figure 11). As previously mentioned, research done in our lab has shown this layout to reliably reduce errors occurring at certain steps within the Transporter task. Hence, it was expected that this manipulation would reduce the pre- to post-change differences in performance (errors and step times) relative to ordering the buttons from bottom to top in each subgoal, which does not adhere to such global expectations.

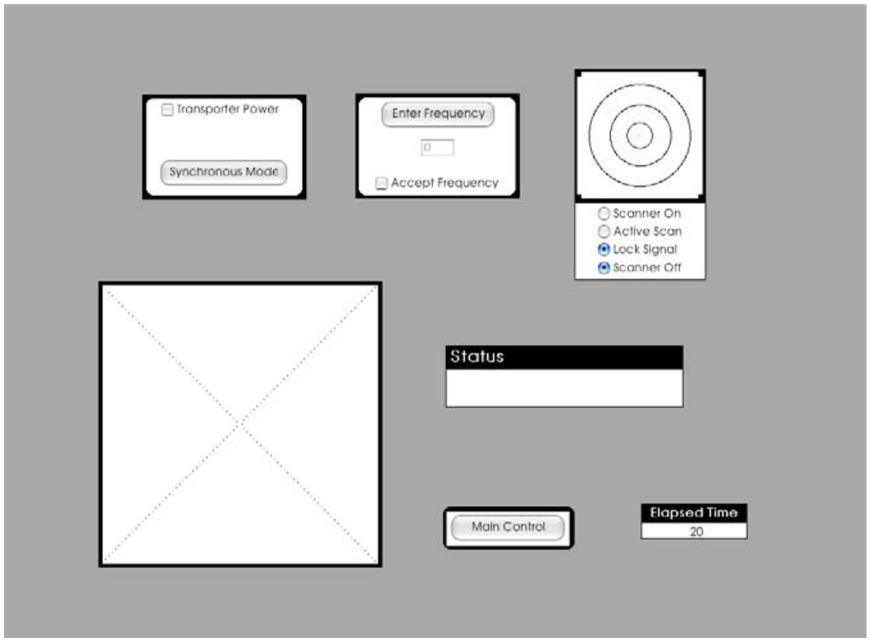


Figure 11. Top-to-bottom (DDD) post-change version of the Transporter.

It was also important to replicate the findings for each condition in Experiment 1 for the opposite task, since the two tasks (Transporter or Phaser) are not exactly the same. For example, the layout of the Phaser task is such that several of the buttons are more similar in appearance and closely grouped than those in the Transporter. Moreover, the Phaser task seems to be slightly more difficult for participants, as it requires the user to go back and forth more across groups of controls. The label removal condition was thus applied in the Transporter task (Figure 12), and the layout change in the Phaser task (Figure 13) to again investigate users' sensitivity to changes in location in contrast to label information, as in Experiment 1.

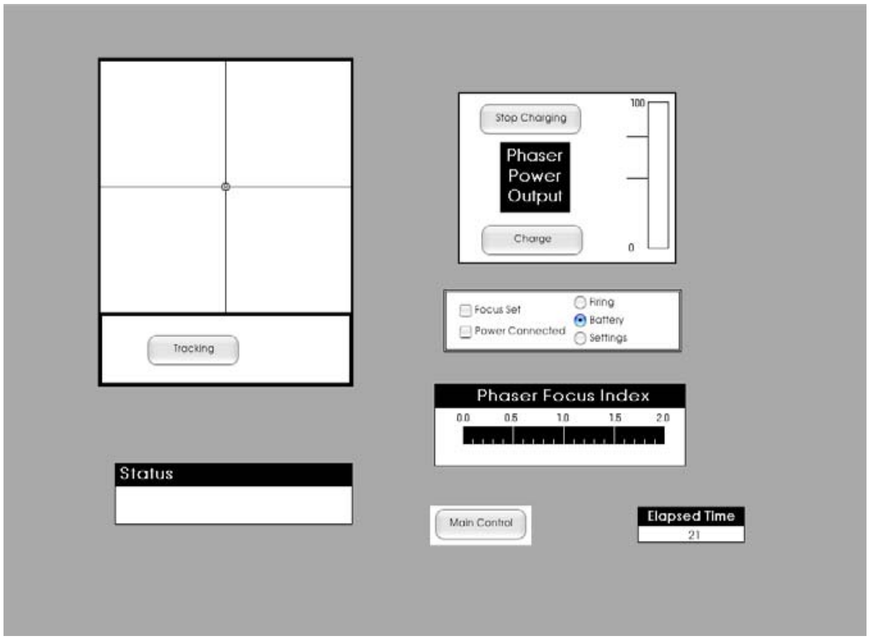


Figure 12. Layout change post-change version of the Transporter.

Pre-change state is the same as in Experiment 1.

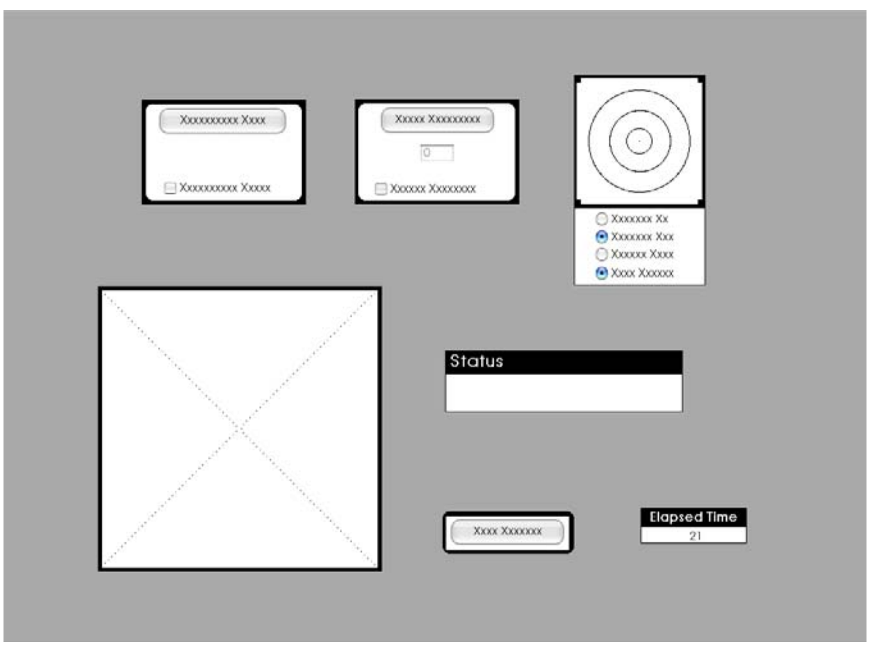


Figure 13. Label removal post-change version of the Transporter.

Pre-change state is the same as in Experiment 1.

A third condition was implemented to examine the effects of another important component of interface design, color. A vital component of visual recognition memory, it has been found that objects such as photographs are remembered better if presented in color versus black-and-white (e.g., Homa & Viera, 1988). Many television remote controls and machine interfaces in the real world feature color labeled controls. Even the Microsoft Xbox game console features color labeled buttons on the controllers, which are consistent through the first and second-generation systems (Figure 14).

Figure 14. Microsoft Xbox original (left) and Xbox 360 (right) controls.

Participants started the task with what was essentially the original version of the Phaser task, although each of the controls related to a subgoal in the task was associated using a unique color (Figure 15). Both buttons related to the battery ("Battery" and "Power Connected") were given a unique color (green) that was the same as the color of the battery-charging window. Similarly, both buttons related to the "Phaser Focus Index" ("Setting" and "Focus Set") were given the same color (blue) as the "Phaser Focus Index" itself. Finally, the "Firing" and "Tracking" buttons, both related to the subgoal of firing the phaser, were both colored orange.

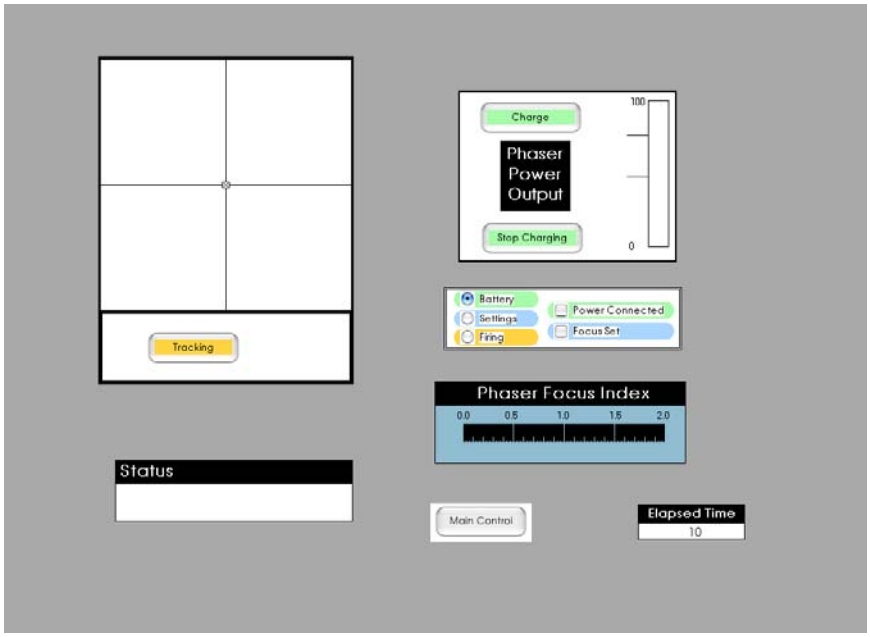


Figure 15. Color pre-change version of the Phaser interface.

A layout change was introduced halfway through testing, but the rearranged buttons retained their original color labeling (Figure 15), as explicitly explained in the manuals at training. Since data from previous experiments in our laboratory have shown that participants seem to retain the hierarchical structure of these tasks (Byrne et al., 2004), the colors were expected to help participants quickly distinguish the buttons related by subgoal and adapt to their new locations with the changed layout. The pre- and post-change differences (in error frequency and step times) were compared to a no-color condition, which featured the same change without colors.

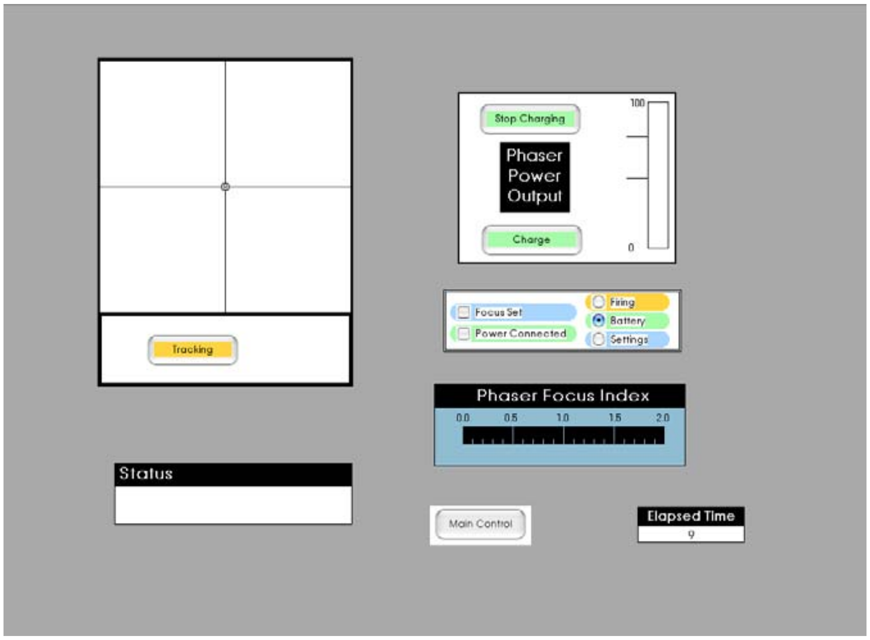


Figure 16. Color post-change version of the Phaser interface.

2.2.1. Method

2.2.1.1. Participants

Thirty-nine Rice University students (16 female, 23 male, average age 19.9 years) took part in this experiment for partial course credit and cash prizes ($10, $15, and $25 to the top three scorers). All participants had normal or corrected vision and were naive to the tasks used in this experiment.

2.2.1.2. Design

To test the ability of color labeling to reduce the detrimental effect of layout change, the color label manipulation was applied in the Phaser task. The two independent factors were condition (color labeling or no color) and change (pre and post). The primary dependent measures were again error frequency (the number of errors committed divided by the number of executions of that step) and step time (the amount of time in milliseconds required to complete a single step in the task).

As a replication of Experiment 1, the layout change condition was applied in the Phaser task, as aforementioned, and the label removal condition was applied in the Transporter task. This was done to see if the manipulations would generate similar effects as found in Experiment 1 when applied in the alternate tasks. As in Experiment 1, the primary dependent measures were error frequency and step time.

Finally, to test the effect of control ordering in the layout change condition within the Transporter task, a global rule layout change with a top-down or down-down-down (DDD) ordering for all three sets of controls was evaluated. Again, the primary measures of error frequency and step time were compared across change (pre and post) for both conditions. Participants were assigned to two different groups setup to address these three primary research questions, as shown in Table 1.

Group	**Phaser**	**Transporter**
1	Layout Change (Color)	Layout Change (DDD)
2	Layout Change	Label Removal

Table 1. Group assignment in Experiment 2.

2.2.1.3. Materials

Materials used in Experiment 2 were the same as those used in Experiment 1, with the experiment program and manuals (Appendix A) updated for the new conditions.

2.2.1.4. Procedure

The procedure for Experiment 2 was also the same as in Experiment 1.

2.2.2. Results

Data from thirty-six of the original thirty-nine participants were retained for analysis. Two participants were dropped due to software failures during the experiment, and one was removed as an outlier, again using the criterion of possessing three or more extreme (3+ inter-quartile range) data points on pre- and post- change error frequencies or reaction times at one of the steps in the target task (Phaser or Transporter). This left seventeen participants in Group 1 and nineteen in Group 2 (Table 1). Finally, five extreme error frequencies (>80%) were replaced with the grand mean for that step. Median trial data was used for each participant to mediate the effect of outliers.

To examine the effect of the color manipulation, data from seventeen participants in the color label condition and nineteen participants in the control (no color) layout change condition of the Phaser were included in the analysis. Only steps 1-8 changed in location in the Phaser task, so only data for these steps were included in the error frequency analysis. For the step times, step 3 was also removed, as in Experiment 1, since it was dependent on the system.

In contrast to our predictions, Figure 17 shows that post-change error frequency was actually higher in the color condition for almost every step. Error frequency for step 4 (second "Power Connected") was oddly higher in the pre-change condition compared to the control, while on step 6 ("Slider") the reverse is true. In the post-experiment questionnaires, 11 of the 15 participants in the color condition claimed that they had relied on the colors. Some comments from participants regarding the colors:

1. "I followed color before position."
2. "The different colors did help keep track of the order of what to do."
3. "I relied mostly on the colors. [Change] made it tough, though."
4. "The colors helped me in distinguishing between the firing, setting, and power buttons."

Note that some participants claimed to use the color to help them remember the subgoal groups (e.g., comment 4).

For the control layout change condition, post-change performance was mixed in terms of error frequency. Analysis of variance showed the main effect of layout change in to be non-significant, $F(1, 18) = 1.19$, $p = .29$, while the change by step interaction was reliable, $F(4.4, 79.6) = 3.43$, $p = .01$. In the color condition, performance was worse across nearly all the steps in the post-change state, with a reliable main effect of layout change, $F(1, 17) = 11.22$, $p < .01$, but a non-significant change by step interaction, $F(3.3, 55.5) = 0.95$, $p = .43$. Finally, the main effect of the between-participants variable of condition (color) was not significant, $F(1, 34) = 1.15$, $p = .29$.

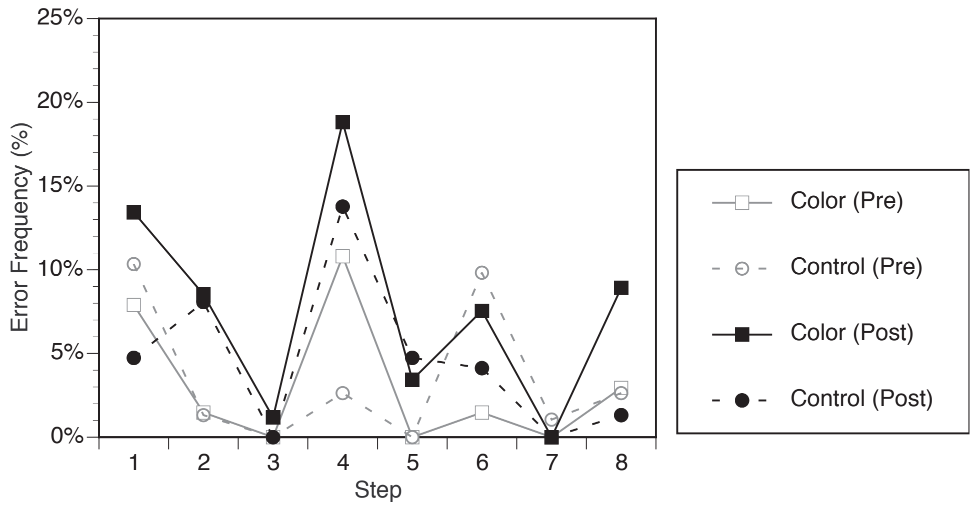


Figure 17. Error frequency by step for color and control groups (pre and post) in the Phaser task.

Step times seemed hardly to differ between the two conditions, but were obviously affected by change, as shown in Figure 18. For the control layout change condition (no color), analysis of variance for step times showed a very reliable main effect of layout change, $F(1, 17) = 29.84$, $p < .01$, and change by step interaction, $F(3.0, 51.5) = 9.58$, $p < .01$. The main effect of layout change was also reliable for participants in the color condition, $F(1, 18) = 39.20$, $p < .01$, as was the change by step interaction, $F(2.9, 52.4) = 11.65$, $p < .01$. Nonetheless, the main effect of condition (color), $F(1, 34) = 1.09$, $p = .30$, was again non-significant. Figure 18 would suggest that the main effect of change was primarily driven by the first step, but even with that data removed from the analysis, it was reliable across both conditions, $F(1, 34) = 20.89$, $p < .01$.

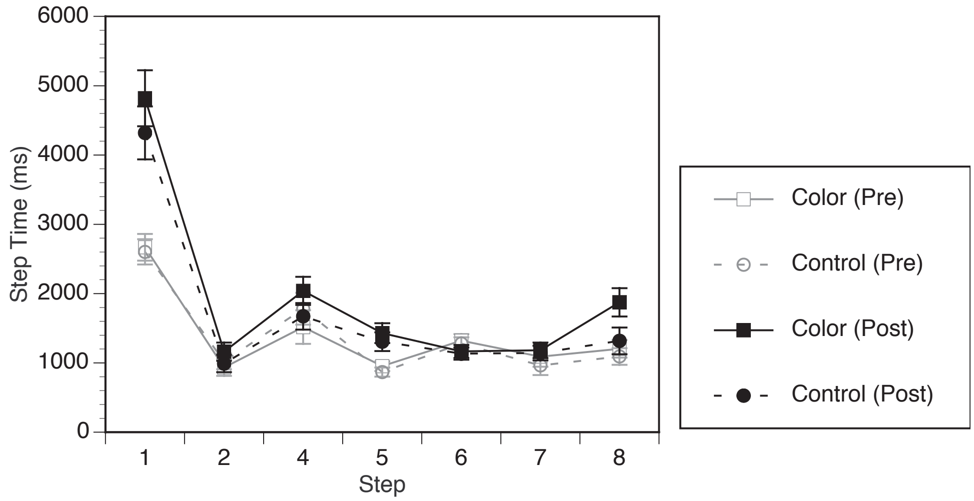


Figure 18. Step times by step for color and control groups (pre and post) in the Phaser task. Bars indicate standard error of the mean.

2.2.2.1. Experiment 1 replication

As a replication of Experiment 1, the layout change condition was applied in the Phaser task, as reported already, while the label removal condition was applied in the Transporter task. This was done to ascertain if the effects found in Experiment 1 for these manipulations were dependent on the task type, since the two tasks are not perfectly isomorphic. Only steps 1-10 were included in this analysis of Transporter error frequency, as in Experiment 1. As seen in Figure 19, label removal led to slightly increased error frequencies at certain steps. However, analysis of variance showed both the main effect of label removal, $F(1, 18) = 1.34$, $p = .26$, and the change by step interaction, $F(3.7, 66.9) = 0.40$, $p = .80$, to be non-significant.

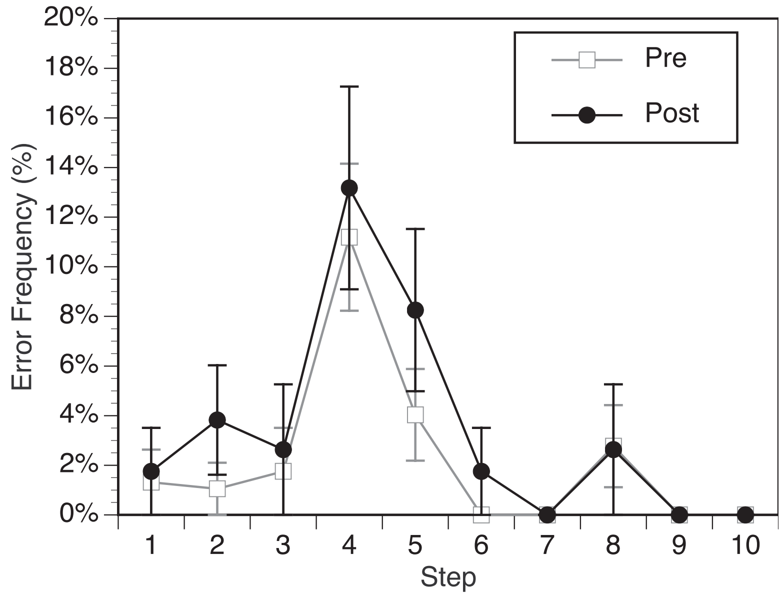


Figure 19. Error frequency by step for the label removal in the Transporter task. Bars indicate standard error of the mean.

For the step time analysis, again steps 3 and 6 were removed from the analysis, since they were dependent on the system in the Transporter condition. Analysis of variance for step times (Figure 20) for the label removal in the Transporter task showed an non-significant effect of change, $F(1, 18) = 0.39$, $p = .54$. The change by step interaction was reliable, however, $F(1.8, 32.4) = 3.78$, $p = .04$.

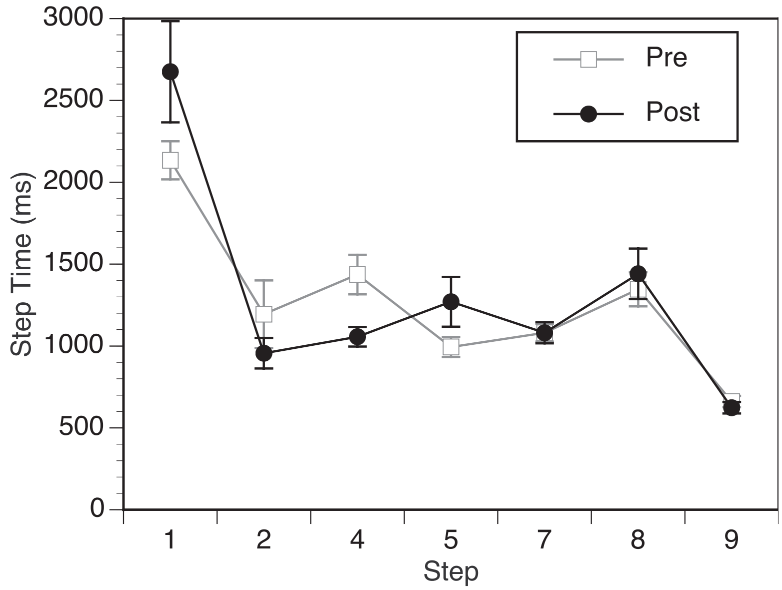


Figure 20. Step time by step for the label removal in the Transporter task. Bars indicate standard error of the mean.

To compare across the tasks (Phaser and Transporter), difference scores (post - pre) were again computed as in Experiment 1. Figure 21 shows layout change had mixed effects on error frequency, whereas with label removal error frequency either increased slightly or hardly changed. Analysis of variance for the error frequency difference scores across the layout change and label removal conditions revealed a non-significant effect of condition, $F(1, 18) = 0.04$, $p = .85$. Neither was the condition by step interaction significant, $F(5.1, 91.5) = 1.51$, $p = .19$.

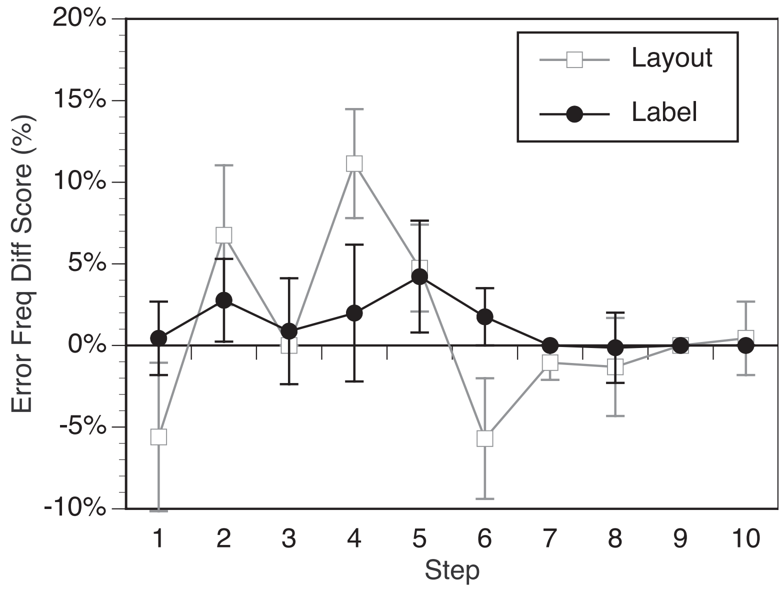


Figure 21. Error frequency difference score (post - pre) by step for the layout change (Phaser) and label removal (Transporter) conditions. Bars indicate standard error of the mean.

Label removal, however, seemed to have a much larger detrimental effect on step time at step 1 than did layout change for step time (Figure 22). Analysis of variance showed the main effect of condition, $F(1, 18) = 21.35$, $p < .01$, and the condition by step interaction to be reliable, $F(3.5, 63.1) = 3.02$, $p = .03$.

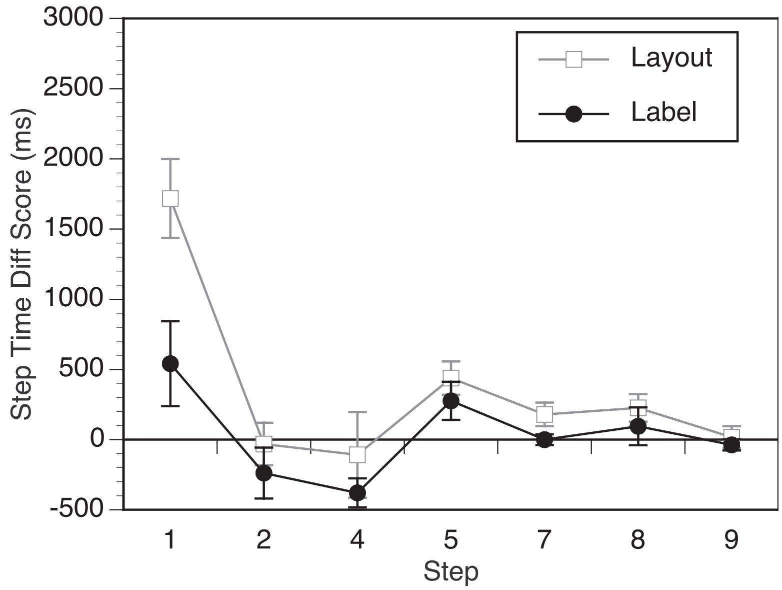


Figure 22. Step time difference score (post - pre) by step for the layout change (Phaser) and label removal (Transporter) conditions. Bars indicate standard error of the mean.

2.2.2.2. DDD control ordering in layout change

To examine the effect of top-to-bottom or down-down-down (DDD) control ordering in the layout change condition of the Transporter task, data from fifteen participants were included in the analysis. Figure 23 shows the pre and post change error frequencies for the DDD layout change condition, which for nearly all of the steps decreased post-change. Only step 3 ("Lock Signal") was markedly worse after the change. In contrast to the layout change in Experiment 1, neither the main effect of change, $F(1, 17) = 1.82$, $p = .20$, nor the change by step interaction, $F(2.4, 40.6) = 1.11$, $p = .35$, was found to be reliable here.

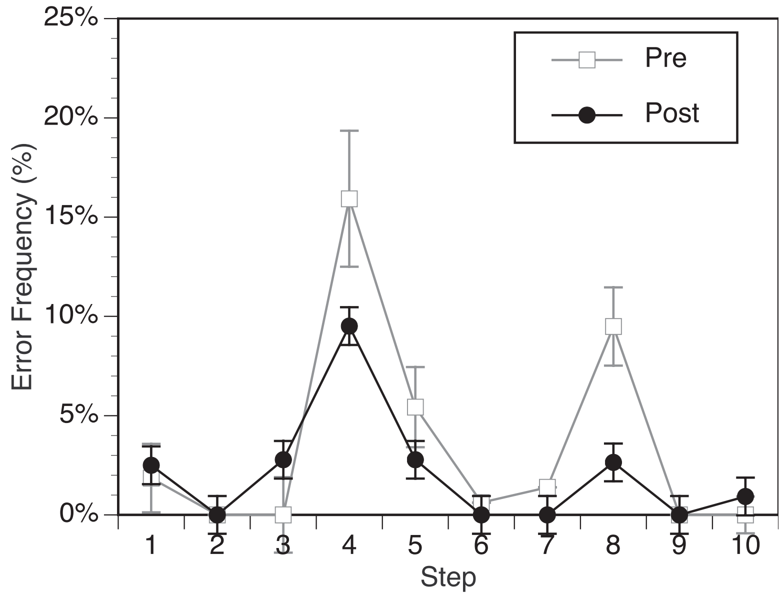


Figure 23. Error frequency by step for the DDD layout change in the Transporter task. Bars indicate standard error of the mean.

Step time (Figure 24) increased noticeably at step 1 but remained nearly constant for most of the other steps in the DDD condition. Analysis of variance for step times also revealed an unreliable main effect of change here, $F(1, 17) = 2.60$, $p = .13$. The change by step interaction also missed statistical significance, $F(1.7, 28.6) = 2.72$, $p = .09$.

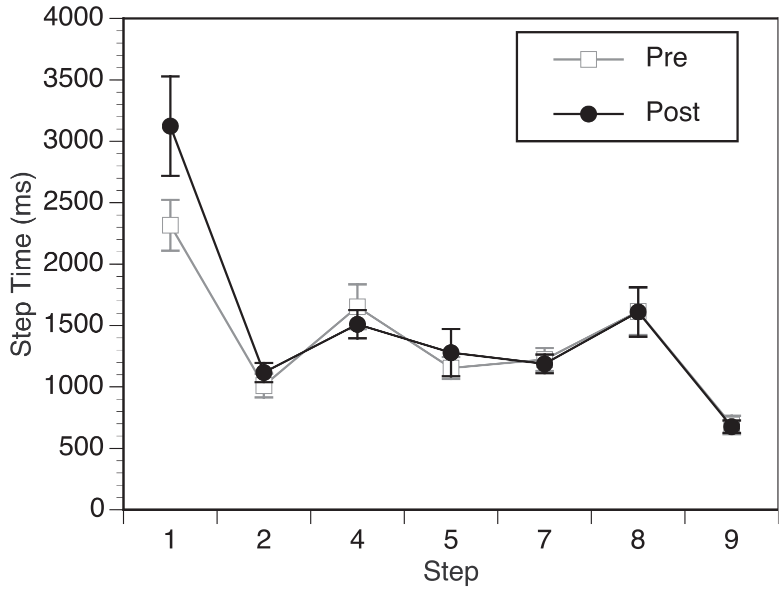


Figure 24. Step time by step for the DDD layout change in the Transporter task. Bars indicate standard error of the mean.

2.2.3. Discussion of Experiment 2

The results of Experiment 1 made it apparent going into Experiment 2 that, given some level of skill, changing the layout has a much more negative impact on performance than does simply removing the labels. Thus, in Experiment 2 two manipulations were introduced to explore methods to remediate the effect of a layout change to an interface, a certain occurrence with many real-world computer applications. First, the application of color, a primitive component of vision, was predicted to facilitate grouping of controls on the interface according to their subgoal grouping. Hence, it was thought that adaptation to the post-change locations would occur much more efficiently with the color labels than without. Ehret (2002) showed that color labels required the least effort by users to read, so it was expected that this information would be rapidly utilized.

Looking at the results, however, there seems to have been no immediate effect of color following the layout change relative to the control condition. There were even slight differences between the two conditions in terms of error frequencies in the pre-change states. As noted, error frequency at step 4 ("Power Connected") was worse with the color, while slightly better at step 6 ("Slider") in the pre-change states. Clearly, participants used the color labels somehow, since questionnaire feedback indicated that some believed the color information helped them initially learn and adapt to the post-change interface. Nevertheless, after the layout change occurred, participants in the color condition had higher error frequencies across nearly all of the steps.

One possible explanation is that having the extra dimension of color to help participants remember the controls at training may have detracted from their development of memory for the higher cost text labels (Ehret, 2002). Hence, when it came time to do the experiment under time and performance pressures and participants needed to adapt to the new layout relying on memory for the text labels, those in the color condition may have had worse memory compared to the others in the control condition. In terms of speed, there was hardly any effect of color across the steps for step times. Despite the apparent the lack of benefit in performance provided by color information, however, designers in the real world often utilize color to enhance interface design.

In Experiment 2, the layout change and label removal conditions tested in Experiment 1 were replicated in the alternate tasks. In contrast to Experiment 1, the control (no color) layout change in the Phaser task had mixed effects in terms of error frequency. This suggests that certain types of layout changes are more detrimental than others and some are even beneficial, as reflected in the different scores. Specific properties of the layout change, such as the distance between or grouping of related controls, may moderate the amount of disruption or facilitation caused. For instance, error frequency at step 4 (second "Power Connected") grew much worse in the post-change state, which may be explained by the fact that the "Stop Charging" button (step 3) was moved to the top of the screen and "Power Connected" was moved away, to the bottom of the arrangement of controls. In contrast, error frequency at step 6 ("Phaser Focus Index") decreased, perhaps as a result of moving "Settings" (step 5) closer, to a location directly above it. For step times, the layout change manipulation applied in the Phaser task had a very reliable effect, which may similarly be explained by the fact that the Phaser task is slightly more difficult than the Transporter. In contrast, the label removal condition applied in the Transporter task had no significant effect on error frequency, but a reliable change by step interaction for step times was found, suggesting that labels may be explicitly utilized for some controls more than others.

Finally, the data generated by the DDD ordering manipulation for the layout change led to some interesting findings to further support the idea that the specific type of layout change matters. First, changing to the DDD layout brought a reduction in error frequency at nearly every step in the task save one. In contrast to the reliable negative effect of the random Transporter interface layout change in Experiment 1, the DDD layout change did not generate reliable differences in error frequency. This follows the aforementioned findings from our laboratory demonstrating the effect of control ordering on performance in terms of errors. Instead of having to acquire an entirely new set of knowledge for the post-change locations of the controls, participants in the DDD condition could rely on a single preexisting rule of top-to-bottom ordering.

In summary, these results further imply that layout (or location) information is particularly important for users in routine procedural tasks. Labels, with the possible exception of the first step, seem to be increasingly ignored with the development of skill. As indicated earlier, this decreased dependence on labels comes after only 11 trials with a given task and despite having been trained with manuals emphasizing the semantic hierarchical goal structure of the task. What this suggests in terms of real-world application is that unless changes to an interface are made apparent, small revisions to the labeling may be overlooked if the overall layout remains otherwise constant. For instance, with the case of the ASCOMETAL factory worker, the buttons on the single anomalous machine were reversed in their locations. Although the labeling or the button colors may have indicated this difference, fatigue or lack of attention made it possible for the worker's knowledge of the layout to override the incongruous contextual information. Hence, if a layout change is to be made to an interface, it must be carefully managed since it can lead to either better or worse performance, as demonstrated by the layout change in the Phaser and the Transporter tasks in Experiments 1 and 2. These findings were further investigated in the subsequent field study.

3. Field Study

Human-computer interaction practitioners in industry are increasingly criticizing their fellows in academia for failing to directly address issues in the field (e.g., Gray & Salzman, 1998). This is a difficult criticism for those in academia to dodge, however, since conducting field research often requires sacrificing experimental control. For example, in the medical field it is quite costly to procure significant participation from physicians who are the primary users of various medical systems. Nurses, however, may be more readily accessible as potential participants in research (e.g., Chung et al., 2004) and were relied upon in this study.

The topic of transfer and interface design from Experiments 1 and 2 directly translates to a specific situation in this field study. A family medicine clinic in Houston, TX has been using the GE Centricity electronic medical records (EMR) system for several years. The larger institution of which this clinic is a part has been readying for a larger rollout of the Centricity system in other clinics. However, they have encountered issues with the visual layout of the forms used to collect data from patients in the family clinic, and thus desired a method to systematically analyze the data entry forms for usability. Fortunately, the GE Centricity system allows customization of the forms using a native programming language. But how can changes to the visual design be carried out without disrupting existing user performance or increasing new opportunities for error as a result of unintentional transfer of skill? One method of action based on the results of Experiments 1 and 2 was investigated.

To constrain the scope, the study focused on the vital signs form (Figure 25), which was introduced for use at the local family medicine clinic in early 2005. It is one of the few forms used by the nursing staff during a patient visit to collect and record patient information (e.g., blood pressure, height, weight, temperature) for subsequent use by a physician. There have been anecdotal instances reported in which the form was filled out by a nurse and reviewed by a physician, but abnormal vital signs information was overlooked until after the patient had left. On other occasions, nurses have incorrectly filled similarly labeled fields, due to local user expectations. For example, taking blood pressure in the supine position is less common than taking it in the sitting position, yet the field for "BP supine" is located above the field for "BP sitting."

As research in our lab has shown (e.g., Byrne et al., 2004), it is likely that if errors in human-computer interaction are systematic, there is an opportunity to reduce or eliminate their frequency through a redesign of the interface. Other issues that were considered during this field study include intra-nurse differences in the way system is used. That is, it was anticipated that some nurses might enter information into the form intermittently throughout their interaction with the patient, whereas others would gather the information first and enter it all together at the end. When redesigning the form, it was critical to assess the needs of all potential users. Hence, task analyses (Kirwan & Ainsworth, 1992) were conducted for each nurse based on observations of their workflow. The main purpose of this was to determine the implicit preexisting knowledge of the nurses, such that it might be leveraged in the redesign of the form. Changing the layout of the forms to closely follow the nurses' workflow was expected to make the forms much easier to use, just as the top-to-bottom layout change in Experiment 2 improved performance.

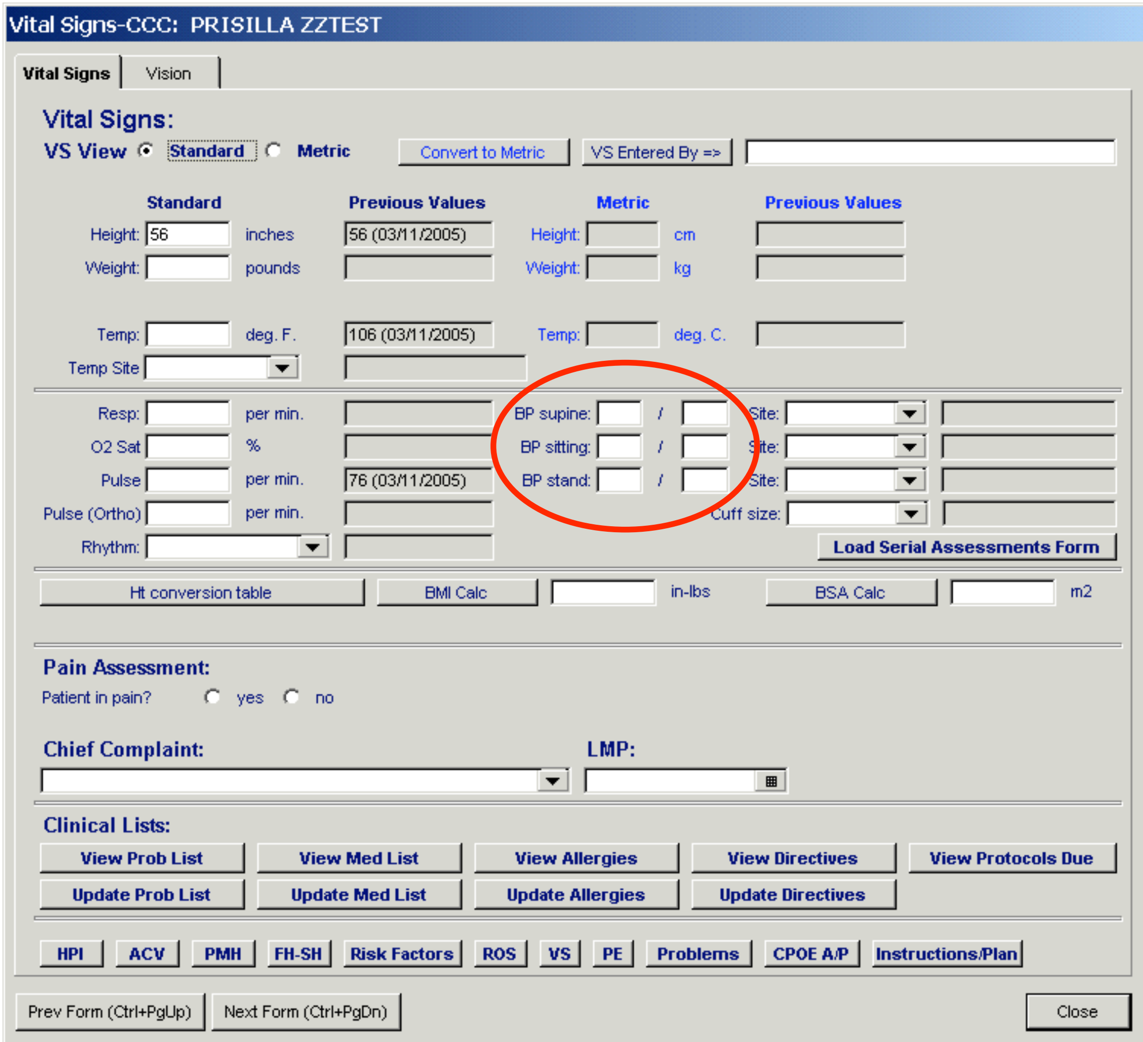


Figure 25. A partially filled vital signs form. Information on this form must be correct, visible, and complete, since it does not appear on any other form. The blood pressure fields are circled.

3.1. Method

3.1.1. Participants

Six nurses recruited by flyers posted at a local family medicine clinic participated in two sessions for $50 total and a free meal at the end of their workday. All were females aged 20 to 30 years except for one who was between 40 and 50 years. The average number of years in practice for the nurses was 5.9 years, with 2.3 years at their current clinic and 3 years of total EMR use. All were familiar with the GE Centricity system and the existing CCC forms through daily use in the workplace and rated themselves as being very comfortable with EMRs. Two confederates, both graduate students from Rice, working from a provided script also participated as "patients," and were compensated with a meal at the end of the session.

3.1.2. Materials

The Clinical Content Consultants, LLC (CCC) forms for the Centricity system developed in the Centricity environment were presented to participants on the clinic PCs. For the second session, a redesigned form was presented to nurses in a test environment on the same system. TechSmith SnagIt 3.0 screen capture software was also run on the PCs during both experiment sessions to capture on-screen activity. The System Usability Scale (SUS; Brooke, 1996), a ten item standardized questionnaire, was presented on paper at the end of the first and second sessions to assess participant opinions of system usability. The version of the SUS used in the present study was in fact slightly modified from the original version (Bangor, Kortum, & Miller, 2006; Appendix B). Short demographic questionnaires (Appendix B) for the nurses were presented on paper, and task analyses were generated using Inspiration software, based on observational notes taken in a notebook by the researcher and a walkthrough with each nurse completed at the end of the first session.

3.1.3. Procedure

All study sessions took place at the family medicine clinic at which the nurses worked. At the first session, the six nurse participants were asked to go through a typical patient visit scenario with the researcher acting as the patient. Once the nurses had read and signed the consent form and filled out a demographic questionnaire, they were told to treat the researcher as a visiting patient. This was so that the researcher could observe the nurses' workflow, which was essential to developing the task analyses and redesigned vital signs form. The screen capture software SnagIt was installed on the clinic PCs prior to each session to record nurses' use of the EMR during the experiment. Screen capture software was used to maintain a more naturalistic setting (versus an external video camera), as the nurses were going through their visit with the patients, while still making it possible to measure general task times and observe any errors. Screen captures were recorded at a rate of two per second to avoid slowing down the system's processor, which was observed to occur when using full motion video.

After starting the SnagIt screen capture software, the researcher assumed the role of the patient and requested that the nurse proceed with the visit as usual. The patient provided the name of a fake patient already existing in the system, with which the nurse was then able to proceed using the EMR. When asked for a reason for visit, the patient claimed he was there to receive immunizations for travel abroad. The nurse then began to measure and record all of the patient's vital signs (i.e., weight, temperature, respiration, blood pressure, and pulse) into the EMR system. Nurses stated that they were finished once their normal duties were complete, and the screen capture software was turned off. The researcher then recorded observational notes regarding the nurse's workflow.

All first sessions conducted with the six nurses were done in a single examining room, using the PC installed in the room (Figure 26). Sessions took up to 45 minutes and were concluded with a debriefing, during which the nurses were asked to complete the SUS questionnaire (Bangor et al., 2006). After completing the questionnaire, the researcher reviewed his notes with a walkthrough, during which the researcher verified his observations with each individual nurse. Finally, the nurses were questioned verbally about their likes, dislikes, and viewpoints regarding the system.

Figure 26. Clinic PC station: SnagIt software being setup for a session (left), Centricity EMR loaded (center), and the patient's view of the station (right).

A redesign of the form based on the information collected at the first session was then developed in the Centricity native programming environment. The design was based on the task analyses, interviews, and reviews of the screen captures from the first session. Once the redesigned form was complete, it was installed on the clinic PCs within a test environment, in preparation for the second session. Only a single menu not often accessed by nurses was not functional, and nurses were warned of this before the session during an introductory walkthrough with the form. This walkthrough was important to familiarize nurses with the changes to the form before the experiment session began and was typical of how nurses at the clinic had been trained to use the system in the past.

The second session was conducted one week later over a period of two days, with the two confederates acting as additional patients. They interacted with the nurses according to a memorized script provided by the researcher before the session. Three patients were run this time to generate more data with the new form, give nurses an opportunity to overcome any learning curve, and allow comparison with the existing form. Again, the setup with the room and PC was the same as before, although there were three rooms used at once this time, with one patient per room.

As is typically the case in practice, the nurses were asked to move from room to room to see the three patients. The order in which this was done was counterbalanced as much as possible given the time restrictions, since scheduling required accommodating the nurses' schedules as well as the patients and clinic's. This time the researcher could not physically observe each session, but nurses were asked to review a printout of their own personalized task analysis following their visit with the researcher to verify that nothing had changed. When they had finished seeing all three patients, nurses were asked to fill out the SUS questionnaire, debriefed, and briefly interviewed.

3.2. Results

3.2.1. Task analysis

Hierarchical task analyses (Kirwan & Ainsworth, 1992) were constructed for each of the nurses after the first session in the clinic. The nurses later verified these at the second session, once they had been printed out for review. The main purpose of this task analysis was to look for any intra-nurse differences in workflow, since these differences needed to be supported with any redesign of the system. Since the central purpose of the field study was to apply findings from Experiments 1 and 2 in a real-world situation, task analyses of the nurses' workflows were used to determine preexisting knowledge to guide the ordering or layout of the objects on the new form.

Generally, there were not many intra-nurse differences observed during the first session of the experiment. The task of "rooming" a patient for the physician was simple enough that there was minimal room for variability in the workflow. Figure 27 shows a task analysis compiled for all nurses observed at the first session with variable steps shown in dotted lines. The three main sources of variability, if any, were patient differences (e.g., a talkative patient), how fast the system was running (nurses verbally reported that sometimes the network could be slow, in which case they might proceed with activities not dependant on the computer), and finally how much information the nurse preferred to keep in his or her short-term memory at once. This last variable determined whether or not the nurse would go back to the system after measuring each vital sign to enter it immediately into the system or not.

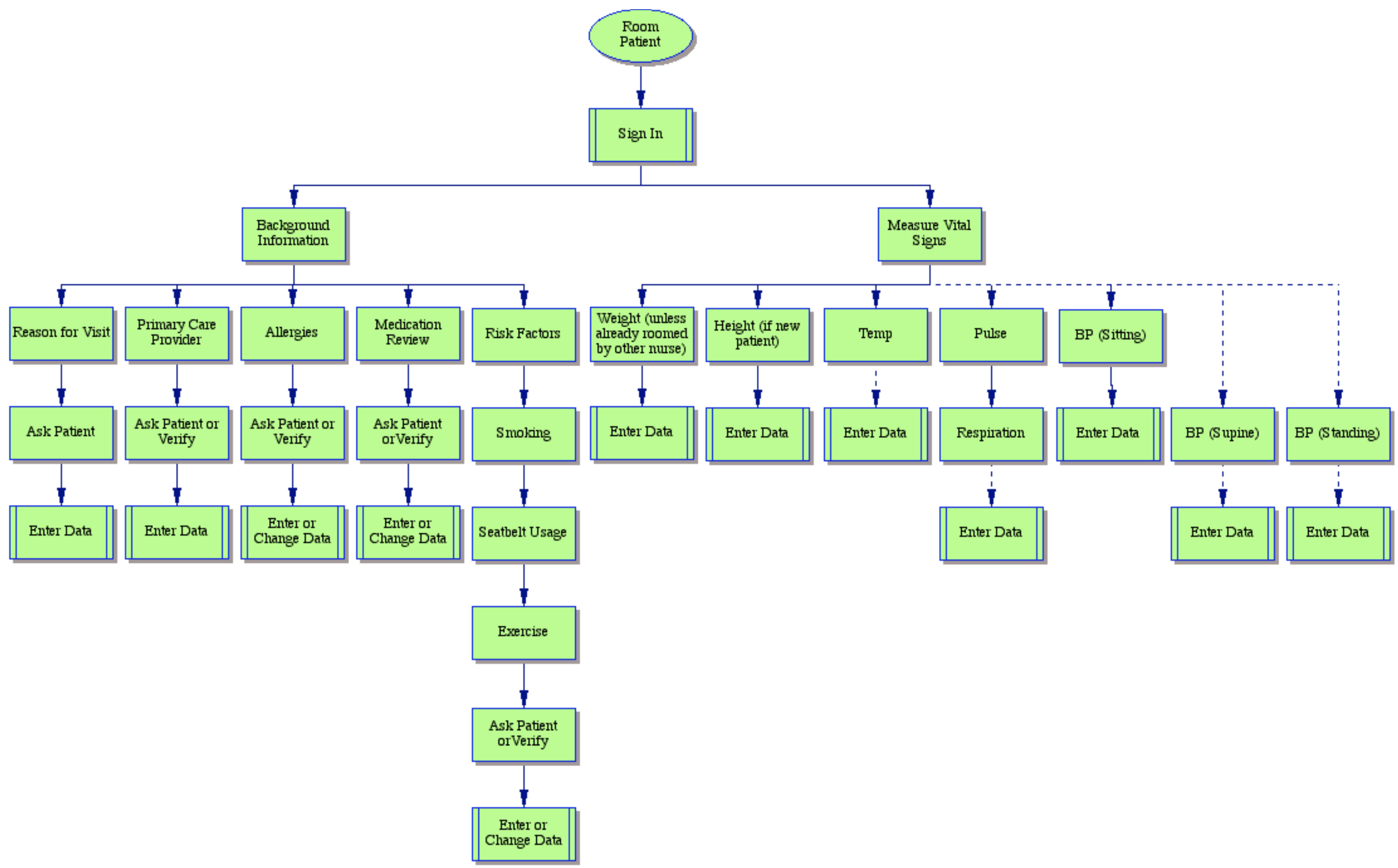


Figure 27. Compiled hierarchical task analysis for the process of "rooming" a patient. Observed variations are noted with dotted lines. For example, BP (Supine) and BP (Standing) are usually only done in special situations. Regular boxes indicate goals, while boxes with double lined borders indicate goals requiring interaction with the EMR.

3.2.2. Redesign

The new form (Figure 28) was designed with visual clutter reduced as much as possible. This usability goal was motivated by the findings from the color manipulation in Experiment 2 and was achieved by using checkboxes to hide fields that were not often used. Moreover, the new layout followed nurses' workflow, with the fields in order of use from top-to-bottom and left-to-right within the space of the existing window. This was relatively easy to do, since all nurses were observed to enter the data in the same order. As suspected, the only major difference found between nurses was that some tended to move back and forth more often with the patient and PC rather than gathering several vital signs measures from the patient, keeping them in their head, and entering the data into the PC all at once.

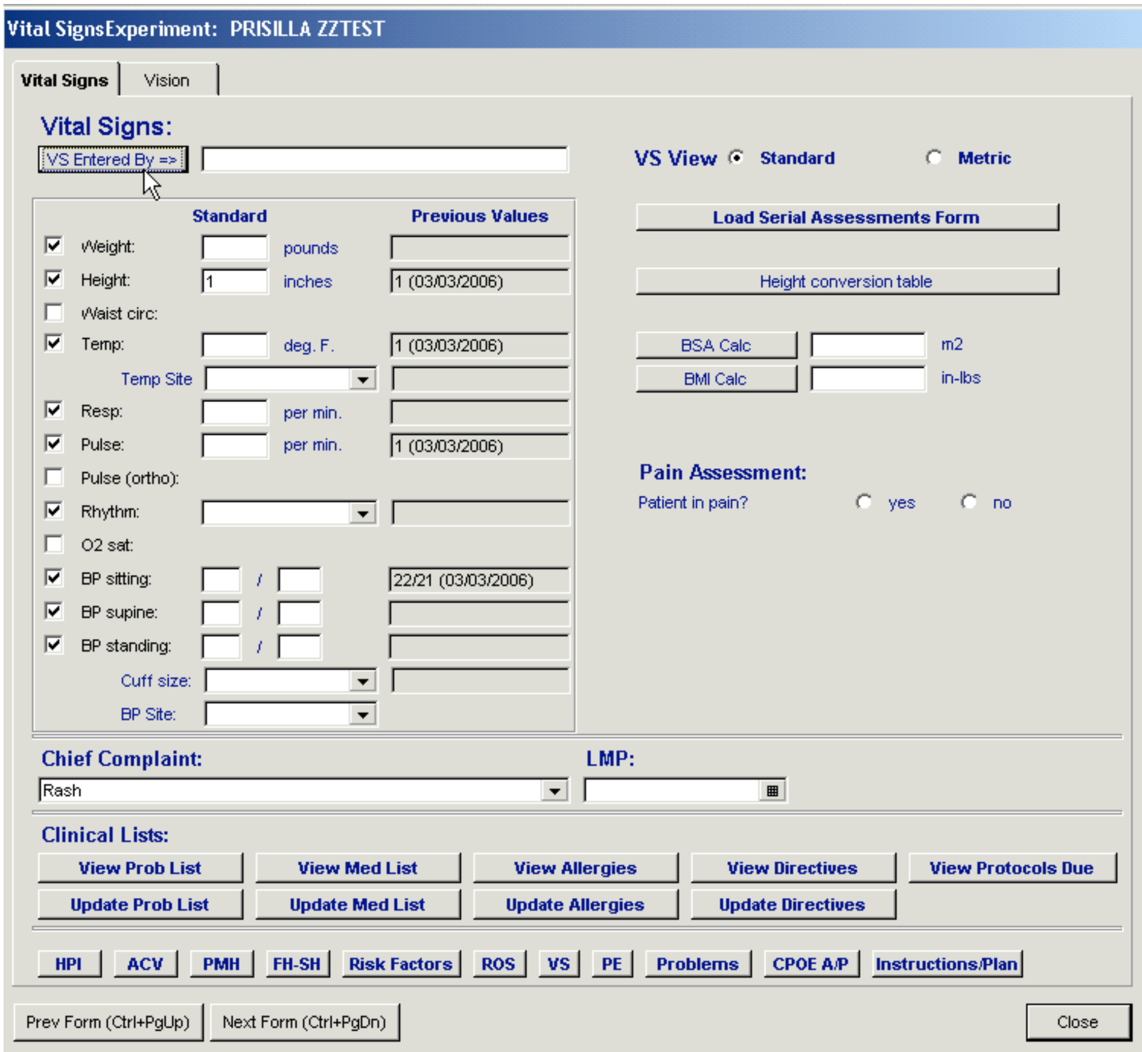


Figure 28. Redesigned vital signs form. All three BP fields are checked and open to allow for potential capture errors.

3.2.3. Screen captures

All six nurses participated in the first session on the same day and all of their data were included as Trial 1. Only one trial was conducted on the first day due to time constraints with the nurses. The second session was run over two days with each nurse participating in three trials.

With the redesigned form at the second session, all three BP fields were left open at the beginning of each trial to see if nurses might commit an error of recording the blood pressure in the middle field, which would be the wrong field with the new form. This was anticipated as a potential error, since seeing the familiar arrangement of the three BP fields could lead nurses to enter data in the middle field (as in the old form) without reading the labels carefully. For the second session, one of the nurses (F) was unable to participate at a scheduled time and was instead run by a colleague of the researcher, a physician in the clinic, who herself acted as the patient for a single trial (versus three patients and trials with the other nurses). In this session, only the top (sitting) BP field was left open. A second nurse mistakenly accessed the old form during her first two trials at the second session, so those data were removed from this analysis.

Form completion times were measured using the number of screenshots captured (one every two seconds), from when the vital signs form was first opened to when the vital signs form was finally closed. Although this did not control for the length of interaction with the patient, it was difficult to tell exactly when the nurse was in front of the computer. Moreover, the primary goal here was to simply get a general trend in completion times for each nurse, since there were some factors in the field that were impossible to control such as prior experience and others noted previously.

Two cases in which nurses measured and recorded three BP values for the single patient whose chief complaint was being dizzy were excluded (nurse C, trial 2 and nurse A, trial 4). This was done to allow fair comparison across the data, since every other nurse measured and recorded only one BP value. Overall, it seems that form completion times were the same or less than what they were at trial 1 with the existing form (Figure

29). Completion time for nurse D, particularly, exhibited a drastic decrease after the new form was implemented at the second session, starting with trial 2.

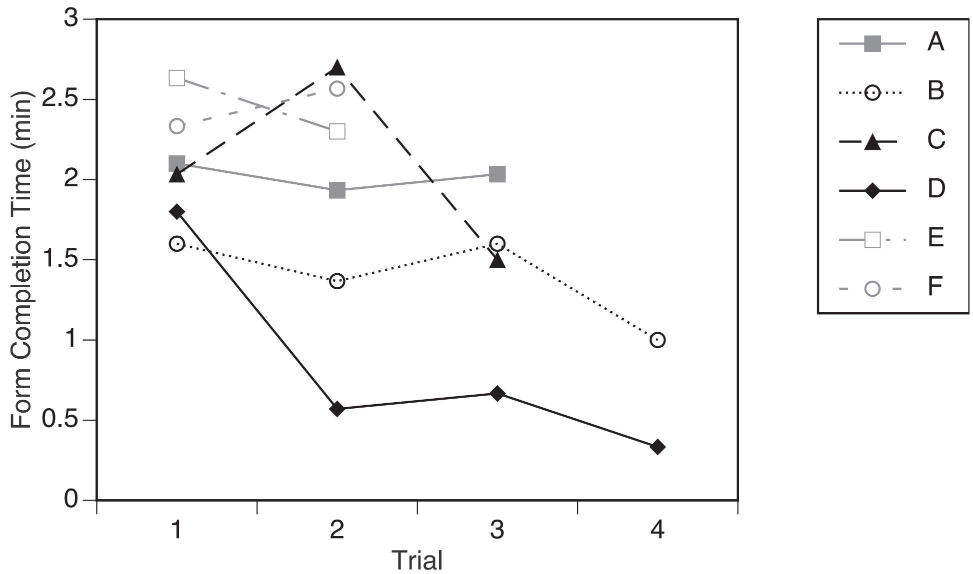


Figure 29. Form completion time (min) by trial and nurse (A-F). The original form was used in trial 1. Two longer trial instances, where a nurse measured a patient's blood pressure in all three positions instead of just one, are not presented in this graph.

None of the expected low-frequency errors were discovered in review of the screen captures from the second session. However, one nurse was found to have made an error in data entry with the original form entering "175" for weight instead of "135," and another nurse incorrectly entered "987.3" degrees in the temperature field. Also, with the new form it was found that three of the nurses always clicked the checkboxes to close the unused BP fields before filling in the BP sitting field.

3.2.4. SUS questionnaire

Figure 30 shows the SUS questionnaire scores for the nurses from both sessions. Scores range from 0 to 100, with higher scores indicating better perceived usability. Although the mean score for the second session (85.4) was higher than the mean score for the first (73.3), the paired samples t-test for the six participants did not show a reliable effect of session (or form), $t(5) = 2.01$, $p = .10$. However, two of the nurses' scores (slightly lower for the second session) contradicted their positive comments regarding the new form from the interviews. In a follow up via email, they claimed to have rated the entire system as a whole both times rather than just the changed form. Nurses were not explicitly asked to rate only the vital signs form when the questionnaire was administered, since they were asked to go through the entire process of "rooming" a patient and the vital signs form is a key (but not the only) component of that process. It was thus expected that any usability improvement made by the new form would be captured by the general questionnaire, although in retrospect it seems it may have been better to ask them explicitly to focus on the vital signs form.

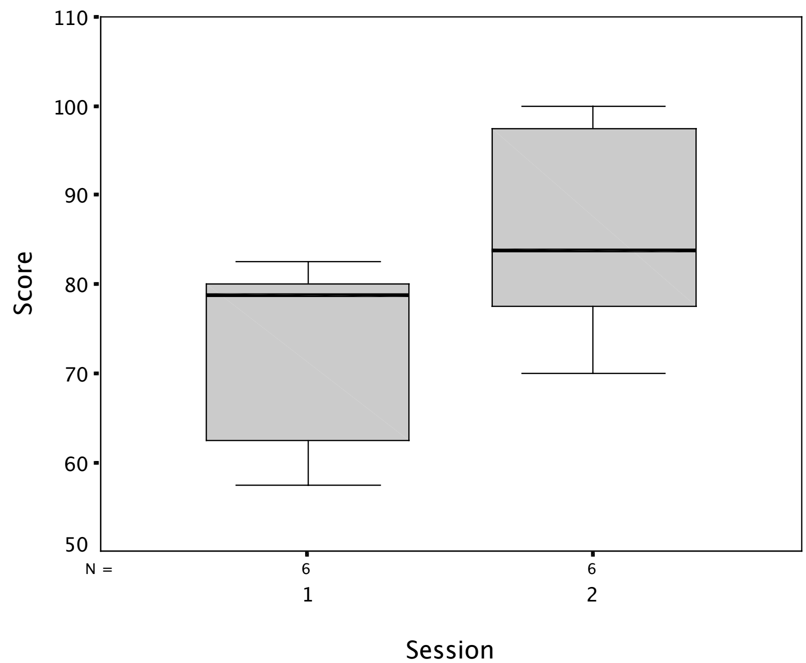


Figure 30. SUS scores for all six nurses from session 1 (existing form) and session 2 (redesigned form).

3.2.5. Interviews

The nurses claimed that the training provided by the clinic itself for the EMR was not comprehensive. In fact, several mentioned that when minor changes were made to the forms, they would simply be notified via email or the change might be briefly mentioned at a weekly meeting. Nevertheless, the nurses were so experienced with the existing form at the time of the study that they could report no major problems with it. Almost all of them, however, did recall that the initial learning curve had been steep. One of the newer nurses claimed that her first experiences with the EMR were "scary," because she was never sure to whom the data would get sent when she entered it into the system. She thus noted that in the beginning she would double-check what she had entered into the system with other nurses.

The nurses mentioned that they wanted to spend as little time as possible navigating the system and entering data. One nurse noted that in the past the system required entered data to be verified each time, by pressing a highlighted button to record. She felt that the existence of so many dependencies within the system was what caused the steep learning curve. In contrast, several of the nurses immediately recognized that the revised form matched their workflow much better. One of the nurses complained, at the end of the first session, that a previous version of the form had better matched their workflow, but for some unknown reason it had been suddenly replaced with the existing form. More likely, in light of two other nurses' comments, the form evolved over the year, as administrators added new fields into it without consulting the nurses who were actually using the forms. Hence, the resulting form that was in use at the time of the study no longer matched the nurses' workflow, as the old form may have once done.

In contrast, all of the nurses said that the new form was far "simpler," with the unnecessary fields being available but hidden with the checkboxes. Too many sections, they exclaimed, often led to errors with information being entered in the wrong fields. They also liked the fact that the frequently used fields were all placed along one side from top-to-bottom rather than scattered across the form as before. According to the nurses, workflow was determined not by the form itself, but rather by each individual nurse and each individual situation with the patient. For example, if the system was being slow, the nurses would get the patient's weight or take other vital signs measures until the system finished loading. Thus, ordering the fields to follow nurses' workflow and allow easier place keeping was seen a welcome improvement to reduce errors in data entry.

3.3. Discussion of the Field Study

The redesigned form was well received by the nurses, and the form completion time data suggest that it would have perhaps led to improved performance with extended use. By spending no more than five hours in the clinic with six different nurses, it was possible to model the nurses' workflow and generate a redesigned form that was better able to support their job. Furthermore, the nurses were positive and seemed to thoroughly appreciate the fact that their input was being considered in the design of a tool they themselves had to use for everyday work. Although we were unable to get a baseline measure of error frequencies with the existing form due to delays with the institution's IT department, even in this short field study it was possible to capture two data entry errors in a single trial. In comparison, in the three trials conducted with each nurse using the new form, there were no identifiable data entry errors. This finding underlined the nurses' comments in the interviews that the unnecessary complexity or visual clutter of the existing form has sometimes caused them to incorrectly enter data into the wrong field.

For whatever reason, the existing forms prominently display fields that are rarely used in a typical visit, leading to unnecessary complexity and clutter. In large organizations where the entities managing the IT systems are distant from the users themselves, changes pushed by the administration (e.g., the addition of extra fields on the forms to collect other desired information) over time can potentially lead to the inefficient and error-prone design of tools for the users. Such problems can be avoided through user research, and in the long run such actions may even reduce training requirements. As reflected in the form completion times, there was practically no learning curve to overcome with the redesigned form.

4. General Discussion

The findings from these experiments suggest that, contrary to many theories emphasizing the importance of labels (e.g., Polson & Lewis, 1990) and the hierarchical goals structure of tasks (e.g., John & Kieras, 1996) in interactive behavior, users of computer systems come to rely heavily on the locations of objects on the interface with practice. That is, as skill grows and performance becomes increasingly automatic (Anderson, 1982), reliance on the labels or semantic information seems to decrease. Even with the sudden removal of labels, error frequencies generally continued to decrease. On the contrary, the layout change manipulation by and large hurt participant performance after the change, as revealed by the difference scores for error frequency and the types of errors made.

In both experiments, however, the layout change had minimal effect on step times. This may be related to the findings for form completion times from the field study, which showed almost no negative effect in terms of changing the layout and, in some cases, an immediate improvement in performance after the change. Perhaps, regardless of any uncertainty caused by layout changes to an interface, users do not slow down their interaction, if they possess solid task representations and some incentive to perform the work quickly. Since the objects and labels were not changed on the form in the field study, nurses had only to adjust to the new locations (and were given a walkthrough prior to the second session). Moreover, because the redesigned form followed their well-practiced workflow and a single global rule (top-to-bottom and left-to-right), adjusting to the new layout was relatively easy for them. This was similar to what we found in the DDD layout change manipulation from Experiment 2.

There does, however, seem to be at least one special case where labels are particularly important, at least with the level of skill participants' attained in these experiments. That is the first step in a task, where the labels seemed to matter for initial orientation on the interface, as performance slowed markedly when they were removed. Participants also claimed to rely on the text labels when recovering from errors made in the post-change state, and even the color labels were said by many to have been helpful in initially learning the task. Yet, it seems that although most participants in the layout change conditions reported that the labels were the most reliable post-change source of information to aid their recovery, given the lack of detriments in performance generated in the label removal condition, clearly participants were not always reliant on them.

Layout change also seems to be multidimensional, with factors such as the distance between controls moderating the effect it has on task performance. As demonstrated by the layout change in the Phaser task from Experiment 2, changing the layout did not affect performance negatively at every single step. In fact, similar to the DDD condition, performance at certain steps where controls were moved closer together improved after the layout change. This finding further supports the claim that layout information is more important than previously thought, and implies that changing the layout of an interface can not only hurt but also benefit task performance.

The work of Fu and Gray (2004) and other researchers (Gray & Boehm-Davis, 2000) that has demonstrated the tendency of users to economize in interactive behavior may explain why participants came to rely on less reliable location memory (since it is primarily internal). Simply, it affords less effortful interaction. This makes sense if one considers the difference in cognitive operations required between the two different processes. Interactive behavior that relies on location memory would require only a retrieval of knowledge for the location of the appropriate control at each step. On the other hand, interactive behavior that always relies on label information would require two steps: one to retrieve semantic knowledge for the labels themselves and an additional visual search of the interface to find the matching label for the control. Thus, as in the work of Fu and Gray (2004) and others, participants in these experiments may have came to rely upon location memory in as few as eleven trials with a given task, because operationally it requires fewer cognitive steps. This may also explain why using a post-change layout that could be remembered with a single rule (DDD) improved performance.

What this suggests is that in many real-world situations where there are external factors driving users to work quickly in computer-based tasks, it is probable that they will turn to rely on their imperfect memory for locations of controls to reduce cognitive demands. In the field study none of the predicted errors of putting the blood pressure in the middle field were captured, perhaps due to the low number of trials. However, one can suspect that if a nurse who was experienced with the original system was in a hurry to get to the next patient, he or she might rely on existing memory for locations and incorrectly enter the blood pressure in the middle field. In the case of the ASCOMETAL factory worker, fatigue or lack of attention while working the night shift may have caused him to fall back on location memory, which was correct for any of the other machines in the factory. However, by relying on the wrong knowledge for the anomalous machine the worker failed to meet a demand imposed by the task or tool, as Byrne's (2003) definition of error states.

Experiment 2 examined methods to mediate the detrimental effects of the layout change. With the color labeling manipulation, the use of color to group related controls by subgoal was expected to help users, by supporting their memory of the hierarchical goal structure. Contrary to the predictions, the addition of color labels actually made performance slightly worse, although step time differences from the control were again nearly non-existent.

In contrast, the global rule (DDD) layout change manipulation was quite effective, because it supported participants' reliance on location memory and simplified what they needed to learn and recall. Instead of having to learn new individual locations for the controls, a very simple global rule could be relied upon: all buttons in the subgoals were arranged from top-to-bottom. Hence, even with the time and performance pressures, they could work very quickly, and accuracy improved despite changes in the locations of the controls. This may correspond with the findings from the field study, which showed hardly any negative effect on form completion times for the changed layout in the vital signs form. Since the new form matched their preexisting workflow or knowledge, there was little for them to remember in terms of the new locations other than the simple rule that they could put data into the form in the order it was collected. This appeared to shorten the learning curve, and the nurses unanimously stated it was much easier to use than the existing form.

Even low-level graphical differences cannot defeat this strong tendency to rely on memory for locations. As already discussed, the use of color labels was ineffective at reducing the deleterious effect changing control locations had on error frequency. Thus, if a safety critical system needs to be redesigned in such a way that memory for location can potentially lead to a capture error, such as with the BP fields in the field study, it is imperative that measures be taken to minimize the potential risk and severity of that error. For example, with the vital signs form in the field study, although the locations were changed, the checkboxes prevented users from entering data into the less frequently used fields below, thereby preventing any potential errors.

If a process such as that used in the field study is used to design the layout of an interface in a way that follows preexisting knowledge, user adaptation to the new interface seems to require minimal effort. Final performance may eventually even surpass that of skilled users with a poorly designed interface. Extraction of the tacit preexisting knowledge, however, requires designers to study both the target user group's task representation and device layout knowledge (Kieras & Polson, 1985). For the DDD manipulation in Experiment 2, participants' task representation was simply considered as what was given in the manuals, and all were expected to have preexisting knowledge to work top-to-bottom on the interface. However, with the vital signs form it was necessary to conduct interviews and observe nurses working with patients to ascertain task representation and screen capture data to determine the optimal device layout.

One possibility to examine in future work would be to conduct the layout change experiment and emphasize the importance of correct performance (e.g., using the point system) versus speed to see if errors still increase as much in the post-change state relative to the label removal manipulation. Although the layout change led to increased errors in Experiments 1 and 2 versus the label removal manipulation, it is possible that this was only the case because participants were motivated to work as quickly as possible. Since implicit memory for location takes longer to acquire than good label information, it would seem that somewhere along the line before the change occurred, participants in both the layout change and label removal conditions would have had better memory for labels relative to the locations. However, it seems that with further practice participants in both conditions came to rely on location memory more, since it allowed less effortful interaction.

It may also be worth replicating the results of the label removal condition with the labels completely removed instead of being crossed out. In these experiments, Xs were used to replace the label text, since the objective was to remove semantic label information, not graphical information. However, it is possible that the retained graphical information provided by the shape of the replacement Xs may have helped users remember the controls in the post-change state.

The application of these results may be most relevant to routine computer-based tasks. This can be any computer-based task where the user must repeat specific interactive behaviors, such as is the case with computer systems used by assembly line workers or even many everyday applications on the personal computer. Given these results, particularly in situations where there are external time and performance pressures, it seems prudent to consider layout seriously when designing a computer interface for any safety-critical application. In situations where change is necessary, this tendency of users to rely on layout information can be leveraged to design more usable interfaces, by first examining users' task representation, and then arranging the device layout to follow it. This was done with the vital signs form, by studying the nurses' workflow, and then organizing the objects on the interface appropriately, from top-to-bottom and left-to-right. Traditionally, goal structure (John & Kieras, 1996), labels (e.g., Polson & Lewis, 1990), and other aspects of visual design (e.g., colors and font size) have been given most of the attention in usability practice and the study of human-computer interaction. Needless to say, these aspects of interface design are important, but in routine tasks it seems that our natural tendency to take the path of least effort in interactive behavior will undoubtedly lead to heavy reliance on more efficient memory for control layout.

5. References

Anderson, J. R. (1982). Acquisition of cognitive skill. *Psychological Review, 89(4)*, 369-406.

Anderson, J. R. (1990). *The adaptive character of thought*. Hillsdale, NJ: Erlbaum Associates.

Anderson, J. R. (1995). *Learning and memory*. New York: Wiley.

Annett, J., & Duncan, K. D. (1967). Task analysis and training design. *Journal of Occupational Psychology, 41*, 211-221.

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.

Baig, E. C. (2005, November 11). Why are tech gizmos so hard to figure out? *USA Today*. Retrieved April 11, 2006 from http://www.usatoday.com/money/industries/technology/2005-11-01-usability-cover_x.htm

Ball. L. J., Ormerod, T. C., & Morley, N. J. (2004). Spontaneous analogizing in engineering design: a comparative analysis of experts and novices. *Design Studies, 25*(5), 495-508.

Bangor, A., Kortum, P. & Miller, J. T. (2006). *An empirical evaluation of the system usability scale (SUS)*. Manuscript in preparation.

Bartlett, F. C. (1932). *Remembering: An experimental and social study*. Cambridge, UK: Cambridge University Press.

Besnard, D. & Cacitti, L. (2005). Interface changes causing accidents. An empirical study of negative transfer. *International Journal of Human-Computer Studies, 62*, 105-125.

Beyer, H. & Holtzblatt, K. (1998). *Contextual design: defining customer-centered systems*. San Francisco: Morgan Kaufmann.

Brooke, J. (1996). SUS: a 'quick and dirty' usability scale. In P. Jordan, B. Thomas, B. A. Weerdmeester, & I., McClelland. (Eds.), *Usability evaluation in industry*. Bristol, PA: Taylor & Francis.

Byrne, M. D. (2003). A mechanism-based framework for predicting routine procedural errors. In R. Alterman & D. Kirsh (Eds.), *Proceedings of the Twenty-Fifth Annual Conference of the Cognitive Science Society*. Austin, TX: Cognitive Science Society.

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

Byrne, M. D., Chung, P. H., & Fick, C. S. (2004, May). *Mitigating errors in the execution of routine procedures*. ONR Workshop on Attention, Perception, and Modeling for Complex Displays, Newport, RI.

Byrne, M. D., & Davis, E. M. (in press). Task structure and postcompletion error in the execution of a routine procedure. *Human Factors*.

Byrne, M. D., Maurier, D., Fick, C. S., & Chung, P. H. (2004). Routine procedural isomorphs and cognitive control structures. In C. D. Schunn, M. C. Lovett, C. Lebiere & P. Munro (Eds.), *Proceedings of the Sixth International Conference on Cognitive Modeling* (pp. 52-57). Mahwah, NJ: Erlbaum.

Carroll, J. M. (1995). *Scenario-Based Design: Envisioning Work and Technology in System Development*. New York: Wiley.

Chapanis, A. (1996). *Human factors in system engineering*. New York: Wiley.

Chung, P. H., & Byrne, M. D. (in press). Cue effectiveness in mitigating postcompletion errors in a routine procedural task. *International Journal of Human-Computer Studies*.

Chung, P. H., Zhang, J., Johnson, T. R., & Turley, J. P. (2004, September). *A comparative study of patient safety using infusion pumps*. Paper presented at the meeting of the International Medical Informatics Association (MEDINFO), San Francisco, CA.

Craik, F. M., & Lockhart, R. S. (1972). Levels of processing: A framework for memory research. *Journal of Verbal Learning and Verbal Behavior, 11*, 671-684.

Detterman, D. K. (1993). The case for the prosecution: Transfer as epiphenomenon. In D. K. Detterman & R. J. Sternberg (Eds.), *Transfer on trial: intelligence, cognition, and instruction* (pp. 1-38). Norwood, NJ: Ablex.

Ehret, B. D. (2002). Learning where to look: location learning in graphical user interfaces. In *Human Factors in Computing Systems: Proceedings of CHI 2002* (pp. 211-218). New York: ACM.

Fiske, S. T. & Taylor, S. E. (1991). *Social cognition*. New York: McGraw-Hill.

Fleetwood, M. D., & Byrne, M. D. (in press). Modeling the visual search of displays: A revised ACT-R/PM model of icon search based on eye tracking data. *Human-Computer Interaction*.

Gray, W. D. (2000). The nature and processing of errors in interactive behavior.

Cognitive Science, 24(2), 205-248.

Gray, W. D., & Boehm-Davis, D. A. (2000). Milliseconds Matter: An introduction to microstrategies and to their use in describing and predicting interactive behavior. *Journal of Experiment Psychology: Applied, 6*(4), 322-335.

Gray, W. D., & Fu, W. T. (2004). Soft constraints in interactive behavior: the case of ignoring perfect knowledge in-the-world for imperfect knowledge in-the-head. *Cognitive Science, 28,* 359-382.

Gray, W. D., Fu, W. & Salzman, M. C. (1998). Repairing damaged merchandise: A rejoinder. *Human-Computer Interaction, 13*(3), 325-335.

Hajdukiewicz, J. P., & Vicente, K. J. (2002). Designing for adaptation to novelty and change: functional information, emergent feature graphics, and higher-level control. *Human Factors, 44*(4), 592-610.

Holyoak, K. J., & Koh, K. (1987). Surface and structural similarity in analogical transfer. *Memory and Cognition, 15*, 332-340.

Homa, D., & Viera, C. (1988). Long-term memory for pictures under conditions of thematically related foils. *Memory & Cognition, 16*(5), 411-421.

Janzen, M. E., & Vicente, K. J. (1998). Attention allocation within the abstraction hierarchy. *International Journal of Human-Computer Studies, 48*, 521-545.

John, B. E., & Kieras, D. E. (1996). The GOMS family of user interface analysis techniques: Comparison and contrast. *ACM Transactions on Computer-Human Interaction, 3*(4), 320-351.

Kieras, D. E., & Polson, P. G. (1985). An approach to the formal analysis of user complexity. *International Journal of Man-Machine Studies, 22*, 365-394.

Kimball, D. R., & Holyoak, K. J. (2000). Transfer and expertise. In E. Tulving & F. I. Craik (Eds.), *The oxford handbook of memory* (pp. 109-122). New York: Oxford University Press.

Kirwan, B. & Ainsworth, L. K. (1992). *A guide to task analysis*. London: Taylor & Francis.

Koffka, K. (1935). *Principles of gestalt psychology*. New York: Harcourt, Brace, and World.

Lewicki, P., Czyzewska, M., & Hoffman, H. (1987). Unconscious acquisition of complex procedural knowledge. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 13*(4), 523-540.

Neisser, U. (1976). *Cognition and reality: principles and implications of cognitive psychology*. San Francisco: W.H. Freeman.

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

Oulasvirta, A., Kärkkäinen, L. & Laarni, J. (2005). Expectations and memory in link search. *Computers in Human Behavior, 21*(5), 773-789.

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

Raskin, J. (1994). Intuitive equals familiar. *Communications of the ACM, 37*(9), 17.

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, UK: John Wiley.

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

Rieman, J., Young, R. M., & Howes, A. (1996). A dual-space model of iteratively deepening exploratory learning. *International Journal of Human-Computer Studies, 44*(6), 743-775.

Salminen, S., & Tallberg, T. (1996). Human errors in fatal and serious occupational accidents in Finland. *Ergonomics, 39*(7), 980-988.

Singley, M. K., & Anderson, J. R. (1989). *The transfer of cognitive skill*. Cambridge, MA: Harvard University Press.

Somberg, B. J. (1987). A comparison of rule-based and positionally constant arrangements of computer menu items, In Carroll, J. M., & Tanner, P. P. (Eds.), *Proceedings of ACM CHI+GI'87 Conference on Human Factors in Computing Systems and Graphics Interface* (pp. 255-260). New York: ACM.

Sweller, J. (1980). Transfer effects in a problem solving context. *Quarterly Journal of Experimental Psychology, 32*, 233-239.

Tamborello, F. P., & Byrne, M. D. (2005). Information search: The intersection of visual and semantic space. *CHI 2005 Extended Abstracts on Human Factors in Computing Systems* (pp. 1821–1824). New York: ACM.

Ulrich, K. T., & Eppinger, S. D. (1995). *Product design and development*. New York: McGraw-Hill.

U.S. Department of Health and Human Services. (2003). *Research-based web design & usability guidelines*. Retrieved from http://www.usability.gov/guidelines/

Vicente, K. J. (1992). Memory recall in a process control system: A measure of expertise and display effectiveness. *Memory & Cognition, 20*(4), 356-373.

Woltz, D. J., Bell, B. G., Kyllonen, P. C., & Gardner, M. K. (1996). Memory for order of operations in the acquisition and transfer of sequential cognitive skills. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 22*, 438-457.

Woltz, D. J., Gardner, M. K., & Bell, B. G. (2000). Negative transfer errors in sequential cognitive skills: strong-but-wrong sequence application. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 26*(3), 602-625.

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

6. Appendix A

6.1. Main control manual (control version)

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: Tactical, 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.

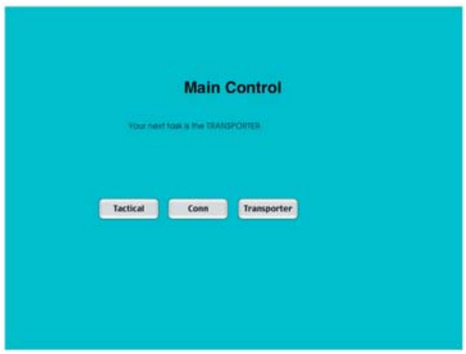


Figure 1. Main Control console

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

1. Tactical
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 one week 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 (Tactical, 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 aid 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 3 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!

6.2. Main control manual (change version, page 2)

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 3 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.

Finally, in order to ensure that our pilots are able to operate all systems in any field situation, we will be testing your on-the-fly thinking and ability to adapt. Being able to operate Starfleet systems under any external circumstance is imperative, particularly in emergency situations. You will be warned by the system halfway through the examination immediately before a change to the interface occurs. Please do your best to continue with the tasks and complete them as you did previous to the change.

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

Good luck, cadet!

6.3. Navigation manual

Starfleet Operations Manual

Model RD-x51 Navigation System (CONN)

This manual describes how to operate the RD-x51 Starfleet standard navigation system. Understanding how to operate this system is crucial for any Starfleet officer.

Figure 1 below shows the navigation console.

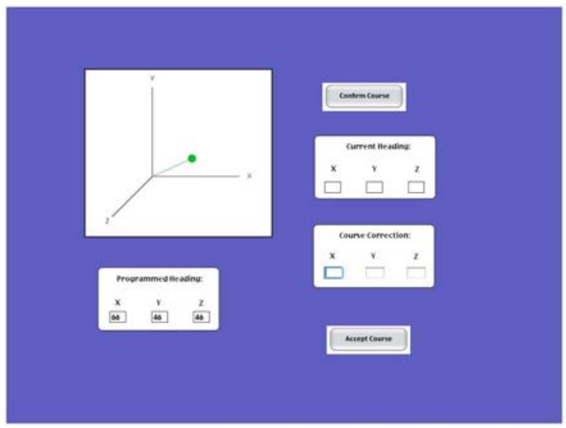


Figure 1. Navigation console

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.

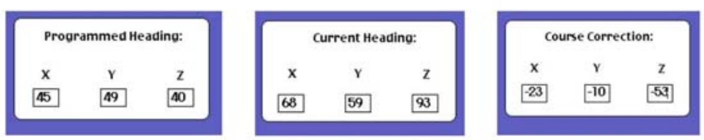
If the course idenfitied in 'Current Heading' does NOT match the 'Programmed Heading', you must compute the difference between the intended (programmed) heading course and the actual (current) course. To compute the difference, subtract the 'Current Heading' values from the 'Programmed Heading' values. 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, correcting the ship's course.

Review. Summary of Steps

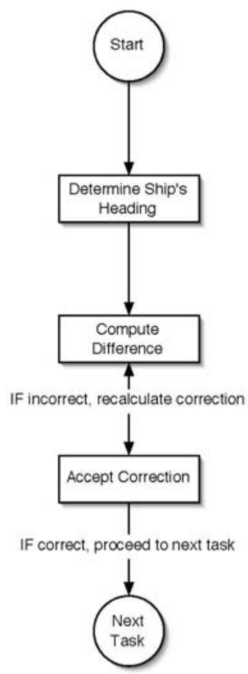
Overview of steps:

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'

Return to Main Control.



6.4. Phaser manual (control, layout change, and label removal)

Starfleet Operations Manual

Model MB-X55.1 Phaser Control Bank (TACTICAL)

This manual describes how to operate the MB-x55.1 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.

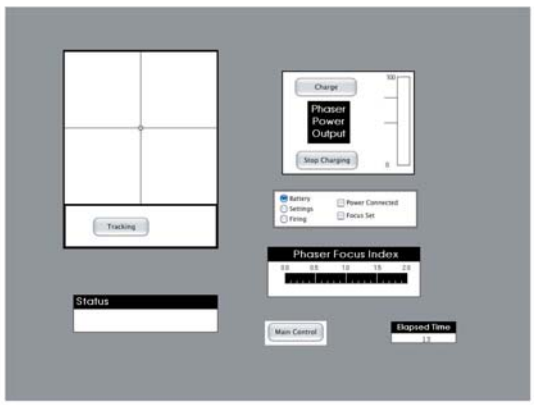


Figure 1. Phaser Control Bank interface

There are four essential steps involved in operating the phaser:

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 x55 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.

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.

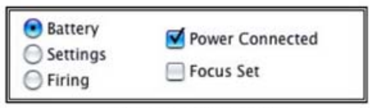


Figure 2. 'Power Connected'

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.

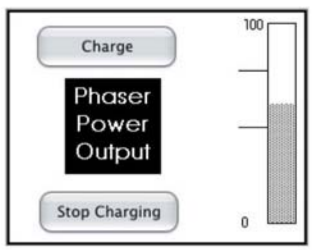


Figure 3. 'Charge', 'Stop Charging', and meter

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. Set Phaser Focus Index to desired focus
3. Click 'Focus Set'

The x55 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 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.

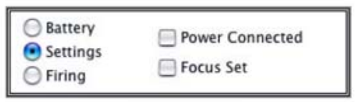


Figure 4. 'Settings'

The focus index must be set using the Phaser Focus scale. Click on the scale at the desired value. A black triangle will appear under the selected index (Figure 5). To change the selected index, simply click the new desired value. Setting the focus to approximately 2/3 level is acceptable for most targets. As you progress through training you will get a better feel.

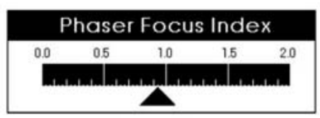


Figure 5. Focus Index chosen

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.

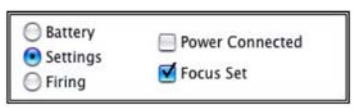


Figure 6. 'Focus Set'

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 x55 class phaser contains a sophisticated 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.

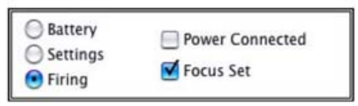


Figure 7. 'Firing' button engaged

Next, the tracking system is 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.

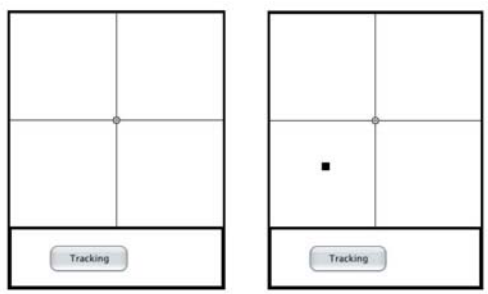


Figure 8. Tracking system in off (left) and on (right) states

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.

Moving the crosshairs closer to the target will produce better results. However, 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. *It is thus optimal to aim quickly and fire as soon as you are within close 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 onto 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. 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:

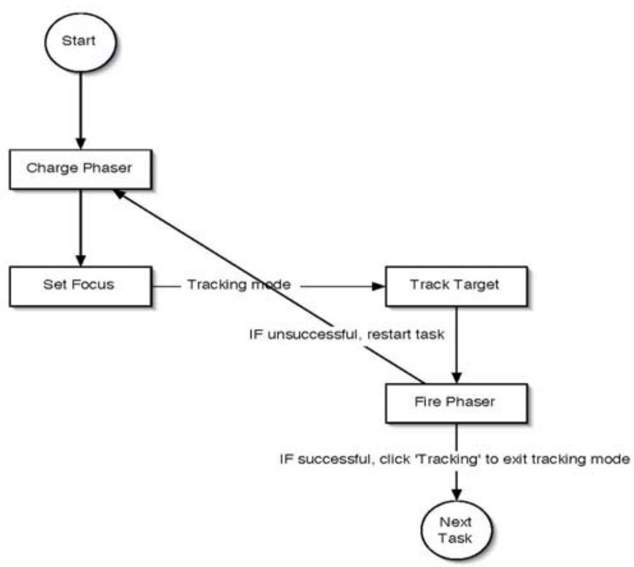
Step 1. Charge the Phaser
1. Click 'Power Connected'
2. Click 'Charge'
3. Wait until phaser charges the appropriate amount
4. Click 'Stop Charging'
5. Click 'Power Connected'

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 not, return to Step 1
4. If so, click 'Tracking'

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



6.5. Transporter manual (control, layout change, and label removal)

Starfleet Operations Manual

Model MB-x51.0 Manual Transporter System (MTS)

This manual describes how to operate the MB-x51.0 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.

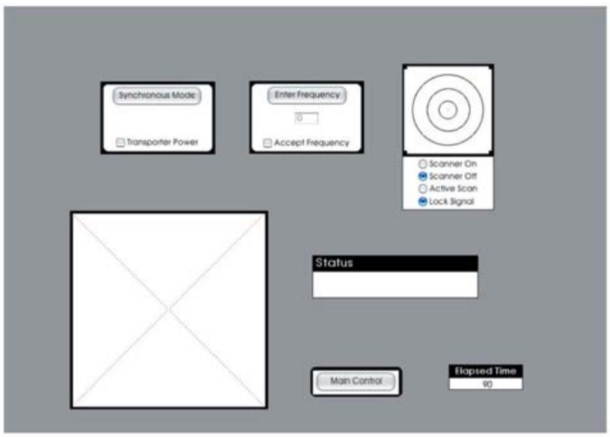


Figure 1. Manual Transporter System interface

There are four essential steps involved in operating the MTS:

1. Locking onto 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. Locking onto 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.

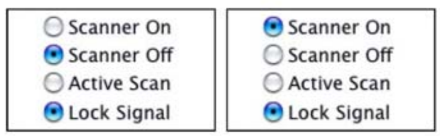


Figure 2. 'Scanner On' button

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' button to lock on. This is illustrated in Figures 3 and 4.

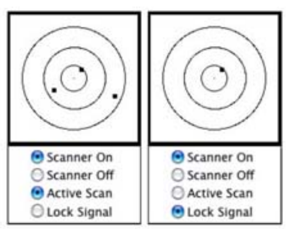


Figure 3 and 4. 'Active Scan' and 'Lock Signal'

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 responsiblity 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.

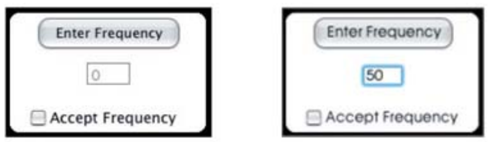


Figure 5 (left) and 6 (right). 'Enter Frequency'

After text entry is enabled, the frequency value between 1 and 100 should be entered on the keyboard. With experience you may adjust the value 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.

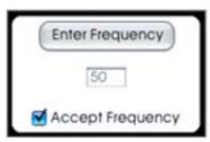


Figure 7. 'Accept Frequency'

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 x51 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.

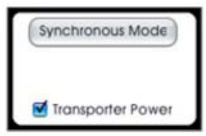


Figure 8. 'Transporter Power' engaged

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.

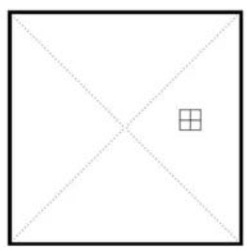


Figure 9. Tracking area and crosshairs

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 onto 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:

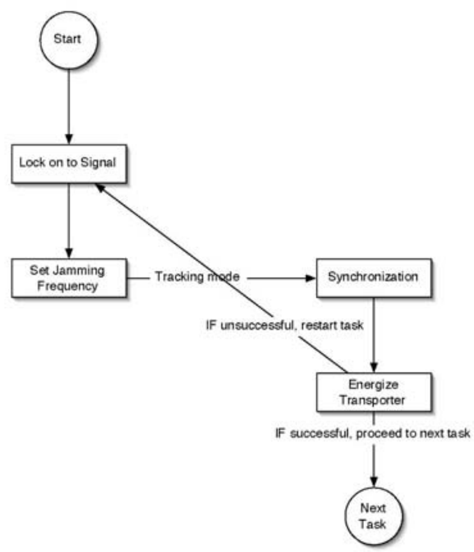
Step 1. Lock onto the Homing Signal
1. Click 'Scanner On'
2. Click 'Active Scan'
3. Wait until scanner homes in on a valid signal
4. Click 'Lock Signal'
5. Click 'Scanner Off'

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.



6.6. Phaser manual (color)

Starfleet Operations Manual

Model MB-X55.0 Phaser Control Bank (TACTICAL)

This manual describes how to operate the MB-x55.0 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.

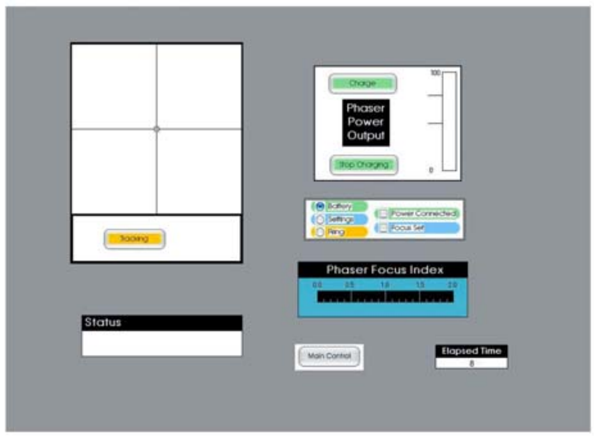


Figure 1. Phaser Control Bank interface

There are four essential steps involved in operating the phaser:

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 x55 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.

Several steps are involved in charging the battery. First, the battery must be connected to the power source by clicking the green 'Power Connected' box 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 green 'Charge' button (Figure 3) and waiting for the meter to fall within the safe range marked by the horizontal lines. At this point, the green 'Stop Charging' should be clicked.

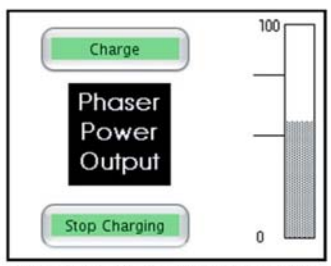


Figure 3. 'Charge', 'Stop Charging', and meter

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 green 'Power Connected' box (Figure 2). Unless the power has been disconnected, it will not be possible to operate other phaser controls.

TIP: all phaser charging related controls are colored GREEN.

Step 2. Setting Phaser Beam Focus

Overview of steps:

1. Click 'Settings'
2. Set Phaser Focus Index to desired focus
3. Click 'Focus Set'

The x55 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 first step in setting the Focus Index is to enable the alteration of current settings. This is done by clicking the blue 'Settings' box, as shown in Figure 4.

Figure 4. 'Settings'

The focus index must be set using the large blue Phaser Focus scale. Click on the scale at the desired value. A black triangle will appear under the selected index (Figure 5). To change the selected index, simply click the new desired value. Setting the focus to approximately 2/3 level is acceptable for most targets. As you progress through training you will get a better feel.

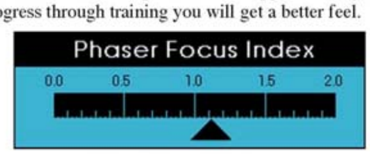


Figure 5. Focus Index chosen

Once the Phaser Focus Index has been set, the system must be locked. This is done by clicking the blue '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'

TIP: all phaser focus related controls are colored BLUE.

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 x55 class phaser contains a sophisticated 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 orange 'Firing' button, as shown in Figure 7.

Figure 7. 'Firing' button engaged

Next, the tracking system is activated by clicking the orange '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.

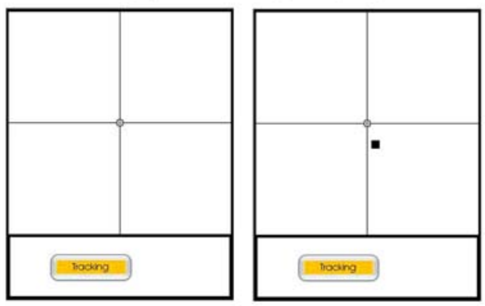


Figure 8. Tracking system in off (left) and on (right) states

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.

Moving the crosshairs closer to the target will produce better results. However, 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. *It is thus optimal to aim quickly and fire as soon as you are within close range rather than waiting until you are dead on.*

TIP: all firing/tracking related controls are colored ORANGE.

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 onto 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. Once you have done this by again clicking the orange 'Tracking' button, click 'Main Control', at the bottom, to move on to the next task.

Review. Summary of Steps

Overview of steps:

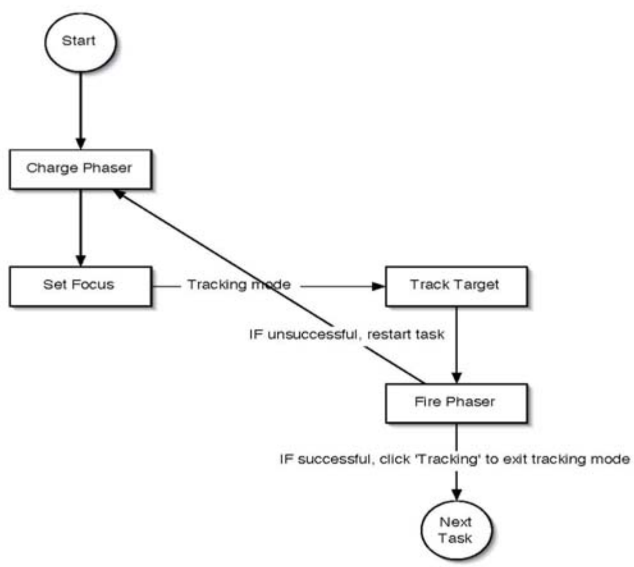
Step 1. Charge the Phaser
1. Click 'Power Connected'
2. Click 'Charge'
3. Wait until phaser charges the appropriate amount
4. Click 'Stop Charging'
5. Click 'Power Connected'

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 not, return to Step 1
4. If so, click 'Tracking'

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



6.7. Prize and point scale

PRIZE AND POINT SCALE

Prizes
1st place = $25
2nd place = $15
3rd place = $10

Conn
< 10 sec = +100

Transporter
< 13 sec = +100

Tactical
< 20 sec = +100

Task Bonus
+25 correct step
-50 incorrect step

Letter Recall Task Penalty
-200 incorrect

6.8. On-screen change instructions

Experiment 1

Beginning of test session (both conditions)

"In order to ensure that our pilots are able to operate all systems in any field situation, we will be testing your on-the-fly thinking and ability to adapt. Being able to operate Starfleet systems under any external circumstance is imperative, particularly in emergency situations. You will be warned by the system halfway through the examination immediately before the change occurs. Please do your best to continue with the tasks and complete them as you did previous to the change."

Immediately before change (layout change):

"As explained previously, system controls in the transporter task will now be rearranged in the following trials to simulate an emergency situation in which console damage has been sustained. Please do your best to complete the tasks as before."

Immediately before change (label removal):

"As explained previously, all labels in the phaser task will now be XXXed out in the following trials to simulate an emergency situation in which a system malfunction has occurred or system operation is required in the dark. Please do your best to complete the tasks as before."

Experiment 2

Beginning of test session (all conditions):

"In order to ensure that our pilots are able to operate all systems in any field situation, we will be testing your on-the-fly thinking and ability to adapt. Being able to operate Starfleet systems under any external circumstance is imperative, particularly in emergency situations. You will be warned by the system halfway through the examination immediately before the change occurs. Please do your best to continue with the tasks and complete them as you did previous to the change."

Immediately before change (all conditions):

"As explained previously, system controls in the next task will now be changed in the following trials to simulate an emergency situation in which console damage has been sustained. Please do your best to complete the tasks as before."

6.9. Post-experiment questionnaire

Experiment 1

How difficult was it for you to recall the task in this experiment (second day)?

When you made an error on the second day, what did you do to try to remember the right step or recover from the error?

How much effort did you put into the tasks, especially at testing?

How well do you think you did?

How hard was the letter recall task?

How hard was it to deal with the time pressure?

In the first step of the phaser task were you aware of making an error by clicking "Tracking" instead of "Power Connected"?

If yes, what made you think that "Tracking" was the right step at the time?

At some point during the experiment, all labels in the phaser task were XXXed out. How did this change affect the manner in which you completed the phaser task?

Experiment 2

How difficult was it for you to recall the task in this experiment (second day)?

When you made an error on the second day, what did you do to try to remember the right step or recover from the error?

How much effort did you put into the tasks, especially at testing?

How do you consider your performance (regardless of your score)?

How hard was the letter recall task?

How hard was it to deal with the time pressure?

In the first step of the phaser task were you ever aware of making an error by clicking "Tracking" instead of "Power Connected"?

If yes, what made you think that "Tracking" was the right step at the time?

Did you generally remember the buttons by their label or by their appearance and location? Did you remember the individual task components (e.g., beam focus, tracking system, etc.)? Please contrast this to how you learned the tasks on the first day.

Experiment 2 label removal

Please describe how you adjusted to the XXXing out of the labels in the Transporter task (i.e., what strategy you used and what went through your mind):

Experiment 2 layout change

Please contrast the original phaser interface from the changed interface and describe how you adjusted to the switch:

Please contrast the original Transporter interface from the changed interface and describe how you adjusted to the switch:

Experiment 2 color

Please describe how you adjusted to the interface change in the Phaser task. Did you

notice anything about the colors and did they help you in any way?

7. Appendix B

Demographic Questionnaire

Age:	20-30	30-40	40-50	50-60
Gender:	M	F		

Touch type (type without looking): Yes No

Years in practice: ________________________________

Years at this clinic: ________________________________

Years of computer use: ________________________________

Comfort using a computer, 1 (most) to 10 (least): ________________________________

Years of EMR use: ________________________________

Comfort using an EMR, 1 (most) to 10 (least): ________________________________

7.2. SUS questionnaire

Baylor EMR Survey

EMR Survey

Please respond to the following statements about the electronic medical records system you have just used. "System" refers specifically to the data entry form for the patient vital signs information. On a scale from 1 to 5, indicate how much you disagree or agree with the following statements. Mark how much you disagree or agree by checking one of the boxes next to the statement.

Statement	Strongly disagree				Strongly agree
1. I think that I would like to use this system frequently.	1	2	3	4	5
2. I found the system unnecessarily complex.	1	2	3	4	5
3. I thought the system was easy to use.	1	2	3	4	5
4. I think that I would need the support of a technical person to be able to use this system.	1	2	3	4	5
5. I found the various functions in this system were well integrated.	1	2	3	4	5

Baylor EMR Survey

	Strongly Disagree 1	2	3	4	Strongly Agree 5
6. I thought there was too much inconsistency in this system.	1	2	3	4	5
7. I would imagine that most people would learn to use this system very quickly.	1	2	3	4	5
8. I found the system very awkward to use.	1	2	3	4	5
9. I felt very confident using the system.	1	2	3	4	5
10. I needed to learn a lot of things before I could get going with this system.	1	2	3	4	5