Applying Human Factors Engineering to Medical Device Design: An Empirical Evaluation of Patient-Controlled Analgesia Machine Interfaces

by

Laura Lin

A thesis submitted in conformity with the requirements
for the degree of Master of Applied Science

Department of Mechanical and Industrial Engineering
Institute of Biomedical Engineering
University of Toronto

September 1997

© Copyright by Laura Lin 1997

National Library of Canada
Acquisitions and Bibliographic Services
395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale du Canada
Acquisitions et services bibliographiques
395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-29431-5

Canada

ABSTRACT

This thesis was motivated by the fact that poor design of medical devices plays a significant role in inducing human error. The hypothesis explored in the thesis is that medical equipment can be made safer, more efficient, and easier to operate by applying human factors to equipment design. The Lifecare 4100 PCA Plus Infuser, a commonly used medical device was chosen as the testbed. The thesis first describes an analysis of the Lifecare PCA interface which employed cognitive task analysis and human factor design principles. Guided by the results of the analysis, a redesigned interface was developed. A set of empirical evaluations was then conducted to compare the redesigned interface with the existing Lifecare PCA interface with novice and experienced PCA users. The results of the evaluations provided evidence of improved efficacy of the redesigned interface over the existing Lifecare PCA interface; programming times were significantly faster and fewer errors were made with the redesigned interface than with the existing one. Additionally, a significant portion of the novice population reported lower mental workload with the redesigned interface, and the experienced population reported comparable levels of workload for both interfaces, despite having much more extensive experience with the existing one. These findings demonstrate that PCA machine operation can be improved by adopting a human factors approach to interface design. They also lend support to the broader prospect of applying these methods to other devices to improve medical device safety.

ACKNOWLEDGEMENTS

I would like to extend my sincere gratitude to my advisor, Dr. Kim Vicente whose expertise, guidance and support have made this work possible, and to my co-advisor, Dr. John Doyle whose vital contributions and vision have led this project since its inception.

I would also like to thank Bill Muto for providing comments on earlier work, Sue Bogner for suggesting MDRs, David Gaba for his support and comments, Karen McCormick for her insights, and the recovery room nurses at The Toronto Hospital for their invaluable input to this study. I would also like to gratefully acknowledge the support from NSERC equipment and research grants.

Many thanks also go to my colleagues in CEL and ETC lab, who have made graduate studies a lively and rewarding experience, and finally, to the many students, staff and researchers at IBME who have been both supportive and inspirational.

TABLE OF CONTENTS

LIST OF TABLES AND FIGURES

Tables

Figures

1. INTRODUCTION

As many as 100,000 preventable deaths or serious injuries occur each year in the United States' health care institutions as a result of medical accidents (Van Cott, 1993). A substantial number of these accidents are related to the misuse of medical devices (Burlington, 1995; Carstenson, 1995). It has been long recognized that the most common trigger of medical device mishaps is human error. However, what is less often acknowledged is that poor design of medical devices plays a significant role in predisposing medical device operation to human error (Bogner, 1994a; Bogner, 1994b; Hyman, 1994; Rachlin, 1995; Burlington, 1995; Cooper et al., 1978). This thesis explores the hypothesis that medical equipment can be made safer, more efficient, and easier to operate by applying human factors to equipment design.

Recently, there have been a number of studies on human factors of medical equipment. Many of the studies focus solely on analyses, such as task analysis, of commercially available products without translating their results to design recommendations (e.g., Henriksen et al., 1991; Serig, 1991; Callan et al., 1991; Moll van Charante et al., 1992). Where design recommendations are made, the predominant approach is ad hoc or piecemeal in nature (e.g., Bancroft, 1989; Obradovich et al., 1996). Moreover, none of these studies have empirically demonstrated that their resulting interface design/redesign is advantageous over the existing design.

The objective of this study is twofold: (1) to demonstrate a coherent human factors guided process of analysis and redesign of a medical device interface, and (2) to test the efficacy of the redesigned interface through empirical evaluation. With these objectives in mind, a commonly used medical device, a Patient-Controlled Analgesia (PCA) machine, was selected as the testbed. PCA machines are ideal candidates for analysis because they have been identified as a source of stress for its users (McConnell, 1995), and have been cited as being prone to human error (Callan, 1990).

2. PATIENT-CONTROLLED ANALGESIA

This section provides a brief background on PCA machines and an overview of the literature on PCA-related medical accidents. This is followed by an analysis and redesign of the interface for the Lifecare 4100 PCA Plus II Infuser, a commercially available PCA device currently used in U.S. and Canadian hospitals. The Lifecare PCA was chosen for this study based on its pervasive use in several local hospitals and its accessibility for study.

2.1 Background

Patient-controlled analgesia was developed in response to the need for a more effective method of administering analgesics. Traditional intramuscular opioid regimens were found to be ineffective in providing adequate pain relief for over half of patients suffering from post-surgical pain (Ferrante et al., 1990). PCA machines provide an alternative system of delivery which is based on the concept of smaller, on-demand, patient-controlled, intermittent dosing. The PCA system of delivery uses traditional opiate analgesics (e.g., morphine) and the traditional intravenous route of administration, but it has been found to provide more effective pain relief in addition to reduced sedation, and lower narcotic consumption (Ferrante et al., 1990; Smythe, 1992).

In the U.S., PCA machines were first introduced to clinical practice in 1984. Currently, PCA machines are used in approximately 96% of all U.S. hospitals with 100 or more beds (Ready, 1995). The prevalence of PCA pain therapy underscores the importance of conducting comprehensive assessments of PCA safety. To a limited extent, effectiveness and safety have been addressed, but only in terms of patient/nurse satisfaction with the pumps and its mechanical safety, e.g., accuracy and reliability of infusion rate (Owen et al., 1986; Sawaki et al., 1992; Isley et al., 1994). While there is no dispute that mechanical soundness is a significant factor in device safety, it is important to recognize that failures also occur in device operation at the user-machine interface, and that user satisfaction does not necessarily correlate with objective measures of performance, nor does it guarantee invulnerability to errors in device operation.

2.2 Adverse Incidents

To determine the extent to which user errors occur with PCA devices and the manner in which errors are treated by the medical community, a review of medical accidents reports was conducted. This first involved a literature search of PCA accidents published in medical and biomedical instrumentation journals; then a review of Medical Device Reports related to PCA devices. The results of the literature search and review of Medical Device Reports are discussed in turn.

2.2.1 Literature Overview

Literature reports of PCA-related mishaps or adverse incidents generally fall into one of three categories: mishaps caused by mechanical failure, human error, and adverse reaction to the drug. According to figures published in 1990, adverse drug reactions and mechanical failures together account for less than half of all cases. It was estimated that 67% of problems associated with PCA use are attributable to user error, where "user errors" include patient tampering of the device and errors made by nurses in setting up the PCA pump, e.g., "misprogramming" the drug concentration (Callan, 1990). The latter type of error appears to constitute the predominant source of problems with PCA use (Cohen, 1993).

In general, the literature on PCA mishaps places blame for programming errors on the user. As one author surmised after investigating one such incident involving PCA misprogramming, "adequate instruction for use had been provided but had been disregarded" (Callan, 1990). In most cases, the recommendations for alleviating the problem of user errors focused on improving training, increasing supervision, or simply "paying more attention" (Cohen, 1993; Patt et al., 1993). While training and supervision are important aspects in ensuring the safety of medical device operation, they are limited in their impact because even the most highly trained, experienced, and conscientious people will inevitably commit errors. What the PCA literature has neglected to recognize is the notion that interface design has a significant impact on human performance, and that a poorly designed interface represents a latent failure that can be *triggered* by the user.

2.2.2 FDA Medical Device Reports

In order to gain a more accurate estimate of the prevalence of PCA mishaps, and the breakdown of their causes, a review of Medical Device Reports (MDRs) was conducted. In the U.S., hospitals are required to file MDRs on equipment-related accidents as part of the U.S. Food and Drug Administration's (FDA) post-market surveillance on medical devices. Medical Device Reports for PCA pumps were obtained from the FDA's Freedom of Information Office. An analysis was conducted only on the MDRs specific to the Lifecare PCA pump investigated in this thesis for a randomly chosen year, 1993, in order to determine the causal origins of the accidents.

A total of 2,017 reports were reviewed and categorized according to three key entries provided in the MDRs: (1) the reported cause, (2) patient outcome, and (3) whether the machine was in use at the time of the incident. Of all the reports, only a small fraction of incidents (308 out of 2017) were known to have occurred while the PCA device was in use. Only these incidents were further analyzed. The cause and consequences of these 308 incidents are reported in Table 2.1.

Table 2.1 Medical Device Reports on PCA mishaps in which the device was known to be in therapeutic use. Cells represent the number of reports that fell into the categories for incident causes and patient outcome.

Patient Outcome / Incident Cause	No Adverse Patient Reaction	Serious Injury or Death	Unknown	Total
Malfunction	**246**	6	13	265
Human error	8	**13**	0	21
Unknown	3	9	0	12
Other*	2	8	0	10
Total	259	36	13	308

* e.g., terminal illness

An analysis of the MDRs focused only on the incidents which had **known causes** and **known patient outcome** (bolded numbers in Table 2.1). Out of the incidents with known outcomes (i.e., "no adverse patient reaction" or "serious injury or death"), there were 252 malfunctions, which greatly outnumbered the 21 human error related incidents ($\chi^2(1)=195.46$,

$p<0.0005$, one-tailed). However, only 6 out of the 252 malfunctions led to an adverse patient outcome, while 13 out of 21 of the human errors led to an adverse patient outcome. Stated as conditional probabilities:

$$P(\text{serious injury or death} \mid \text{malfunction}) = 0.02$$
$$P(\text{serious injury or death} \mid \text{human error}) = 0.62$$

To test whether there was any significant interaction between incident cause ("malfunction" or "human error") and patient outcome ("no adverse patient reaction" or "serious injury or death"), a χ^2 test was conducted. The results showed a statistically significant interaction between incident cause and patient outcome ($\chi^2(1)=106.015$, $p<0.0005$, one-tailed). These statistics would lead us to believe that malfunctions occur much more frequently than human errors, but also that human error is much more likely than a malfunction to cause serious injury or death. However, some studies suggest that human error, particularly those with no serious consequences to the patient, is vastly under-reported (McConnell, 1995) for reasons such as fear of blame, risk of litigation, and criticism of competence. Hence the risk of patient injury associated with human error, 62% as estimated by MDR analysis, is likely overestimated.

An alternative method of estimating the risk of injury or death associated with human error is to examine only those incidents with serious outcomes. Because errors with serious consequences are more difficult to overlook or conceal, they are more likely to be reported and thus, may provide a more accurate representation of the breakdown of causes. In examining only the incidents with adverse patient outcome (serious injury or death), it was found that human error was twice as likely as malfunctions to have caused the incident ($\chi^2(1)=2.778$, $p<0.05$, one-tailed):

$$P(\text{malfunction} \mid \text{serious injury or death}) = 0.32.$$
$$P(\text{human error} \mid \text{serious injury or death}) = 0.68$$

This is consistent with other reports in literature that identify human error as the most common cause of PCA-related medical accidents (Callan, 1990).

In order to determine the *types* of errors that occurred most frequently in human error related incidents, a qualitative analysis was conducted on the MDRs. In this analysis, all 21

human error-related incidents were reviewed and classified according to the *nature of the error* based on the description of the incident provided in the MDRs. Table 2.2 below provides a summary of the MDRs in which human error was implicated as the cause. As can be seen from the table, three types of errors emerge:

(a) programming errors (errors related to programming incorrect settings),

(b) errors in setup protocol (errors related to mechanical assembly or setup), and

(c) errors in judgment (prescribing inappropriate dosages).

The main observation drawn from this analysis is that "programming errors" are the most common type of human error in PCA use, accounting for 18 out of the 21 human errors. Furthermore, as many as 16 of the 18 programming errors involved setting an incorrect *drug concentration*, all of which led to overdelivery of medication. Interestingly, in the MDRs which provided information on the actual settings that had been incorrectly entered, all of the incorrect settings happened to match the Lifecare PCA machine defaults. This is highly suggestive that many of the errors resulted from users selecting an inappropriate machine default during the PCA programming procedure. As will be explained in later discussions, inappropriate machine defaults are an example of design deficiencies that may be inducing human errors in PCA operation.

In summary, several conclusions were drawn from the MDR analysis: first, device malfunctions are the most frequently reported incidents; second, although there are fewer reports of human errors than malfunctions, human error is *more likely* than malfunction to lead to serious injury or death; third, the most common human errors related to PCA use are programming errors. These conclusions indicate that user programming errors constitute the most serious threat to patient safety for Lifecare PCA pumps.

2.3 Abbott Lifecare 4100 PCA Plus II

The remainder of this section discusses the Lifecare PCA interface and attempts to elucidate the design problems which may have induced these user errors. First, a description of the device is provided, followed by an analysis of the interface, and finally a description of a redesigned interface.

Table 2.2. Description of the 21 Medical Device Reports on human error-related incidents involving the Lifecare PCA pump filed in 1993.

Error Description	Patient Outcome	Error Category
programmed 1 mg/ml instead of 5 mg/ml causing overdelivery	no patient harm	programming error
programmed 150 mg bolus causing overdelivery	serious injury, patient became comatose, narcan was required to bring back to responsive state	programming error
programmed 0 minute lockout causing overdelivery	serious injury, patient suffered respiratory depression	programming error
Dr. prescribed inappropriate dose	death	error in judgment
programmed 1 mg/ml instead of 5 mg/ml causing overdelivery	serious injury, patient had to be resuscitated	programming error
"programmed incorrectly" causing overdelivery	death	programming error
programmed 1 mg/ml instead of 10 mg/ml causing overdose	serious injury, patient become unresponsive and required narcan	programming error
"programmed a lower concentration" than prescribed	no adverse patient reaction, but later died of causes related to terminal cancer	programming error
programmed 1 mg/ml instead of 5 mg/ml causing overdelivery	serious injury, patient became lethargic and required narcan	programming error
programmed 1 mg/ml instead of 5 mg/ml causing overdelivery	no adverse patient reaction	programming error
programmed 0.1 mg/ml instead of 1 mg/ml causing overdelivery	serious injury	programming error
programmed 1 mg/ml instead of 10 mg/ml causing overdelivery	serious injury, patient suffered respiratory depression and required narcan	programming error
"misprogrammed" causing overdelivery	serious injury	programming error
programmed 1 mg/ml instead of 5 mg/ml causing overdelivery	no adverse patient reaction	programming error
programmed 5 mg/ml instead of 10 mg/ml causing overdelivery	no adverse patient reaction	programming error
clamp was inadvertently left on after setup for 3 hours causing underdelivery	no adverse patient reaction	error in setup protocol
"misprogrammed"	no adverse patient reaction	programming error
Dr. prescribed too high of a dosage	serious injury, patient suffered respiratory depression	error in judgment
"programmed incorrectly" causing overdelivery	serious injury, patient suffered respiratory arrest and required narcan to resuscitate	programming error
"user programming error" causing overdelivery	no adverse patient reaction	programming error
"error in programming" causing overdelivery	no adverse patient reaction	programming error

2.3.1 Current Interface

The Abbott Lifecare 4100 PCA Plus II Infuser is illustrated in Figure 2.1. The interface used by nurses to program the pump is the primary focus of the discussion that follows.

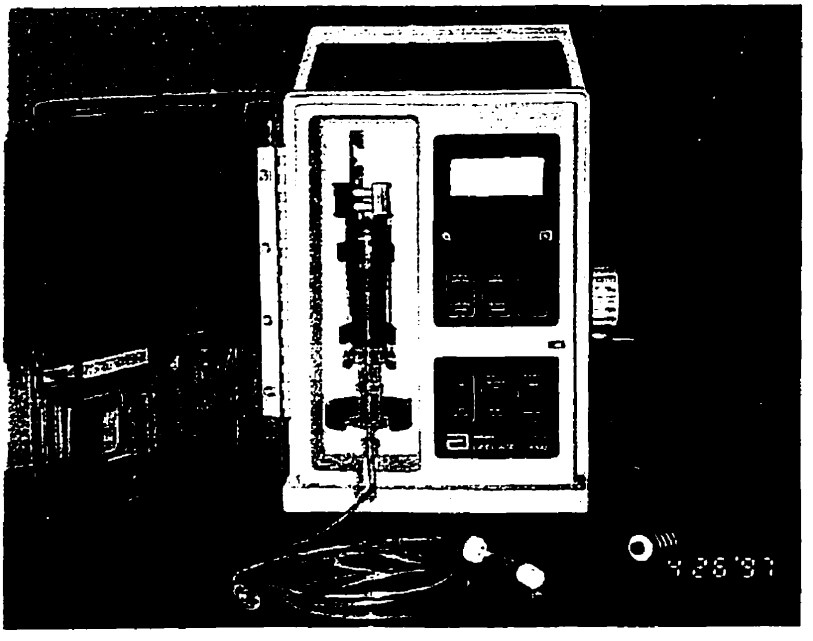


Figure 2.1 The Lifecare 4100 PCA Plus II Infuser.

Physical Interface Description

The programming interface of the Lifecare PCA is illustrated in Figure 2.2. It consists of a touch switch control panel, a liquid crystal display (LCD), and a light emitting diode display (LED). The LCD displays instructional messages which prompt the user to enter various settings (discussed below). The settings are entered using the touch switch control panel, which consists of 12 soft touch switches. The touch switches, however, are spatially organized and visually represented in a manner which suggests to the user that there are only six touch switches; each physical (hard) touch switch contains at least two labels (two soft touch switches). Refer to Table 2.3 for a complete listing of the touch switches and their functions. The final element of the interface is the LED, a redundant display which is dedicated to displaying numerical information whenever the user manually enters values for various operating parameters.

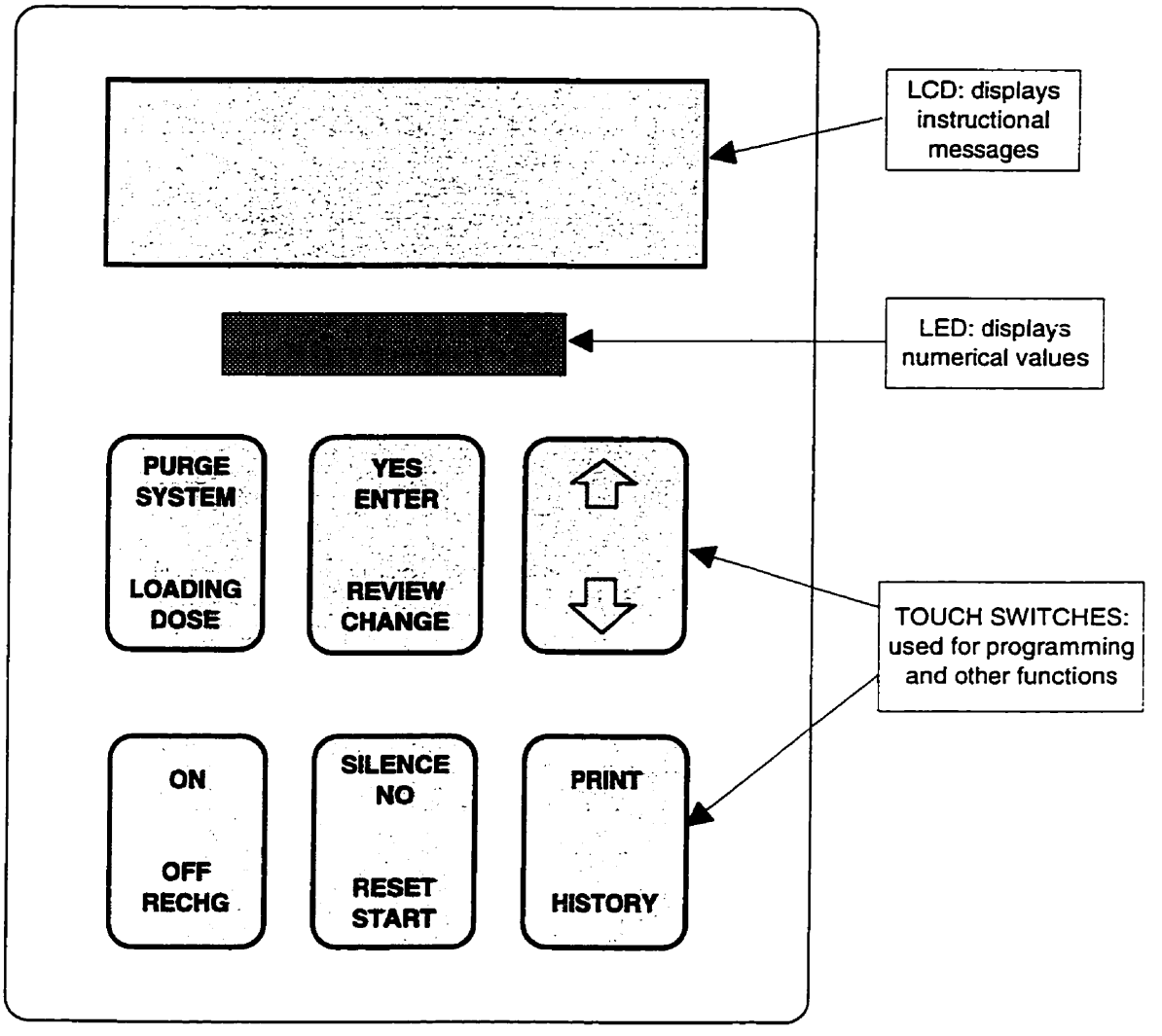


Figure 2.2. Interface of current Lifecare PCA machine.

Table 2.3. Current interface's touch switches and their functions.

Touch Switch	Function
PURGE SYSTEM	Begins purging (which removes slack from the tubing after a new vial/injector is installed).
LOADING DOSE	Initiates the loading dose programming sequence. Begins infusion after the dose has been set.
YES/ENTER	"YES" responds to a "yes or no" message prompt. "ENTER" inputs the setting of an operating parameter into the system.
REVIEW/CHANGE	During programming mode, this touch switch allows the user to scroll back to the previous screen in order to correct any errors in the parameter settings After programming is completed, this touch switch allows the user to make changes to parameter settings by scrolling forward through the programming sequence.
↑ ↓	Increases or decreases a numerical value displayed on the LCD/LED.
ON	Activates a cold start if pressed longer than 4 seconds or if the machine has been off for longer than 60 minutes.
OFF/RECHG	Turns the machine off, retaining settings and history for 60 minutes.
SILENCE/NO	"SILENCE" temporarily mutes audible alarms during an alarm condition. "NO" responds to a "yes or no" message prompt.
RESET/START	Stops or starts continuous infusion.
HISTORY	After programming is completed, this touch switch allows the user to review the parameter settings entered. During therapy, the touch switch can be pressed to display the parameter settings as well as the history of all doses delivered.
PRINT	Executes a printout of time, date, parameter settings, and dose history.

Operational Description

To program a Lifecare PCA pump, the user must advance through a sequence of "screens" displayed on the LCD. The series of screens, also referred to as the programming sequence, consists of instructional messages which prompt the user to enter the settings for operating parameters, such as drug concentration, drug dosages, and safety limits. Parameter settings, prescribed by the patient's physician, are provided to the nurse on a PCA order form.

The Lifecare PCA pump can be programmed to administer analgesic in one of three *modes*:[1] discrete, continuous, or a combination of both. In the PCA mode (discrete mode of operation), the patient can request a *dose* of analgesic by pressing the button on a handset. Depending on the time since the last dose, and the cumulative amount delivered in the last 4

[1] italics in this section denote an operating parameter.

hours, the machine delivers a dose through an intravenous line. The minimum time period between doses is specified by the *lockout interval*, while the cumulative total a patient is allowed to receive in 4 hours is specified by the *4 hour limit*. In the CONTINUOUS mode, analgesic is administered continuously at a specified baseline infusion rate (*continuous dose*[2]). The 4 hour limit restricts the total amount that can be administered in this mode, as well. Finally, the baseline infusion and the discrete mode of delivery can be combined by programming the PCA pump for the PCA + CONTINUOUS mode of operation.

As the user is programming, errors may occur while entering the settings. If they are detected, the user can recover from errors while still in the programming mode, or by making the correction after the programming procedure is completed. While in the programming mode, the user can press the "REVIEW/CHANGE" touch switch, which allows him/her to scroll backwards to previous screens to locate the parameter to be changed and make the amendment. If the user wants to correct an error after the programming sequence is completed, he/she uses REVIEW/CHANGE to scroll forward through the sequence of operating parameters to locate the parameter to be changed and make any necessary corrections.

2.3.2 Analysis of the Current Interface

Analysis of the Lifecare PCA interface was conducted in two previous studies by Isla & Lin (1993) and Doniz & Harkness (1994). A brief description of the methodology and the results they obtained from the analysis are provided here in order to outline the underlying principles that guided the process of redesigning the interface (for a more detailed discussion, see Lin et al., 1995). The analysis is discussed in two sections, the first describing cognitive task analysis and the second describing human factors design principles.

2.3.2.1 Cognitive Task Analysis

Redesigning the interface to maintain functionality and conform to user needs, required a thorough understanding of both the system and the psychological demands it places on the user. To acquire this understanding, cognitive task analysis (CTA) was employed

---

[2] "continuous dose" and "continuous rate" are used synonymously throughout the thesis.

following the methodology used by others in previous research in this area (Charante et al., 1992; Yue et al., 1992; Cook et al., 1991).

Methodology

Data collection for CTA first involved bench tests. The bench tests aided in identifying the characteristics of the device which make its operation prone to errors. This led to a thorough mapping of the internal system structure which was instrumental in identifying interface design deficiencies.

Next, field observations were conducted in the recovery room of The Toronto Hospital, where nurses were observed programming the device and were also interviewed. The study of the machine and its users in context drew attention to environmental factors which placed added demands on the users.

Finally, after a preliminary design was developed based on the results of the bench tests and field study, the redesigned interface was demonstrated to recovery room nurses to obtain a

preliminary assessment of the redesign and practical suggestions. Several design changes were directly instigated by the recommendations and comments obtained from the nurses (see Doniz & Harkness, 1994 for details). In addition, the demonstration also served to foster user acceptance of the new design ideas.

CTA Results

The bench testing led to an analysis of state transition diagrams which summarize the structure of the tasks involved with PCA programming. State transitions at three levels of detail were analyzed:

1) general flow of activities (function flow diagram of the "subtasks" performed to complete the task of programming - see Figure 2.3),

2) decisions and actions required of each subtask (Figure 2.4), and

3) a detailed mapping of sequential LCD messages and user input (see Isla & Lin, 1993).

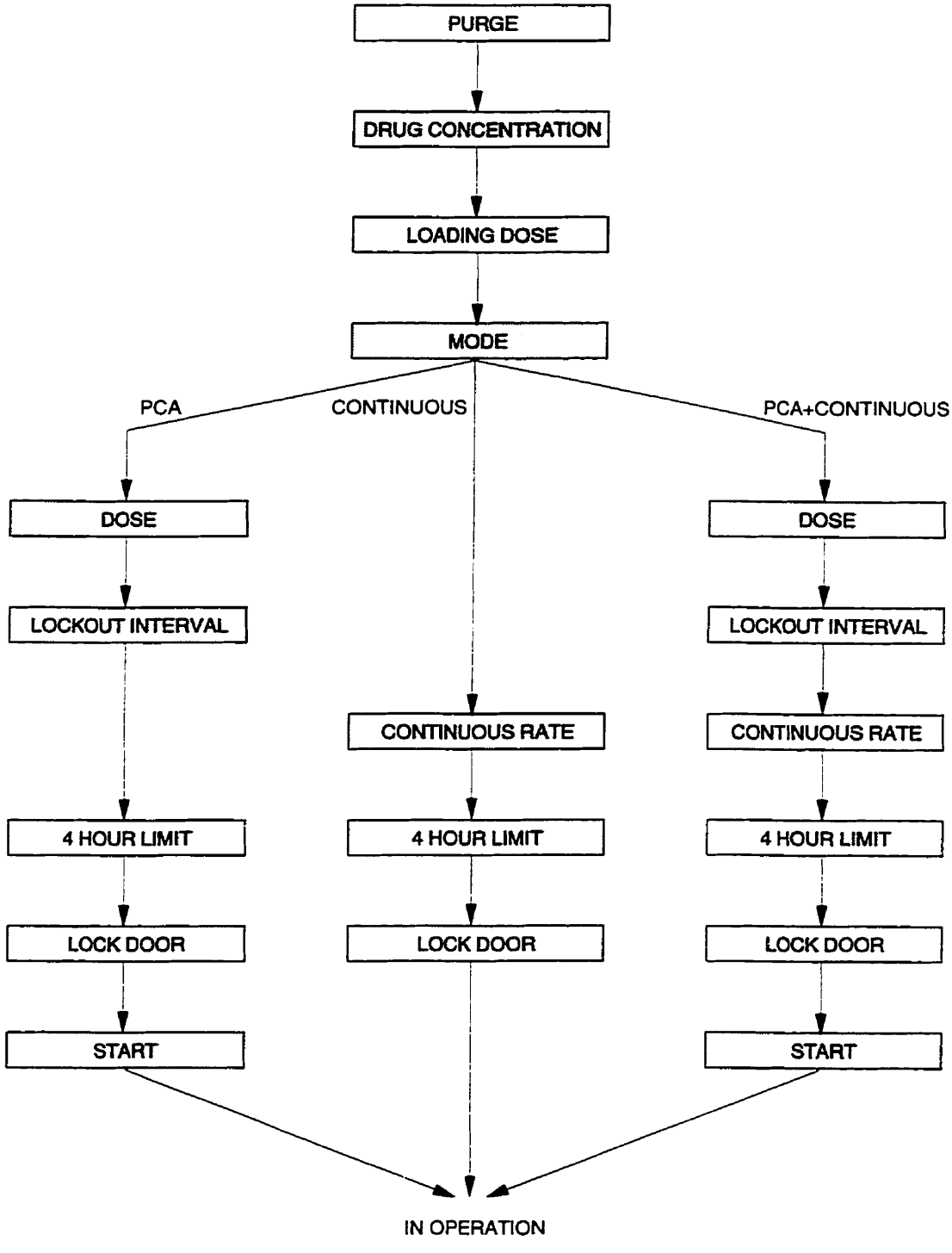


Figure 2.3. Function Flow Diagram for the current interface.

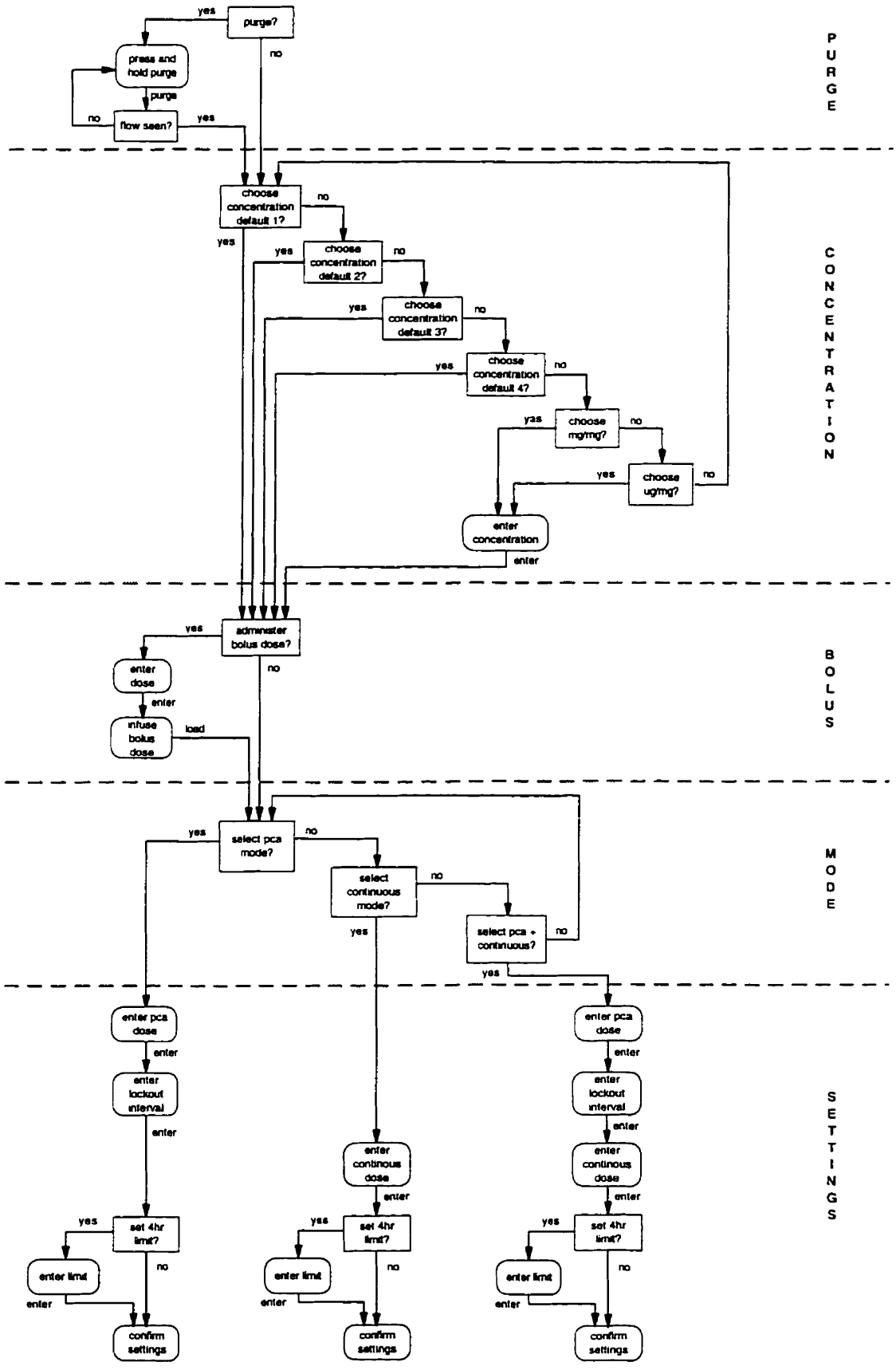


Figure 2.4. Decision/action strucure of programming sequence for the current interface. The column at the right shows the programming stages.

The state transitions at the intermediate level of detail proved to be the most instrumental in the development of the new design. Figure 2.4 shows the sequence of decision and actions for the subtasks within the context of the overall programming sequence for the current PCA interface. In this decision/action structure, message screens in the programming sequence are categorized into "decisions" and "message-guided" actions. A "decision" (denoted by a square in Figure 2.4) refers to a message screen in which the user is prompted to make a choice between alternatives. In contrast, a "message-guided" action (circle) refers to a message screen which instructs the user on what action to perform.

The main observation drawn from the assessment of subtask structures was that many of the subtasks embodied unnecessary complexity. For instance, consider the subtask of mode selection (see the 'MODE' programming stage in Figure 2.4). In this subtask, the user must select one of the three operating modes, based on the information written on the PCA order form. The programming sequence presents the options serially in three separate message screens. Since the user can only view one screen at a time, this forces the user to make 3 separate decisions. Although the user is making a choice between three related alternatives, the options are portrayed as three unrelated choices by the interface. Not only is this procedure time consuming, but it also neglects the need for a global view of the decision.

A second example of structure complexity is the procedure involved in setting the concentration (see the 'CONCENTRATION' programming stage in Figure 2.4). A series of defaults are presented to the user, who is required to make a decision to accept or reject each of the defaults based on that prescribed on the PCA order form. Note that the default options are presented serially in separate message screens similar to the manner in which the mode options are presented. Furthermore, the concentration defaults presented by the interface are not the standard operating values used at The Toronto Hospital (refer to Doniz & Harkness, 1995). Therefore, the user must reject all of the default screens in order to reach the screen which allows him/her to select the appropriate concentration units and then to enter the prescribed concentration value. This example highlights two factors contributing to unnecessary complexity in the programming structure, serial presentation of decision options and inappropriate machine defaults. Coincidentally, the most common programming error

identified in the MDR analysis in section 2.2.2 is related to entering an incorrect concentration setting. As stated earlier, it is very likely that these errors were a result of selecting one of the wrong concentration defaults in the programming sequence since the incorrect settings found in the MDRs all happen to match one of Lifecare PCA machine's concentration defaults.

Other examples of unnecessarily complex task structures were also identified (see Isla & Lin, 1993), many of which revolved around the decision-making activities. These observations anticipate the need for significant changes to the interface.

Information Requirements

Having determined the structure of the programming task, an analysis of the information requirements associated with each subtask was performed. This aspect of CTA identifies the information that should be included on an interface to perform the subtasks involved in programming. To describe the extent to which user activity is supported by information on an interface, it is important to distinguish between 'surface control' and 'deep control' of a system (Vicente & Rasmussen, 1992). The user is relying on surface control if he/she is depending on perceptual cues provided by the interface in order to decide what actions are appropriate. In contrast, if they have to go beyond the perceptual features, relying instead on their conceptual understanding (i.e., "deep" model) of the device's internal structure in order to determine which actions to perform, then they are relying on deep control. Although both control modes can be useful, surface control tends to be faster and requires less effort since it allows operators to control the system by taking advantage of their powerful pattern recognition capabilities. The perceptual features of the display serve as external cues which relieve the burden of working memory load. In contrast, deep control is slower and requires more effort because it involves analytical reasoning.

The current Lifecare PCA interface supports surface control through the sequential display of instructional messages, which acts as an external cue in guiding the user through the programming sequence. This cue allows the user to "perceive and act", thus reducing the dependency on working memory. However, the effectiveness of this cue is compromised by the neglect of other potential cues, particularly the lack of spatial organization of the touch

switch panel, lack of external representation of the system structure, and inconsistent mapping of touch switch labels to their functions. These are discussed below.

Lack of spatial organization. The lack of meaningful grouping (e.g., procedural order) and misleading visual grouping of the touch switches fails to reinforce the structure of the programming sequence. The logic of the sequence is seemingly arbitrary without sufficient cues. This is symptomatic of a deeper area of concern discussed in the next point.

Lack of external representation. Although the user is guided through a sequence of tasks (Figure 2.3), there are no indications as to how many parameters there are to program, what order the sequence follows, or where the user's current location is in the sequence. The lack of cues forces unnecessary memory load, undermining the advantages of surface control.

Inconsistent mapping. An inconsistent cue can also pose cognitive demands on working memory. Bench tests yielded several inconsistencies in touch switch labeling, in which functions are misrepresented by the wording of the touch switch labels. For instance, RESET/START does not reset the machine, it starts and stops infusion. Touch switch functions and labels are further discussed in the next section (2.3.2.2) in the context of human factors design principles.

Inconsistencies and lack of cues can impose cognitive demands, forcing the user to rely on deep control to perform the programming task. In developing a new interface design which relaxed this dependency on deep control, it was necessary to identify the information or knowledge the user must draw upon to accomplish decision making.

An information requirements analysis was performed by Isla & Lin (1993) in order to determine the information that was needed by users to complete the task of programming. This information was compared with the information that is actually provided on the interface (see Isla & Lin, 1993 for details). The analysis demonstrated that provision of information for the programming tasks is adequate, but also that error recovery tasks are not supported effectively by the guidance messages (insufficient cues to aid the user in performing the task). This motivated the need for a redesigned interface which reduces cognitive demand by providing the appropriate external cues to encourage effective surface control of the system, and by simplifying tasks most dependent on deep control.

Context of Use and User Population.

The field studies and interviews with recovery room nurses at The Toronto Hospital provided the necessary information needed to characterize the PCA device's primary users and their work environment.

At The Toronto Hospital, PCA pumps are usually programmed in the recovery room where patients are received after their surgical procedure is completed. Recovery room nurses are on a rotational schedule in which they are assigned responsibility for PCA programming for 1 to 3 weeks each month. When they are on the assigned rotation, a nurse may program anywhere from 2 to 10 PCA pumps a day. However, when the case load is heavy, a nurse may program pumps for incoming patients, even though she is not on the assigned rotation. All recovery room nurses have undergone training and are certified in PCA usage.

The number of patients in the recovery room at any one time varies widely, from 2 patients (usually in the early morning and late night shifts) to 15 or more patients during peak hours. During these peak times, the recovery room becomes active, noisy, and congested with people, producing constant interruptions of the nurses' activities. This highly demanding work environment appears to strongly influence the temperament and stress levels of nurses.

During any given day, 30 to 70 patients may be transported through the recovery room where the nurses set up PCA therapy as well as any other equipment needed to monitor the patients as they stabilize. Once stable, patients are transported back to their rooms, along with the PCA pump, where floor nurses are responsible for checking the pump settings, monitoring the history of the patient's PCA therapy, and occasionally reprogramming the pump settings. Though floor nurses may be certified in PCA operation, their proficiency levels are likely different from that of recovery room nurses because of the differential frequency of use. Hence, it is important to distinguish between these two groups of users. Time constraints limited the experimental portions of this study to only one group, the more experienced programmers, recovery room nurses.

Interviews with the recovery room nurses reiterated the problems identified in the bench tests, namely incompatible defaults, complex or lengthy programming procedures,

tedious error recovery, and indistinguishable touch switches. This underscores the problems encountered during bench tests, inasmuch as the bench tests were performed under comparatively favourable conditions.

2.3.2.2 Human Factors Design Principles

In this section, a discussion is presented of the human factors design principles which were adopted to analyze and subsequently guide the redesign of the Lifecare PCA interface. Much of the discussion centres around existing human-computer interaction principles (Cook, Woods, & Yue, 1992; Molich & Nielson, 1990; Wickens, 1992), namely:

1) Provide users with prompt, salient feedback after each action

2) Make the function of the various controls clear and obvious

3) Make displayed messages clearly understandable

4) Minimize the load on the user's memory

5) Provide users with shortcuts to increase efficiency

6) Provide clearly marked exits for the user to leave the system

The following discussion briefly explains each of the principles and illustrates how they apply to the current interface. Readers should refer to Isla & Lin (1993) for a more detailed discussion.

Provide Feedback. Feedback serves to relate the status of the device to the user. This includes commands entered by the user, as well as results of those commands (Cook, Woods, & Yue, 1992). Feedback is especially critical in both detecting and recovering from errors.

A limited amount of feedback is provided by the Lifecare PCA interface via the LCD. For instance, the machine displays a message to indicate when an infusion is in progress. However, during the programming sequence, the machine provides no feedback on which parameters have already been set, or how many are left to program. This can be especially frustrating for a new user of the device; lengthy programming procedures may perpetuate the feeling of lack of control over the system.

Make Functions of Controls Clear. Multifunctional controls may be a source of memory load and confusion. The user must remember the various operations of multifunctional controls and the contexts in which they can be used, resulting in increased mental workload and risk of errors. There are several examples of multifunctional touch switches on the current interface, one of which is the BOLUS DOSE touch switch, which has two functions: to initiate bolus dose setup and to start the infusion of a bolus dose.

Various touch switches which have labels similar in meaning may also be a source of confusion for the user who relies on surface control. For instance, a user must rely on his/her deep control of the system to know the difference between the ON and RESET/START touch switches. The ON touch switch is used to activate a warm- or cold-start of the device (retains or clears previous parameter settings respectively, depending on how long the touch switch is depressed and whether the machine has been off for more than an hour). Meanwhile, the RESET/START is used to start or stop infusion.

Inconsistency in function is the last aspect of control functions which was analyzed. The REVIEW/CHANGE touch switch is one example of inconsistent function. Though it appears that the touch switch performs a single function, which is to scroll back to the previous screen, it was found that there are inconsistencies in this function across various system states. In some system states, pressing REVIEW/CHANGE will scroll two screens back rather than one.

Make Displayed Messages Clearly Understandable. Displayed messages are used to convey information to the user and are useful only when they are valid and clearly understood. Messages which are unclear may themselves be sources of error (Cook, Woods, & Yue, 1992). Several LCD messages on the current interface are awkwardly worded, and ambiguous in some cases. For instance, the following awkwardly worded message appears when the REVIEW/ CHANGE touch switch is pressed at the end of the programming sequence:

CHANGE?
ANY SETTINGS:/
SELECT MODE
YES OR NO

This seemingly convoluted message is intended to query the user as to whether or not he/she would like to change the mode or any settings, yet the language used may not be intuitively understood. Another example is the message which asks the user:

4 HOUR DOSE LIMIT
SET?
YES OR NO

This query is asking the user whether he/she wishes to set the 4 hour limit. However, the question appears to be in past tense. If the user interprets the question this way (i.e., has the 4 hour dose limit already been set?) then answering NO will inadvertently end the programming sequence without allowing the user to program the dose limit. To recover from this error, the user must press the REVIEW/CHANGE touch switch and advance through the entire sequence until he/she is prompted again to set a 4 hour dose limit. As discovered from interviews with nurses at another major hospital in Toronto (where the Lifecare PCA pump is also used), confusion over this particular awkwardly worded message resulted in repeated omittance of programming the prescribed 4 hour dose limit, until two years later when the correct meaning of the messages was pointed out to the nurses.

Minimize User Memory Load. The interface should minimize the extent to which the user must store information in short term memory to operate the device. To relax the dependency on short term memory, perceptual cues should be provided. A device is said to have visibility if by looking at it, one can immediately tell what state it is in and what the alternatives for action are (Wickens, 1992).

The current interface's sequence of programming is one such example of relieving the user's memory load. Since the programming procedure leads the user through the necessary screens for entering the required settings, it is not necessary for the user to remember the list of parameters which must be programmed for each of the three modes of operation. Instead, the relevant screens are automatically brought up based on the user's choice of mode.

Provide Shortcuts. Shortcuts should be provided to eliminate unnecessary steps in a task (Molich & Nielson, 1990), and to allow the experienced user to progress through the task much more quickly.

One method of providing shortcuts is to set appropriate machine defaults for each of the parameter settings. With the current interface, many of the default values conflict with standard operating values used at The Toronto Hospital. For instance, the standard drug concentration used at The Toronto Hospital is 2.0 mg/ml. To program the drug concentration at that value, however, requires the user to advance through four screens of machine defaults before reaching the screen which allows manual entry of the setting. Suitable machine defaults would eliminate these unnecessary programming steps.

Provide Clearly Marked Exits. Providing exits which permit the user to leave a system or subsystem without affecting any settings can prevent errors from occurring (Molich & Nielsen, 1990).

Several subtasks in the programming procedure with the current interface do not provide adequate exits. One such example is the aforementioned procedure for setting drug concentration where the user must advance through a series of screens to set the drug concentration. If the user inadvertently skips the desired screen or default setting, it is not possible to return to the previous screen using the REVIEW/CHANGE touch switch. Instead, the user must advance through a loop of several more screens to start again from the beginning (see the decision/action structure for setting concentration in Figure 2.4).

In summary, the combined results of the cognitive task analysis and the analysis based on human factors design principles helped to identify aspects of the current design which necessitate improvement. Table 2.4 shows a summary of the problems identified, along with their effects, severity of the problem, and a proposed solution. The problems include lack of external cues, poor communication of information (in screen messages and touch switch labels), and complexity of the programming sequence. The conjunction of these problems lead to high cognitive demands, perpetuating the chances of error, particularly when compounded by the demanding and stressful environment in which the machines are used. This shows the need for a PCA interface which minimizes mental effort, the risk of errors, and the time required to complete programming tasks. Thus, a redesigned interface was developed with these objectives in mind.

Table 2.4 Summary of problems with existing interface, along with their effects, their severity, and proposed solution.

Problem	Effects	Severity	Solution
Dialogue structure too complex	- requires many steps - memory load	High	- present all options in parallel
Dialogue structure not visible	- memory load	High	- provide overview
Error recovery is tedious	- many steps - takes time	High	- single step backup function
Program not visible	- hard to detect errors - memory load	High	- show data already entered upon request
Current place in dialogue not visible	- memory load - disorienting	High	- show current place in context of overview
Controls poorly grouped	- takes longer to find them	Med	- provide functional grouping
Misleading & confusing labels	- hard to interpret - slows down the user	Med	- provide labels that are user-driven
Misleading & confusing messages	- hard to interpret - slows down the user	Low	- provide messages that are user-driven
Review/Change inconsistent	- disorienting & confusing	Low	- make consistent
Defaults not appropriate	- many steps - takes time	Low	- use hospital standards

2.3.3 Redesigned Interface

The collective results of the analysis of the current PCA interface provided the necessary evidence to motivate changes to the Lifecare PCA machine's interface. The redesigned interface is described in the two sections that follow. It is noteworthy that the technological requirements of the proposed design closely mirror those of the current PCA interface, as will be evident from the discussions to follow. Constraining the physical format of the design was compelled by our goal of maintaining manufacturing costs roughly similar to those of the current PCA machine.

2.3.3.1 Interface Description

The new interface is illustrated in Figure 2.5. It consists of an enlarged LCD which contains a message field, a menu display, and an indicator field. The message field displays the instructional messages while the indicator field displays the programming task being performed (e.g., purging, setting PCA dose, setting continuous dose, etc.). The menu display, consisting of a menu of the programming stages, tracks the status of the programming task; as

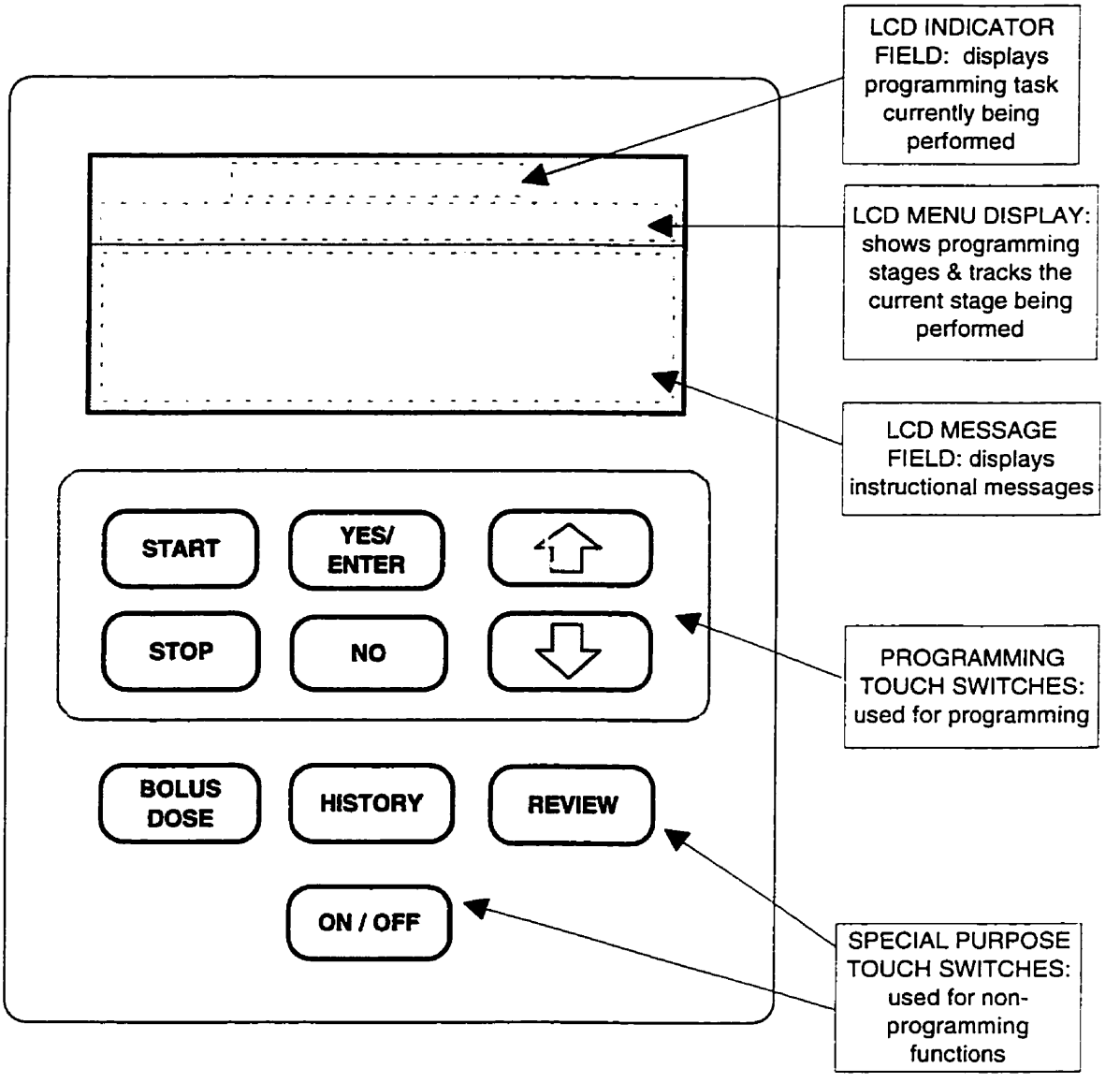


Figure 2.5. Redesigned PCA interface.

the program leads the user through each stage of programming (concentration → mode → settings), the corresponding item on the menu display is highlighted by an indicator box. Figure 2.6 shows a sample screen that illustrates the message screen for the mode selection subtask. The menu display also allows for a global view of the programming sequence which was absent from the existing Lifecare PCA interface. Moreover, awareness is promoted by sequence indicators on the menu display which show users how far they have progressed along the programming sequence, and how many more stages of programming remain to be finished.

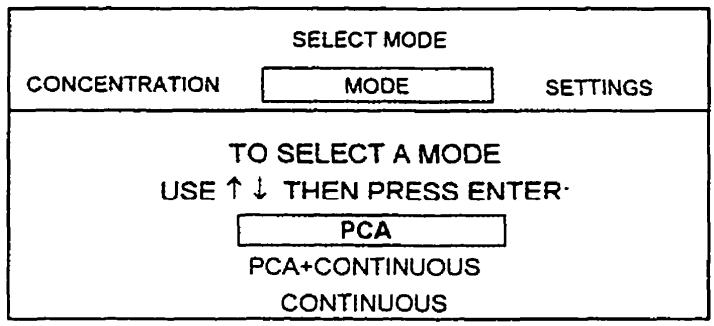


Figure 2.6 Sample message screen on LCD of redesigned interface.

In Table 2.5, touch switches for the new interface are listed along with their revised functions. The touch switches of the current Abbott PCA interface were modified in several ways. First, the number of touch switches was reduced by eliminating the PURGE SYSTEM touch switch. The dialogue of the purging task now prompts the user to press START to begin purging, and to press STOP to end it. Second, the double touch switches on the Lifecare interface were separated into single touch switches for the new interface. Also, programming touch switches were grouped separately from the special purpose touch switches (e.g., REVIEW, HISTORY, ON/OFF, and BOLUS DOSE). The programming touch switches were further organized into corresponding pairs: YES and NO, START and STOP, and the up (↑) and down (↓) arrow touch switches. As the labels imply, each of these physical touch switch represents only one soft touch switch.

Table 2.5. Redesigned interface's touch switches and their functions

Touch Switch	Function
START	Begins purging. Starts continuous infusion when programmed for CONTINUOUS mode. Starts infusion of a bolus dose.
STOP	Stops purging after slack is cleared. Stops continuous infusion.
YES/ENTER	"YES" responds to a "yes or no" message prompt. "ENTER" selects an option or inputs the setting of an operating parameter into the system.
NO	"NO" responds to a "yes or no" message prompt.
REVIEW	During programming mode, this touch switch allows the user to scroll back to the previous screen in order to correct any errors in the parameter settings After programming is completed, this touch switch allows the user to make changes to parameter settings by scrolling forward through the programming sequence.
↑ ↓	Increases or decreases a numerical value displayed on the LCD.
BOLUS DOSE	Initiates the bolus dose programming sequence.
HISTORY	After programming is completed, this touch switch allows the user to review the parameter settings entered. During therapy, the touch switch can be pressed to display the parameter settings as well as the history of all doses delivered.
ON/OFF	Activates a cold start if the machine has been off for longer than 60 minutes. Turns the machine off, retaining settings and history for 60 minutes.

The functions of the touch switches on the new interface differ slightly from those on the current interface, due to modifications to the programming structure. The BOLUS DOSE on the new design, for instance, replaces the LOADING DOSE touch switch. The new label more accurately describes its function, as indicated by the nurses interviewed. Additionally, the prompt, "administer loading dose now? yes or no" has been eliminated from the programming sequence, since a loading dose (bolus dose) can be administered at any time during and after programming (provided that the concentration has been set). Interviews with nurses also indicated that a bolus dose is rarely administered during programming. In designating the BOLUS DOSE touch switch as special purpose, the programming sequence was simplified, the number of programming touch switches was reduced, and the touch switch's multifunctionality was replaced by a single function.

Other modifications to touch switch functions are discussed in more detail in Lin et al. (1995). To summarize, modification of the touch switches and their functions were intended to produce more efficient communication of information on the redesigned interface. By confining user response to a small set of clear alternatives (yes/no, start/stop, up/down), it was expected that decision making would be quicker and simpler. Furthermore, the organization of the touch switches into two distinct groupings was also expected to result in quicker response execution.

2.3.3.2 System Structure

The function flow diagram for the new interface design is shown in Figure 2.7. Note that at this level, the flow of subtasks is similar to that for the current interface, differing only in the fact that the loading dose subtask was removed from the programming sequence.

To achieve the objective of reducing programming time, the structure of the programming sequence was simplified by employing two strategies: using a menu system for the subtasks which involved choices, and using appropriate defaults for each of the operating parameters. Menu systems were implemented in message screens in order to provide the user with a complete list of choices for the programming tasks that involve decision-making. An option from a list can be chosen by moving an indicator box up and down using the↑/↓ touch switches to highlight that option. For instance, the subtask of selecting mode would display the message shown in Figure 2.6 (in section 2.3.3.1).

Using local menu systems achieves two goals: it decreases the number of screens the user must scroll through, and it provides the user with a more global view of all of the decision options on a single screen. Incorporating these local menu systems for the subtasks that required choices (i.e., selection of concentration units and selection of mode) effectively reduced much of the complexity in the programming sequence. The revised decision/action structure for the new interface's programming sequence is illustrated in Figure 2.8 (compare with Figure 2.4, the decision/action structure for the old interface).

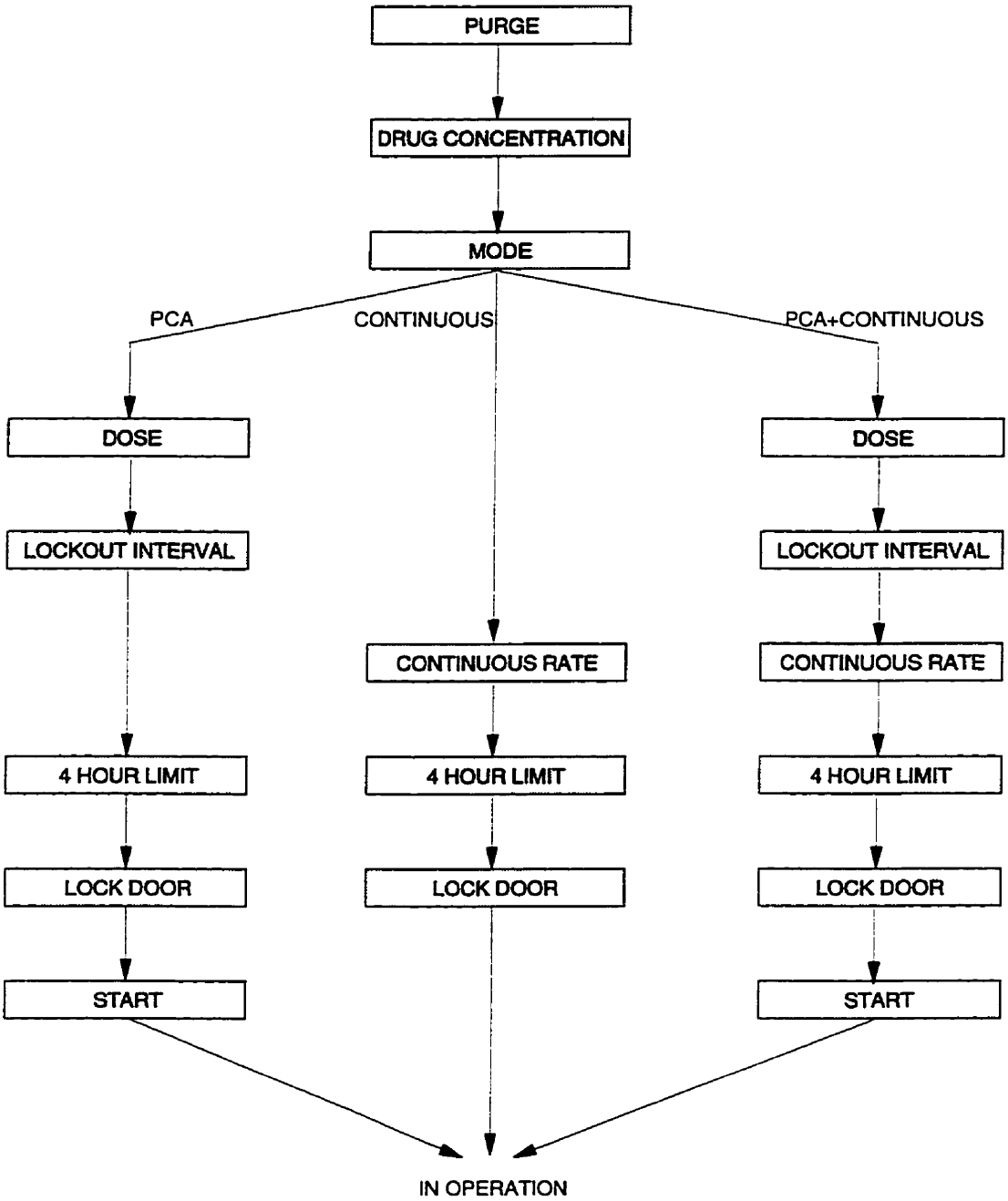


Figure 2.7. Function Flow Diagram for redesigned PCA interface

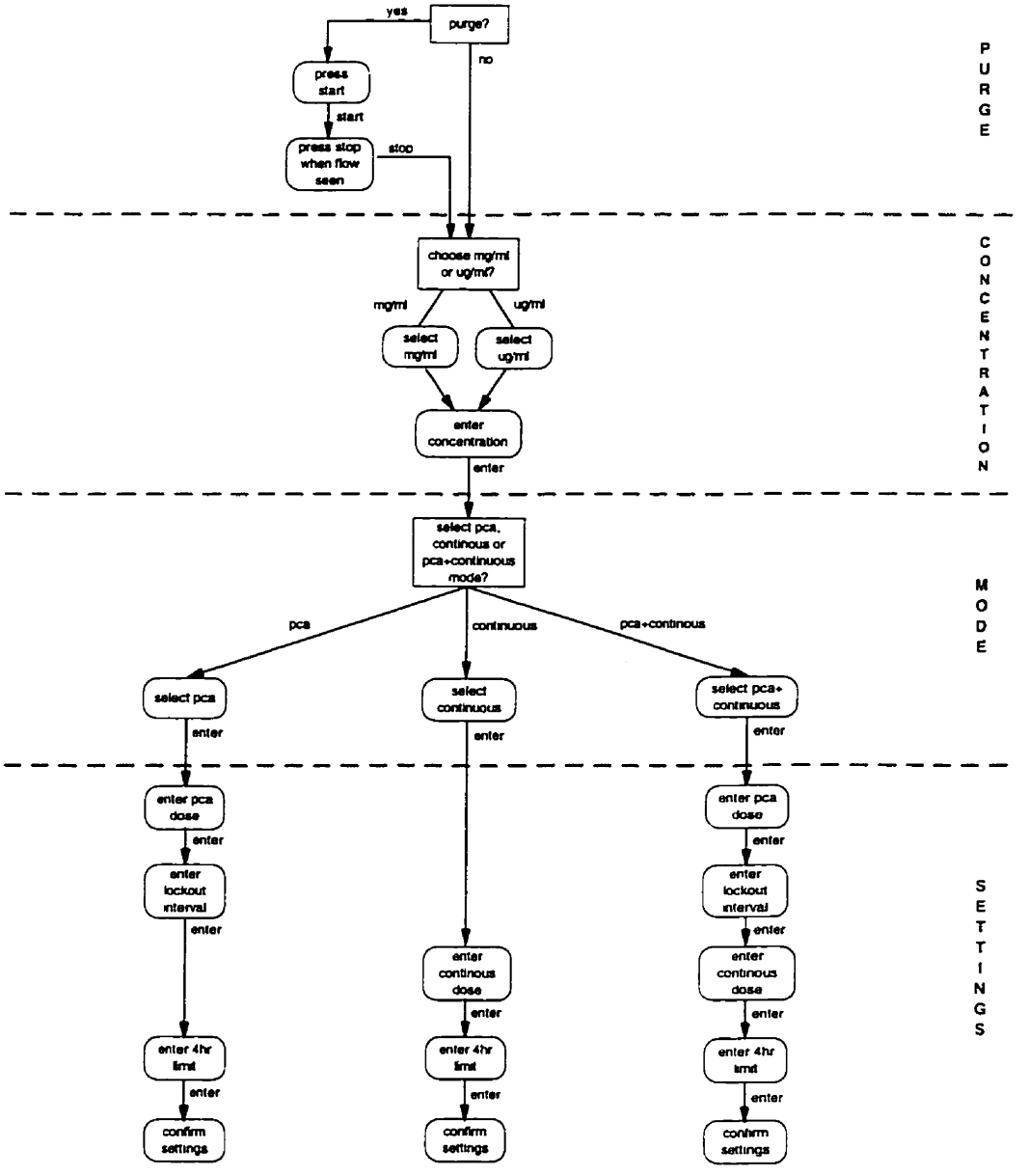


Figure 2.8. Decision/action structure of programming sequence for the redesigned interface. The column at the right shows the programming stages.

The proper machine defaults on the new interface were also expected to contribute to a further reduction in the time required to complete programming tasks. The example above (mode selection subtask) also illustrates the use of proper defaults: the most commonly used mode (PCA mode) is presented at the top of the list and is highlighted as the default.

The final modification to the programming sequence was the removal of message screens for extraneous tasks (e.g., bolus dose), and the simplification of procedures that are fixed in the protocol of machine setup (e.g., setting 4 hr limit). For instance, the screen showing the message "4 hr dose limit set? yes or no" was eliminated from the programming sequence. Instead, the user is led directly to the screen which allows him/her to enter the setting (standard protocol at The Toronto Hospital requires that the 4 hour limit must always be set).

The combined effect of these modifications to the programming sequence is a significant simplification in the programming structure. The improvement in programming structure complexity is evident by comparing the juxtaposed thumbnail images of the decision/action structures for the two interfaces in Figure 2.9. A reduction in the number of decisions is evident in the new design's structure, replaced by more message-guided actions; the new interface's programming sequence involves 3 decision-making activities and 22 message-guided actions, while the old interface requires 15 decisions and 16 message-guided actions.

Furthermore, a comparison of the number of screens involved in the programming sequence for the two interfaces demonstrates the new design to be more efficient. The current Lifecare PCA has a minimum of 8 screens and a maximum of 27, whereas the new design has the same minimum but a maximum of only 12 screens, a reduction of 56%. It is expected that a reduction in programming time should result from the new design, not only due to the reduction in number of programming screens, but also because of the added visibility of the system provided by the global and local menu systems (menu display and menu of decision options, respectively).

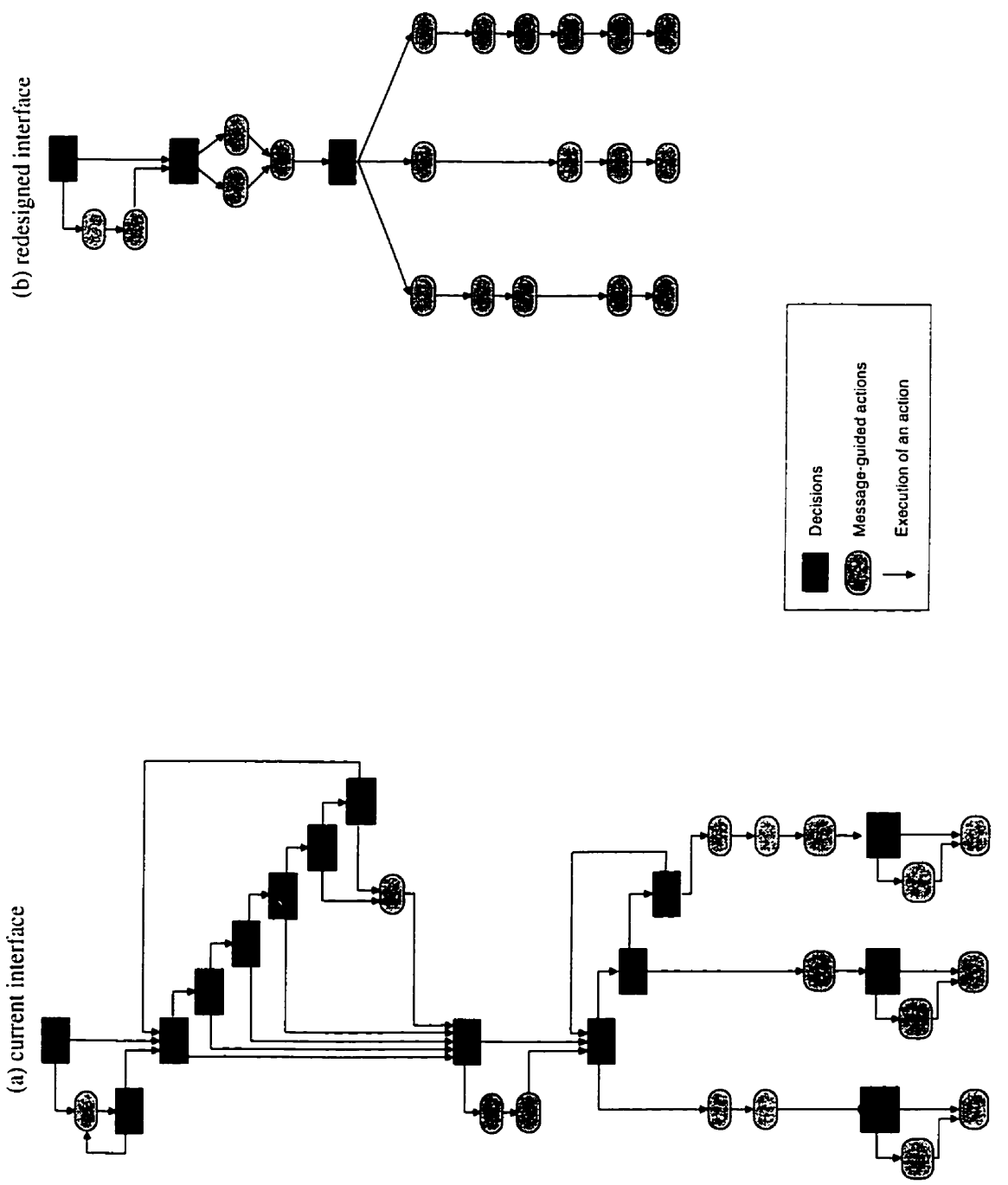


Figure 2.9. Comparison of decision/action structures. (a) Decision/action structure for the current interface. (b) Decision/action structure for the redesigned interface.

To test the hypothesis that the redesigned interface would in fact facilitate programming, this thesis endeavored to empirically evaluate the redesigned interface against the existing one. Two experiments were conducted in which programming time, subjective mental workload, and programming errors were measured. Each experiment involved a different population of users: (i) novice users with no prior experience with PCA devices, and (ii) highly experienced PCA programmers. The experiments are described in the two subsequent sections.

3. EXPERIMENT I: COMPARATIVE EVALUATION WITH NOVICE USERS

3.1 Hypothesis

This experimental evaluation tested the hypothesis that the redesigned interface facilitates the PCA programming task for novice users who have no previous experience with PCA devices. The improvement is expected to be exhibited by faster programming times, reduced mental workload, and fewer programming errors.

3.2 Methods

This section describes the methods employed in the experimental evaluation.

3.2.1 Subjects

The selection of subjects was based upon several criteria. First, it was necessary to select subjects with a similar educational background as that of the current users of PCA devices, hospital nurses. Second, the prospective subjects were to have no previous training on the operation of PCA devices. Third, in order to minimize bias, subjects were limited to those who had minimal exposure to any commercial model of PCA devices. To meet all three of these selection requirements, nursing students at the University of Toronto were solicited to participate in the experiment. A total of 12 subjects, between their 2nd and 4th years of the nursing program, were chosen to participate.

3.2.2 Materials

The experiment was conducted with computer simulations of the current and redesigned interfaces (hereafter, referred to as old and new interfaces respectively). The

hardware devices, software applications, and other materials used in the experiment are described in this section.

Hardware

An IBM compatible 486DX 33 MHz PC was used to run the simulations of the interfaces (described below). The display and input devices used in the experiment include a 14 inch SVGA colour monitor and standard mouse.

Software

Toolbook (version 1.5) was used to run the software simulations in the Windows 3.1 environment. The simulations of the interfaces were developed for this study using Toolbook's programming language (Openscript), and built upon earlier versions of the simulations developed by Isla & Lin (1993) and Doniz & Harkness (1994). Figures 3.1 and 3.2 depict screen captures of the simulated interfaces. Some of the major features of the simulations include:

(1) Integrity of visual appearance. The visual features of the simulations are similar in size, colour, and font to the actual PCA device.

(2) Integrity of function. Simulated touch switches used for programming are fully functional. However, the touch switches are mouse driven, and not driven by touch screen.

(3) Fidelity of internal system structure. The full dialogue structure of the programming sequence was incorporated in the simulations, allowing the user to program the device for all three modes of operation (P, C, and P+C modes).

(4) Electronic data logging. The simulations were coded so that data files would automatically be created to record the time and nature of user control actions (i.e., touch switch activation).

PCA Order Form

The Toronto Hospital's standard PCA order form, shown in Figure 3.3, was used in the experiment to provide the prescribed PCA parameter settings to the subjects.

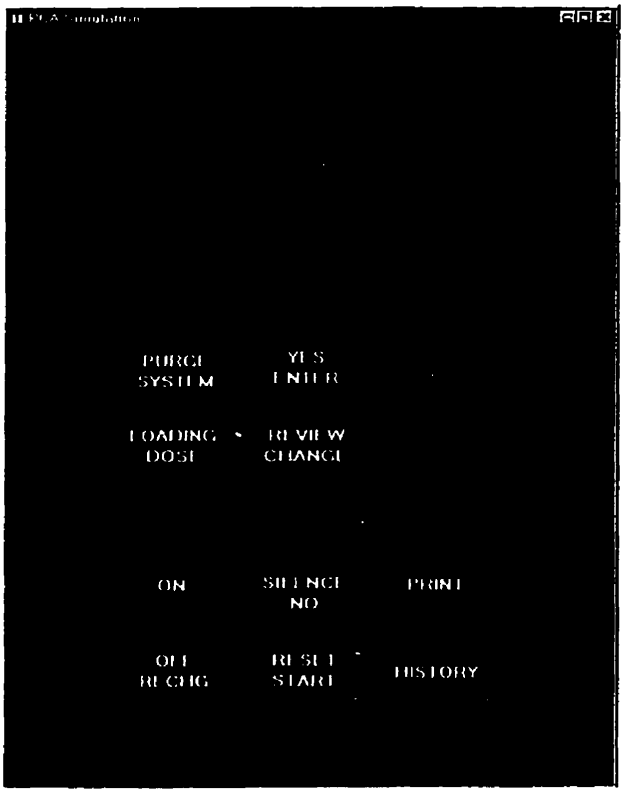


Figure 3.1 Screen capture of old interface simulation.

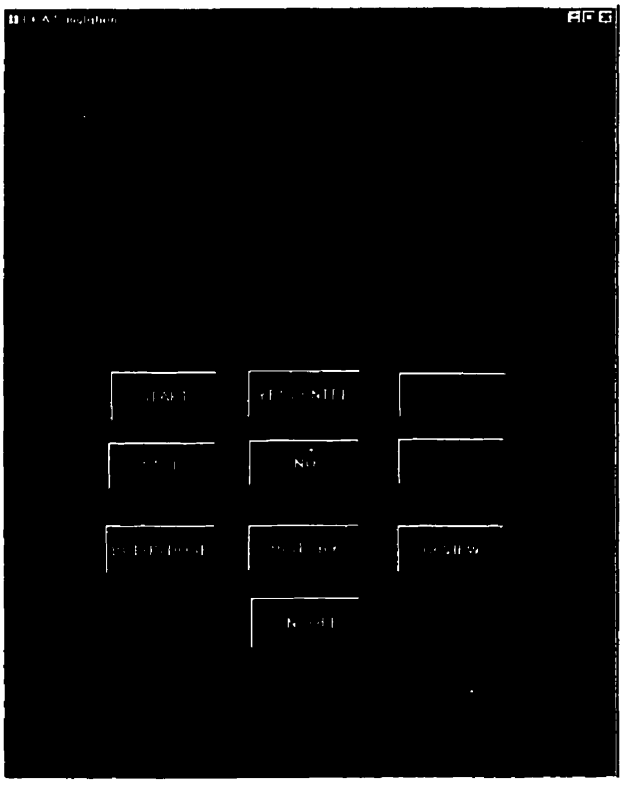


Figure 3.2 Screen capture of new interface simulation.

Figure 3.3. The Toronto Hospital PCA order form.

THE TORONTO HOSPITAL

Doctor's Order Sheet

PATIENT CONTROLLED ANALGESIA (PCA) STANDING ORDERS

ALLERGIES

NO KNOWN ALLERGIES ☐

DATE AND TIME ORDERED	PHYSICIAN'S ORDER	ACTION TAKEN	SIGNATURE AND POSITION

1. While on PCA device, patient is to receive No further supplemental narcotics unless approved by the Anaesthesia/Acute Pain Service.

2. PCA Drug: Morphine _____ mg/ml
   Meperidine _____ mg/ml
   Other _____ mg/ml

3. Pump Settings:
   Dose _____ mg. to _____ mg.
   Initial Lockout Interval _____ min.
   Continuous dose _____ mg/hour.
   Start continuous dose when pt returns to floor post-op.
   Four hour limit _____ mg.

4. Monitoring: Respiratory rate, sedation score q2h x 24 hrs, then q4h. Record on PCA Flow sheet.

5. Treatment of Side Effects:
   Have Narcan (Naloxone) 0.4 mg/ml vial readily available at nursing station.
   Nausea/vomiting: Gravol _____ mg IV/IM q3-4h prn.
   IV dose to be infused in 15 - 30 min.
   Itchiness: Benadryl _____ mg IV/IM q3-4h prn.
   IV dose to be infused in 15 - 30 min.
   Discontinue Gravol and Benadryl when PCA stopped.

6. Call Acute Pain Service Physician beeper # for:
   TGD: 9-582-1321 TWD: Day 9-329-2473 TWD: Night 9-329-2420
   a) Respiratory Rate less than 10/min.
   b) Blood Pressure less than 80 systolic.
   c) Pulse less than 50/min.
   d) Sedation Score of 3 (somnolent, difficult to rouse).
   e) Unsatisfactory analgesia.
   f) If 4 hour limit of drug is reached before 4 hours elapse.

7. In an emergency, if NO response after calling the above Beeper #, call Anaesthesia Resident through locating.

8. If side effects of slow respiratory rate, hypotension or somnolence occur, STOP PCA pump immediately and inform attending service as well as Acute Pain Service.

9. RN will check and verify PCA setting once per shift.

10. Acute Pain Service Nurse beeper number:
    TGD: 9-582-9344 TWD: 9-329-2472 (0900 - 1700 Mon - Fri)

11. Office and answering machine:
    TGD: 340-4997 TWD: 369-5118

_____ _____

Physician's Signature Date

3.2.3 Experimental Design

A 2x3x2x2 mixed design was used, with *interface* (old and new), *programming mode* (P, C, and P+C), and *repetition* (1st and 2nd repetition of each programming task[3]) as within subject factors, and *order of training* (old first and new first) as a between subjects factor. Each subject completed 6 programming tasks with each interface. The order of training and order in which subjects received the programming tasks were each counterbalanced.

3.2.4 Procedure

Each subject participated in two sessions of the experiment, scheduled on two different days to avoid effects from fatigue. In the first of the two sessions, subjects were briefed on the purpose of the experiment and the tasks they would be performing. Twenty minutes of training time was allotted for individual instruction and hands-on practice with the interface to which they had been randomly assigned to be trained on first. Then, for each of the six trials, subjects were provided with a PCA order form, and proceeded to program the prescribed PCA parameter settings using the given interface. After each trial, mental workload ratings were obtained from subjects using NASA-TLX rating scales (Hart & Staveland, 1988).shown in Figure 3.4(a). Pairwise comparisons of the workload factors were performed by subjects at the conclusion of the sixth trial in order to obtain weighting vectors that represented each individual's concept of workload (see Figure 3.4(b)).

In the second session, the alternate interface was given to subjects, and the equivalent training was provided as with the initial interface. Subjects then completed another 6 trials with this alternate interface following the procedure described above for session one.

At the conclusion of the second session, subjects were interviewed in order to obtain comments and their subjective opinion regarding which interface they preferred.

---

[3] task and mode are used interchangeably throughout the thesis, since each programming task consists of programming the PCA interface in one of the three modes, P, C, or P+C.

(a)

NASA-TLX Workload Rating

NAME: DATE: TRIAL #:

Instructions: Place a mark along each scale that represents the magnitude of each factor in the task you just performed

MENTAL DEMAND

Low — High

How much mental and perceptual activity was required (e.g. thinking, deciding, calculating, remembering, looking, searching, etc.)? Was the task easy, simple and forgiving (low mental demand) or demanding, complex and exacting (high mental demand)?

PHYSICAL DEMAND

Low — High

How much physical activity was required (e.g. pushing, pulling, turning, controlling, activating, etc.)? Was the task easy, slow, and slack (low physical demand) or demanding, brisk, and strenuous (high physical demand)?

TEMPORAL DEMAND

Low — High

How much time pressure did you feel due to the rate or pace at which the tasks or task elements occurred? Was the pace slow and leisurely (low temporal demand) or rapid and frantic (high temporal demand)?

PERFORMANCE

Perfect — Failure

How successful do you think you were in accomplishing the goals of the task set by the experimenter (or yourself)? How satisfied were you with your performance in accomplishing these goals?

EFFORT

Low — High

How hard did you have to work (mentally and physically) to accomplish your level of performance?

FRUSTRATION LEVEL

Low — High

How did you feel during the task? Content, gratified, relaxed and complacent (low frustration level) or insecure, discouraged, irritated, stressed and annoyed (high frustration level)?

(b)

NASA-TLX Pair-wise Comparisons of Factors

NAME: DATE: INTERFACE:

1) Physical Demand / Mental Demand
2) Temporal Demand / Mental Demand
3) Performance / Mental Demand
4) Frustration Level / Mental Demand
5) Effort / Mental Demand
6) Temporal Demand / Physical Demand
7) Performance / Physical Demand
8) Frustration Level / Physical Demand
9) Effort / Physical Demand
10) Temporal Demand / Performance
11) Temporal Demand / Frustration Level
12) Temporal Demand / Effort
13) Performance / Frustration Level
14) Performance / Effort
15) Effort / Frustration Level

Figure 3.4. NASA-TLX Mental Workload Rating adopted from Hart & Staveland (1988). (a) Ratings scales. (b) Pairwise comparison

3.2.5 Performance Measures

The three performance measures were performance time, mental workload, and number of programming errors. Performance time and programming errors were evaluated from subject data files obtained from the simulations, while mental workload was obtained using NASA-TLX, a well accepted method of measuring subjective mental workload (Hart & Staveland, 1988).

Performance Time

Performance time was measured in terms of task completion time and subtask completion times. Overall task completion time is a measure of the time it takes to complete one trial of the experiment, or equivalently, the time it takes to complete programming for one PCA order form. Subtask completion time, recorded for each of the 9 subtasks within the programming sequence, is a measure of the time it takes to complete a programming subtask. Overall and subtask completion times are explain further below, along with the methods of extracting these measures from the data files.

Overall Task Completion

Figure 3.5 is a sample subject data file recorded for one trial in the experiment. The data consist of lists of actions which the subject performed during the trial, along with the time at which the actions occurred. Each action is described by two items: the touch switch that was pressed (shown in capitalized letters), and a brief description of the message prompt to which the touch switch was activated in response. In some instances, when there is no user action associated with an event (such as the completion of a loading dose), the data file shows only the description of the event and the time it occurred.

From these data files, the overall task completion time was measured from the "beginning of trial" to the "end of trial". The "beginning of trial" correlates with the time at which the simulated device was turned on by the subject, while the "end of trial" is the time at which the subject completed all the required programming stages.

OLD PCA INTERFACE: TRIAL 2	
ON: beginning of trial	10:28:28 AM
YES: purge system?	10:28:35 AM
PURGE: begin purging	10:28:39 AM
YES: flow seen?	10:28:51 AM
NO: 1MG/ML?	10:28:54 AM
NO: 5MG/ML?	10:28:55 AM
NO: 0.5MG/ML?	10:28:56 AM
NO: 10MG/ML Meperidine?	10:28:57 AM
YES: MG/ML?	10:28:59 AM
YES: entered conc amt (MG/ML): 2.5	10:29:07 AM
YES: load dose?	10:29:11 AM
YES: entered bolus amt: 1.5	10:29:20 AM
LOAD: begin infusion 1.5	10:29:23 AM
Loading dose completed	10:29:44 AM
NO: pca mode?	10:29:48 AM
NO: continuous mode?	10:29:49 AM
YES: selected P+C	10:29:51 AM
YES: entered pca dose: 1.5	10:30:04 AM
YES: entered lockout interval: 6	10:30:09 AM
YES: entered contin rate: 1.0	10:30:13 AM
YES: set 4hr limit?	10:30:17 AM
YES: entered 4hr limit: 25.0	10:30:21 AM
ENTER: begin therapy	10:30:26 AM
END OF TRIAL	10:30:26 AM

Figure 3.5 Sample data file from experiment.

Subtask Completion Time

Subtask completion times were also determined from the data files. Performance differentials between the interfaces were examined at the subtask level in order to localize the subtasks where performance with the new interface improved, degraded, or was unchanged compared with the old interface.

The programming task consists of up to nine subtasks: (1) purging the system, (2) selecting concentration units, (3) setting the concentration value, (4) administering a bolus dose, (5) selecting a mode, (6) setting the PCA dose, (7) setting the lockout interval, (8) setting the continuous rate, and (9) setting the 4 hour limit. To calculate each subtask's completion time, the 'start' of a subtask was defined as the time at which the first of its

message prompts appears, and the 'end' of a subtask as the time at which its final message prompt disappears. The time between successive message prompts was assumed to be negligible, as was the time it took for the computer to process and respond to the user's action (each of these processes took less than 1 second). Thus, the start of a subtask corresponds with the end of the preceding subtask. Accordingly, subtask completion times were calculated based on the time of the final action for a particular subtask and the time of the final action for the preceding subtask.

Mental Workload

Mental workload for each trial was tabulated using the data collected from subjects through rating scales (Figure 3.4(a)) and pairwise comparisons (Figure 3.4(b)). The final measure of mental workload was determined following the procedure outlined by Hart and Staveland (1988).

Programming Errors

For any set of prescribed PCA parameter values, there is only one unambiguously correct and efficient set of programming actions. User programming errors were identified as any action which deviated from this correct set of actions. By comparing the programming actions performed by each subject to the set of required actions for the given PCA order form in each trial, errors were identified and compiled into a list. The total number of programming errors was then tallied for each interface.

Errors were then classified by the severity of consequences, and according to the subtasks in which they occurred. The two classification schemes are elaborated upon below.

Classification Scheme for Error Severity

Programming errors made during each trial were classified based on the potential consequences they would have had on a patient in order to illustrate the impact of interface design problems.

Programming errors which have direct consequences on a patient's health are classified as critical errors. This class of errors includes: selecting the wrong concentration units, setting an incorrect value for drug concentration, selecting the wrong mode, and setting an incorrect value for PCA dose, continuous rate, lockout interval or 4 hour limit. Any one of

these errors would result in the patient receiving an incorrect amount of medication, and is therefore classified as a critical error.

In contrast, if any one of the above mentioned errors occurs and is corrected before the programming procedure is completed, then it is classified as a non-critical error. Non-critical errors also include actions such as responding to a message prompt with the wrong touch switch. Though critical errors provide indications of a device's safeness, non-critical errors may be diagnostically useful. Analyzing errors within the context of interface design may aid in identifying aspects of the interface which are incongruent with the device's internal system structure, the user's understanding of the device, or both.

Subtask Categorization of Errors

Errors can be further categorized based on the subtask in which the programming errors were made. The purpose of subtask categorization is to identify the subtasks which are more prone to errors. Also, a comparison may be made of each of the interfaces' subtask error rates to provide an evaluation of each interface on a more localized level (potentially a diagnostically useful tool for further design improvements).

Errors were categorized into one of nine subtask categories: purging, selecting concentration units, setting the concentration value, setting the bolus dose, selecting the mode, setting the PCA dose, setting the lockout interval, setting the continuous rate, and setting the 4 hour limit.

3.3 Results and Discussion

3.3.1 Performance Time

Interface Effects

For 9 out of 12 subjects, the average task completion time was faster with the new interface than with the old. Subjects completed programming tasks with the new interface in an average of 1.76 minutes (standard deviation of 0.48), which is 14% faster than with the old interface, 2.03 minutes (standard deviation of 0.49). Programming times for each interface are illustrated in Figure 3.6(a). An analysis of variance (ANOVA) showed that this difference in programming time is statistically significant ($F(1,10)=7.23$, $p=0.0228$).

Table 3.1. Summary of statistical results for overall task completion time, mental workload, and programming errors. Statistical significance is denoted by *.

Performance Measure	Factor	Test Statistic	Significance Level
Overall Task Completion Time	interface	$F(1,10) = 7.23$	p=0.0228 *
	order x interface	$F(1, 10) = 9.76$	p=0.0108 *
	repetition	$F(1, 10) = 7.54$	p=0.0206 *
	mode	$F(2, 22) = 7.88$	p=0.0030 *
Workload	interface	$F(1,10) = 2.82$	p=0.1238
		$\chi^2(1) = 4.45$	p=0.025, one-tailed *
	order	$F(1,10) = 8.36$	p=0.0161 *
	repetition	$F(1,10) = 15.00$	p=0.0031 *
Errors	interface	$\chi^2(1) = 3.33$	p<0.05, one-tailed *

Practice

An improvement in overall task completion time was observed over the two repetitions of each programming task ($F(1,10)=7.54$, $p=0.0206$). The average task completion time for repetitions 1 and 2 are shown in Figure 3.6(b). The interface x repetition interaction was not significant, indicating that performance on both the old and new interfaces benefit equally from practice.

Order x Interface

An unequal transfer of training between the interfaces (order x interface effect with $F(1,10)=9.76$, $p=0.0108$) was found to influence the extent to which subjects improved their overall task completion times in the second session of the experiment. Subjects who practiced on the old interface first performed the slowest on the old interface but also performed the quickest on the new interface (Figure 3.6(c)). However, for the second group of subjects who received training on the new interface first, the transfer of training was less beneficial, as is apparent from their less dramatic improvement during their second session with the old interface.

One interpretation of this result is that the old interface appears to have a damping effect on the improvement that would normally be observed in the second session from transfer of training. A more detailed investigation of this phenomenon would be necessary in

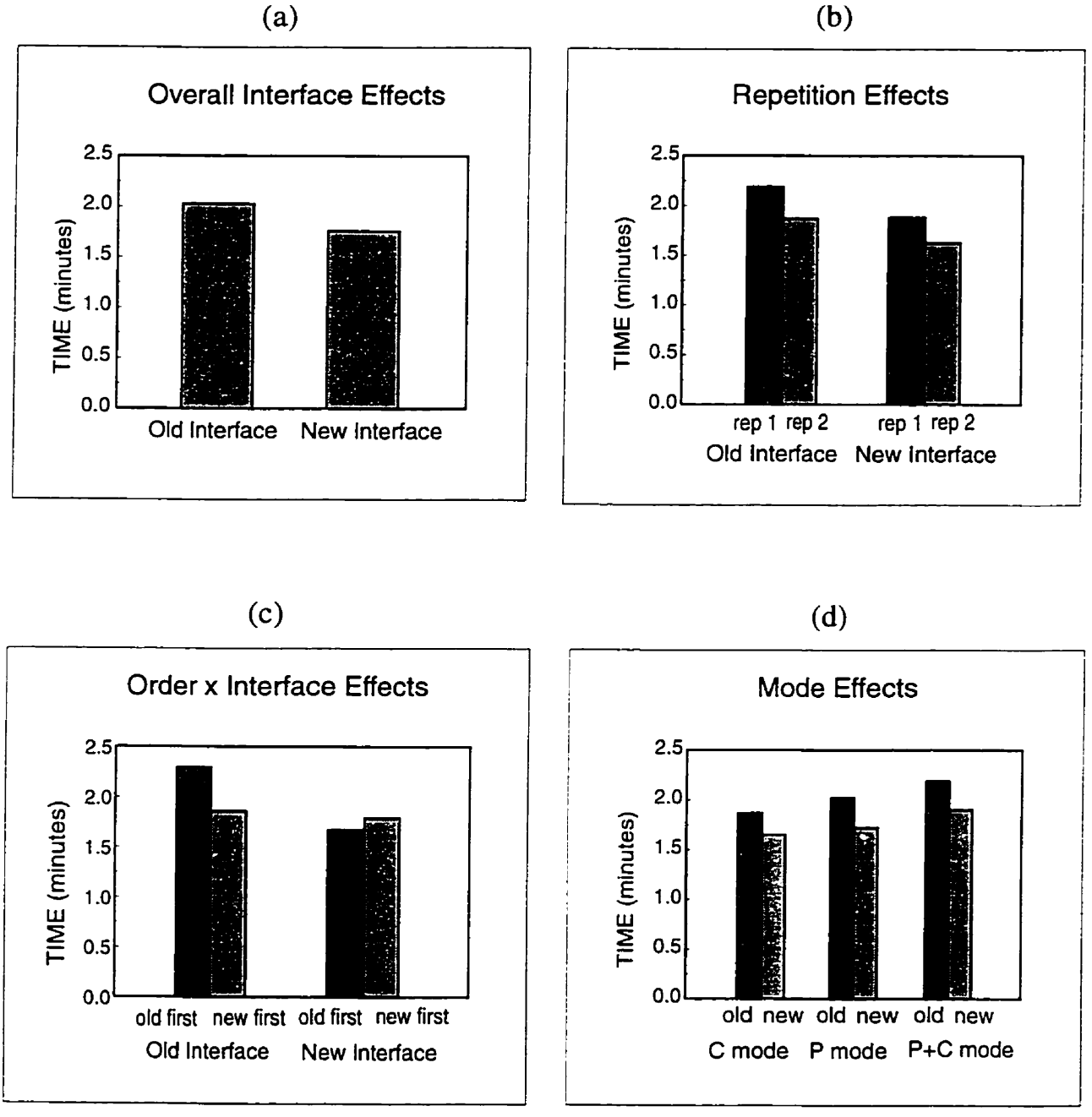


Figure 3.6. Comparison of overall task completion times (TCT) for the old and new interfaces. (a) Average TCT plotted for each interface. (b) Improvement in TCT over two repetitions. (c) Order x interface effects on TCT. (d) TCT of each of the three modes of programming.

order to ascertain the extent to which the new interface takes advantage of the benefits from training, or conversely, the extent to which the old interface hinders it.

Programming Mode

The mode of programming also had a significant effect on the time required to complete each programming task ($F(2,20)=7.88$, $p=0.003$). The P+C mode took the longest to program, while the C mode was the fastest to program. However, the interface x mode interaction was not significant, indicating that the advantage of the new interface is independent of programming mode (see Figure 3.6(d)).

Subtask Completion Times

Another ANOVA was conducted to test the effects of the experiment's independent variables, as well as the effect of subtask, on subtask completion times. The results, recorded in Table 3.2(a), show that interface and an order x interface interaction had significant effects on subtask completion times. There were also significant differences between the subtasks, as well as significant effects from a subtask x interface interaction.

In order to identify which of the nine subtasks were significantly affected by interface effects and by the order x interface interaction, separate ANOVAs were conducted for each of the nine subtasks. The results of these ANOVAs are shown in Table 3.2(b). The results show that subtask completion times for the new interface were significantly faster than that for the old interface for all subtasks except setting the concentration value and mode selection, which were also faster but not significantly so. The marked improvement in subtask completion times of the new interface over the old can be attributed to the simplified subtask structures of the new interface's programming sequence (refer to Figure 2.9). These results suggest that the design improvements incorporated in the new design have impacted virtually the entire programming task rather than just a few isolated segments of the programming task.

The ANOVAs of subtask completion times also showed that the order x interface interaction had a significant effect on 8 out of the 9 subtasks. This interaction effect had also been observed with overall task completion times. The group of subjects who received training on the new interface first appeared to be hindered from improving in their second session when they transferred onto to the old interface. For the second group of subjects, the

new interface facilitated the transfer of training from the old interface to the new. All subtasks, with the exception of entering the concentration value, were significantly influenced by this order x interface effect (see Figure 3.7), demonstrating that nearly all subtasks played a role in facilitating or hindering the transfer of training between the interfaces.

Table 3.2(a). Summary of statistical results for subtask completion times. Statistical significance is denoted by *.

Performance Measure	Factor	Test Statistic	Significance Level
Subtask Completion Time	interface	F(1,10)=137.00	p=0.0001*
	order x interface	F(1,10)=26.67	p=0.0004*
	subtask	F(8,80)=560.99	p=0.0001*
	subtask x interface	F(8,80)=31.75	p=0.0001*

Table 3.2(b). Summary of statistical results for each of the nine subtasks. Statistical significance is denoted by *.

Performance Measure	Subtask	Factor	Test Statistic	Significance Level
Subtask Completion Time	purging	interface	F(1,10) = 5.32	p=0.0438*
		order x interface	F(1,10) = 7.92	p=0.0184*
	concentration units	interface	F(1,10) = 140.10	p=0.0001*
		order x interface	F(1,10) = 8.50	p=0.0154*
	concentration value	interface	F(1,10) = 0.71	p=0.4196
		order x interface	F(1,10) = 4.55	p=0.0587
	bolus dose	interface	F(1,10) = 70.29	p=0.0001*
		order x interface	F(1,10) = 6.59	p=0.0280*
	mode	interface	F(1,10) = 0.56	p=0.4707
		order x interface	F(1,10) = 5.06	p=0.0482*
	pca dose	interface	F(1,10) = 41.68	p=0.0001*
		order x interface	F(1,10) = 9.78	p=0.0107*
	lockout	interface	F(1,10) = 5.19	p=0.0458*
		order x interface	F(1,10) = 7.82	p=0.0189*
	continuous rate	interface	F(1,10) = 82.55	p=0.0001*
		order x interface	F(1,10) = 9.17	p=0.0127*
	4 hr limit	interface	F(1,10) = 148.98	p=0.0001*
		order x interface	F(1,10) = 5.22	p=0.0455*

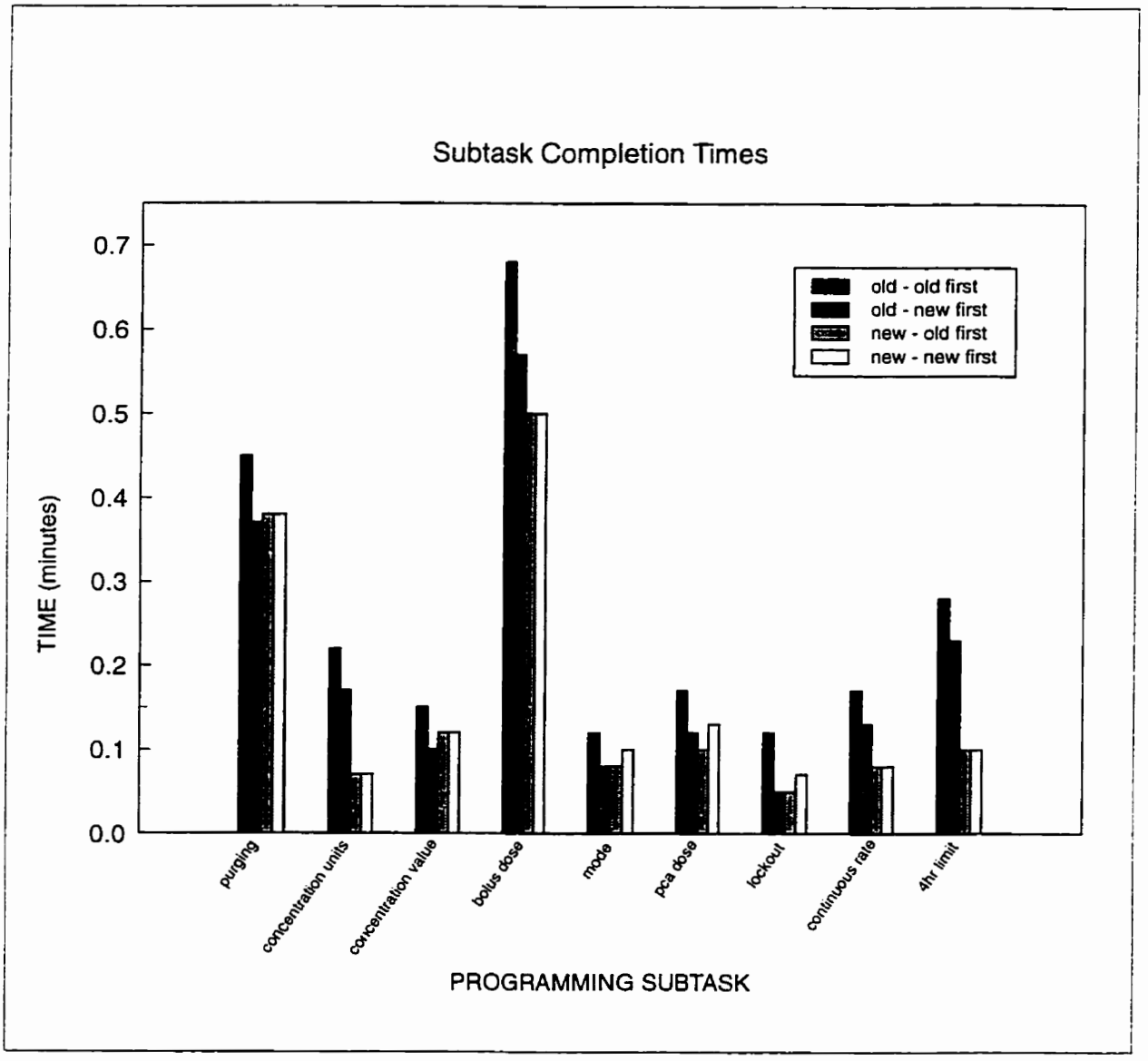


Figure 3.7. Completion times of each of the programming subtasks for novice users. Illustrates the order x interface interaction effect at the level of subtasks.

3.3.2 Mental Workload

An ANOVA of the workload ratings showed that there was no significant difference in workload between the two interfaces ($F(1,10)=2.82$, $p=0.1238$). However, as shown in Figure 3.8(a), the average workload rating for the old interface (14.5% with standard deviation of 0.18) was more than twice as high than that for the new interface (6.8% with standard deviation of 0.099). One reason that the ANOVA failed to detect a significant difference may lie in the discrepancy in absolute magnitude of the workload ratings between the two groups of subjects; the group of subjects exposed to the old interface first had significantly lower workload ratings than those exposed to the new interface first (significant order effect with $F(1,10)=8.36$, $p=0.0161$). This is likely symptomatic of floor effects whereby subjects in the former group confined their workload ratings (in the first session with the old interface) to the lower region of the workload scales, and when given the new interface in their second session, did not have adequate range remaining on the scale to express a significantly lower workload rating.

An alternative method of analysis was employed to circumvent the problem of floor effects. The number of subjects who rated a higher workload with the old interface was tabulated and a χ^2 test was used to test whether it was significantly greater than those who gave the new interface higher workload ratings. Results of this test showed that the proportion of subjects rating a higher workload for the old interface (82%) is statistically significant ($\chi^2(1)=4.45$, $p=0.025$, one-tailed).

Practice

All subjects reported a decrease in workload for the repeated tasks (trials 4-6 were repeats of the tasks in trials 1-3), which is illustrated in Figure 3.8(b). This confirms that for the novice, practice leads to a significant decrease in subjective mental workload ($F(1,10)=15.00$, $p=0.0031$), irrespective of the interface.

3.3.3 Programming Errors

There were significantly more errors made with the old interface than with the new interface ($\chi^2(1) = 3.33$, $p<0.05$ one-tailed) as shown in Figure 3.9(a). Out of a total of 30 programming errors, committed over 144 trials, 20 errors occurred while subjects were

programming with the old interface, while 10 occurred with the new interface. A list of programming errors is presented in Table 3.3.

The old and new interfaces had approximately equal ratios of critical to non-critical errors (critical errors accounted for 30% of total errors for each interface), suggesting that the new interface reduces critical and non-critical errors equally.

The categorization of errors showed that a majority of errors occurred in the concentration and bolus dose subtasks (the distribution of errors across subtasks is illustrated in Figure 3.9(b)). A total of 9 errors were associated with the concentration subtask. In fact, 8 out of the 9 critical errors originated from setting the wrong concentration (while 1 resulted from administering an extra bolus dose). That a majority of the critical errors were related to setting the wrong concentration coincides with that found in the medical device reports related to human error, and underscores the potential hazards that may result from misprogramming.

Both interfaces appear to be equally prone to errors in concentration. Of the 9 concentration errors, a total of 5 were made with the old interface, while 4 concentration errors were made with the new interface. However, a majority of those made with the new interface are suspected to have been caused by misreading the order form (i.e., mistaking the bolus dose setting for the concentration) since many of the incorrect values programmed into the concentration setting coincide with the prescribed settings for other parameters on the order form (refer to Table 3.3). Improving the PCA order form to make the concentration setting more salient may be one solution for alleviating this problem. Concentration errors with the old interface, however, were primarily caused by the mistake of accepting one of the default concentration settings that the programming sequence presents to the user (see Table 3.3). This problem can be traced back to deficiencies in the old interface rather than the order form.

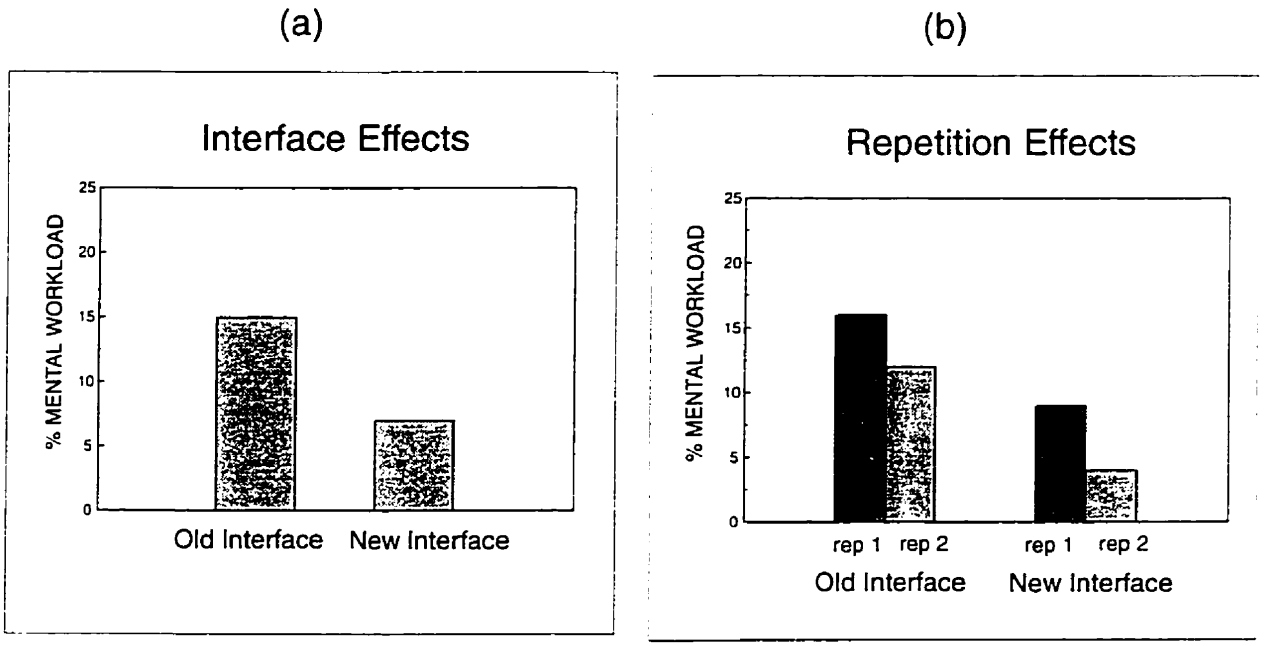


Figure 3.8. Mental workload ratings for the old and new interfaces. (a) Workload for each interface. (b) Workload reduction over two repetitions for each interface.

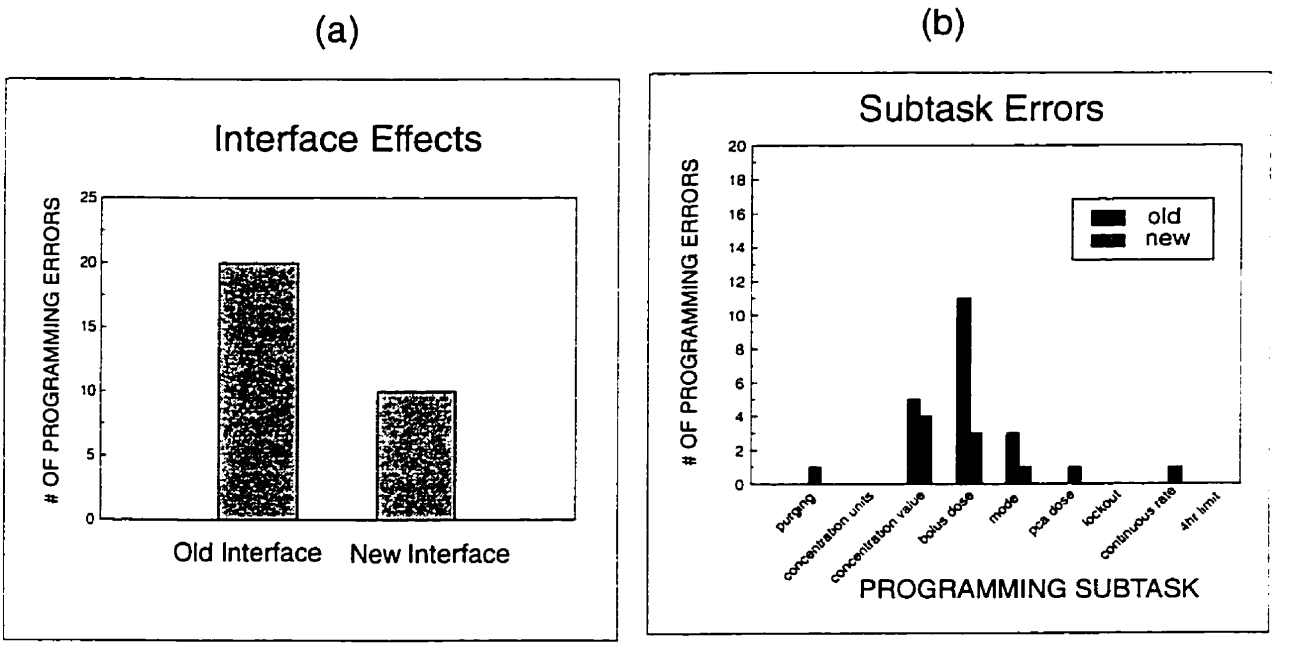


Figure 3.9. Programming Errors. (a) Total number of programming errors for each interface. (b) Distribution of errors across subtasks.

Table 3.3. List of programming errors made by novice users.

Subject	Order*	Interface	Trial	Mode	Error Category	Consequence to Patient?	Description of Error(s)	Consequences	
NL	2	old	1	p+c	**conc value**	**yes**	entered 1 5 instead of 2 5 mg/ml	each dose 4hr limit	2 5 instead of 1 5 mg / dose 41 667 instead of 25 mg
NL	2	old			**bolus (conc)**	**yes**	infused bolus (1 5 mg) calculated based on wrong conc	bolus dose	2 5 mg instead of 1 5 mg
NL	2	old	3	p	conc value	no	entered 1 mg/ml instead of 2mg/ml, later corrected		
NL	2	old	3	p	**bolus (conc)**	**yes**	then infused bolus (1 5 mg) calculated based on wrong conc, then corrected conc value	bolus dose	3mg instead of 1 5 mg infused
NH	2	old	4	c	mode	no	answered no to c mode, pressed Review/Change but ended back at Load Dose task, later corrected		
NH	2	old	2	p	bolus	no	pressed Load Dose instead of Yes/Enter to Load? prompt		
NH	2	new	3	p+c	bolus	no	pressed Start instead of Yes/Enter to set bolus amt		
MW	2	old	3	c	mode	n	selected p instead of c mode, later corrected		
MW	2	new	4	p+c	conc value	no	entered 1 5mg/ml instead of 2mg/ml, later corrected		
MW	2	new	4	p+c	**bolus (conc)**	**yes**	infused bolus (1 5 mg) calculated based on wrong conc, then corrected conc value	bolus dose	2mg instead of 1 5
LJ	2	old	1	c	mode	no	answered No to c mode causing 2nd iteration of loop		
JL	2	new	1	p	pcadose	no	entered 2 mg instead of 1 mg, later corrected		
JL	2	new	2	c	**conc value**	**yes**	entered 2 0 mg/ml instead of 1 5 mg/ml	cont rate 4hr limit	1 125 instead of 1 mg/hr 22 5 instead of 30 mg
JL	2	new	2	c	**bolus (conc)**	**yes**	then infused bolus calculated based on wrong conc	bolus dose	1 125 instead of 1 5 mg
JL	2	new	2	c	continuous rate	no	entered 1 0 instead of 1 5, later corrected		
JL	2	old	1	p	**conc value**	**yes**	entered 1 instead of 2 mg/ml	each dose 4hr limit	2 instead of 1 mg / dose 60 instead of 30 mg
JL	2	old	1	p	bolus amt	no	entered 2mg instead of 1 5 mg, later corrected		
JL	2	old	1	p	**bolus (conc)**	**yes**	infused bolus calculated based on wrong conc	bolus dose	3mg instead of 1 5 mg
JL	2	old	4	p	conc value	no	entered 1 mg/ml (Yes to default) instead of 2 5 mg/ml, later corrected		
SD	1	old	2	p+c	purge	no (if patient not hooked up)	pressed No to MG/ML and pressed Review to revert back 1 screen but ended back at Purge		
SD		old	3	c	bolus	no	pressed Load instead of Yes/Enter to set bolus amt		
MJ	1	new	6	p+c	conc value	no	entered 1 5 instead of 2 0 mg/ml, later corrected		
CM	1	new	6	p	conc value	no	entered 1 5 instead of 2 5 mg/ml, later corrected		
AT	1	old	1	p+c	bolus	no	pressed Load Dose again, did not infuse another		
AT	1	old	1	p+c	**bolus**	**yes**	Reviewed back to check conc units & conc value and infused another bolus dose	total bolus	3mg instead of 1 5 mg
AT	1	old	2	p	conc value	no	entered 1mg/ml (Yes to default) instead of 2 mg/ml later corrected		
AT	1	old	2	p	bolus	no	pressed Load Dose instead of Yes/Enter to Load? prompt		
AT	1	old	4	p+c	bolus	no	pressed Load Dose instead of Yes/Enter to Load? prompt		
AT	1	old	6	p	bolus	no	pressed Load Dose instead of Yes/Enter to Load? prompt		
AT	1	new	3	c	mode	no	selected p instead of c, later corrected		

* Order 1 = old first
2 = new first

Concentration Errors and Their Propagation

Errors in setting drug concentration propagate in the sense that other PCA parameters, even if correctly entered, are affected. For instance, if the pump's concentration setting is lower than that prescribed on the order form, the amount of analgesic that the pump administers for the continuous infusion, bolus dose and PCA doses will be more than that prescribed by the physician (overmedication). For similar reasons, the 4 hour limit will be higher, permitting more medication to be administered over a 4 hour period, thus rendering this safety parameter ineffective for preventing overmedication as it is intended to.

A concentration error that is eventually corrected at later stages of the programming sequence (a non-critical error) may still propagate, causing a critical error if it is not corrected before administering a bolus dose. If a bolus dose is infused before the correction of a concentration error, the patient receives an incorrect amount of medication than that prescribed. Four such bolus dose overmedications occurred as a result of incorrect concentration settings, three of which caused overdoses of 100%. Again, both interfaces are equally prone to these types of errors, but the occurrence of these errors may be an indication of lack of clinical training (of this particular subject population) in administering medical fluids, rather than problems related to the interfaces (one of the certification requirements for PCA operation is that the users must also be trained in medical fluid administration).

Interpretation of Errors as Symptoms

In addition to the analysis of errors in quantitative terms, an interpretation of the various programming errors with respect to interface design is provided here. An error may be viewed as the symptom of a problem rather than the problem itself. Many of the programming errors documented in the experiment can be correlated with the problems identified in the analysis of the old interface. Some of these errors and their corresponding design problem are highlighted below:

* In the loading dose/bolus dose subtask, several mistakes occurred when subjects attempted to use the LOAD DOSE touch switch instead of YES/ENTER to enter the bolus dose setting. Recall that LOAD DOSE performs two functions, one is to initiate the loading dose set up procedure, and the second is to begin infusion of the loading dose after the

dose setting is entered. This error is symptomatic of unclear touch switch functions and appears to be aggravated by the multifunctional nature of the LOAD DOSE touch switch.

* For the concentration and mode subtasks, errors occurred when subjects mistakenly accepted inappropriate default options presented serially by the programming sequence (e.g., accepting the first concentration default, 1mg/ml, or the first mode option, PCA mode). These errors can be linked to the old interface's complex programming sequence, the lack of visibility of decision options and inappropriate defaults settings.

* In the process of recovering from errors such as an incorrect concentration setting or mode selection, subjects attempted to use REVIEW/CHANGE, which would return the user several screens back, in some cases to the previous subtask. This is inconsistent with the use of the touch switch in other system states, and is a clear example of tedious error recovery and inconsistent control functions with the old interface.

3.3.4 Subjective Preference

In the post-experiment interviews, all 12 subjects expressed a preference for the new interface over the old. Comments obtained from subjects throughout the experiment are shown in Table 3.7. In general, subjects felt that the new interface was "easier to use" and cited interface features such as the local menu displays and proper defaults as having facilitated the programming tasks.

Table 3.7. Comments obtained from novice subjects throughout the experiment.

OLD INTERFACE
- loading dose seemed tedious
- defaults on old interface make it tiresome to change
- looked simpler, but was not simpler to use than new interface
- more complicated
- hated it
NEW INTERFACE
- preferred the new interface's screens and buttons
- easy, no problems
- summary at the end is helpful
- liked the options given together on one screen
- straightforward, easy, and you don't have to remember things
- easy to use

To summarize the experimental findings, novice PCA programmers completed programming tasks significantly faster, made fewer programming errors, and associated lower mental workloads with the new interface compared to the old. Moreover, interviews with the nursing students at the end of the experiment showed a unanimous preference for the new interface over the old.

While this experiment has provided evidence of improved performance with the new interface, limitations of this experiment should be addressed. One of the primary limitations is that the results generalize only to nursing students who have no previous programming experience with PCA machines and who have only limited experience in a clinical environment. The results from this study do not necessarily generalize to a population of professional nurses, the actual end-users of the PCA device. This issue motivated the next experiment which evaluated the interfaces with a population of professional nurses who have extensive PCA programming experience.

4. EXPERIMENT II: COMPARATIVE EVALUATION WITH EXPERIENCED USERS

4.1 Hypothesis

The hypothesis of this experiment is that the results observed in Experiment I extend to experienced PCA programmers. More specifically, the redesigned interface facilitates PCA programming which is expected to result in improved programming time, reduced mental workload, and fewer programming errors compared to the old interface.

4.2 Methods

The materials, experimental design, procedure and performance measures used in this experiment were the same as those used in Experiment I (section 3.2). However, in place of the novice users, the subject population in this experiment consisted of experienced PCA users.

4.2.1 Subjects

The subject population for this experiment was chosen based on their training and experience with the PCA pumps. The most experienced group of PCA programmers was

sought. Twelve recovery room nurses from The Toronto Hospital were selected to participate in the experiment. The subjects have an average of 5.15 (standard deviation=1.19) years of experience with PCA programming.

4.3 Results and Discussion

4.3.1 Performance Time

Interface Effects

Out of the 12 recovery room nurses, 11 completed programming tasks quicker with the new interface than with the old. Recovery room nurses completed programming tasks with the old interface in an average of 2.13 minutes (standard deviation=0.60). In comparison, programming tasks with the new interface were completed in 1.74 minutes (standard deviation=0.08), or 18% faster (refer to Figure 4.1(a)). Results of an analysis of variance (see Table 4.1) of overall task completion times showed that this difference between interfaces is indeed significant ($F(1,10)=12.17$, $p=0.0058$).

Programming Mode

Mode was also statistically significant ($F(2,20)=5.4$, $p=0.013$). As was true with the nursing students, the P+C mode took the longest to program. However, unlike the nursing students, the recovery room nurses programmed the P mode most quickly, which reflects several years of experience programming the PCA machines in the P mode (refer to Figure 4.1(b)).

Table 4.1. Summary of statistical results for overall task completion time, mental workload, and programming errors. Statistical significance is denoted by *.

Performance Measure	Factor	Test Statistic	Significance Level
Overall Task Completion Time	interface	$F(1,10) = 12.17$	$p=0.0058$ *
	repetition	$F(1,10) = 21.81$	$p=0.0009$ *
	mode	$F(2,20) = 5.4$	$p=0.013$ *
	repetition x mode	$F(2,20) = 5.64$	$p=0.0114$ *
Workload	interface	$F(1,10) = 1.06$	$p=0.33$
		$\chi^2(1) = 0.4$	$p>0.3$, one-tailed
	repetition	$F(1,10) = 6.27$	$p=0.03$ *
	interface x mode x repetition	$F(2,20) = 8.62$	$p=0.002$ *
Errors	interface	$\chi^2(1) = 9.38$	$p<0.0025$, one-tailed*

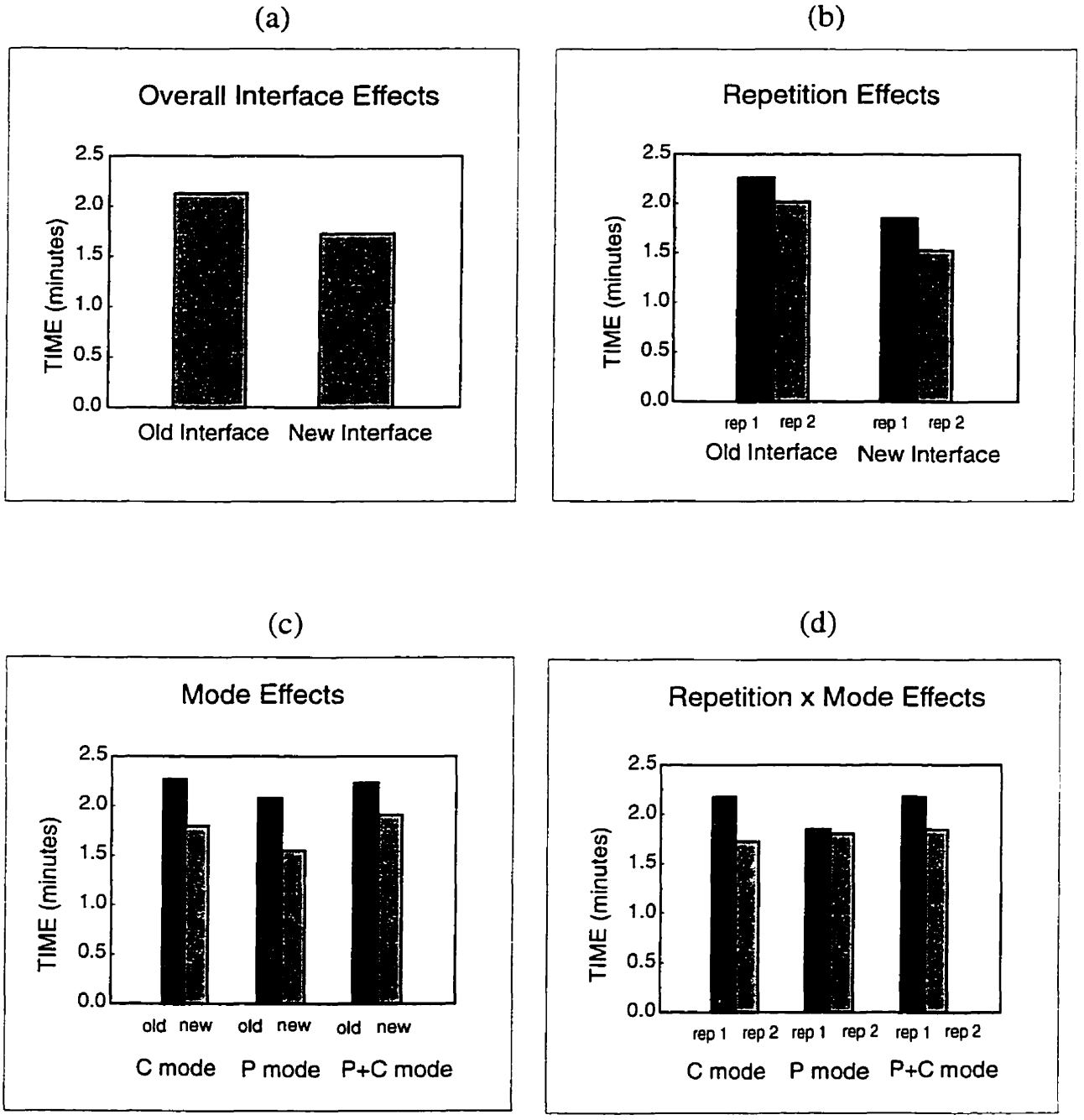


Figure 4.1. Comparison of task completion times (TCT) for the old and new interfaces. (a) Average TCT plotted for each interface. (b) Improvement in TCT over two repetitions. (c) Effects of mode on TCT. (d) Repetition x mode effect on TCT.

Practice x Programming Mode

Overall task completion times improved significantly in the second repetition of each programming task ($F(1,10)=21.81$, $p=0.0009$). The first repetition yielded an average programming time of 2.06 minutes, which improved to 1.81 in the second repetition (Figure 4.1(c)). Furthermore, there was a significant mode x repetition effect ($F(2,20)=5.64$, $p=0.0114$) in which there were marked improvements in the C and P+C modes of programming (the more unfamiliar programming tasks to nurses) over two repetitions, and only marginal improvement in the P mode (refer to Figure 4.1(d)).

Subtask Completion Times

An ANOVA was conducted on subtask completion times to test the effects of the experiment's independent variables and the effect of subtask on performance time at the subtask level. As shown in Table 4.2(a), the results revealed that there were significant differences between the two interfaces in subtask performance independent of the order of training. There was also a subtask x interface interaction in which the difference (in subtask completion times) between the old and new interfaces varied depending on the subtask.

To identify the subtasks in which performance differentials were most pronounced, an ANOVA was conducted for each of the nine subtasks. The results (detailed in Table 4.2(b)) show there that were significant differences between interfaces in subtask completion times for nearly all of the subtasks (refer also to Figure 4.2). Seven out of nine subtasks were completed significantly faster with the new interface than with the old (the time to complete the 'purging' and 'mode' subtasks showed non-significant improvements over the old interface). It appears that experienced PCA programmers take full advantage of the proper default settings on the new interface. However, with the more unfamiliar subtasks, such as purging and mode selection[4], the nurses completed the subtasks with the new interface as quickly as with the old, despite having greater familiarity with the old interface.

[4] In clinical practice, the recovery room nurses at The Toronto Hospital ordinarily purge the tubing *manually* prior to programming the pump; in addition, they ordinarily program the pumps for PCA mode.

Table 4.2(a). Summary of statistical results for subtask completion times. Statistical significance is denoted by *.

Performance Measure	Factor	Test Statistic	Significance Level
Subtask Completion Time	interface	F(1,10)=47.44	p=0.0001*
	order x interface	F(1,10)=0.02	p=0.8805
	subtask	F(8,80)=404.71	p=0.0001*
	subtask x interface	F(8,80)=15.43	p=0.0001*

Table 4.2(b). Summary of statistical results for each of the nine subtasks. Statistical significance is denoted by *.

Performance Measure	Subtask	Factor	Test Statistic	Significance Level
Subtask Completion Time	purging	interface	F(1,10)=1.38	p=0.2677
	concentration units	interface	F(1,10) = 38.8	p=0.0001*
	concentration value	interface	F(1,10) = 11.27	p=0.0073*
	bolus dose	interface	F(1,10) = 9.50	p=0.0519
	mode	interface	F(1,10) = 4.87	p=0.0001*
	pca dose	interface	F(1,10) = 15.76	p=0.0116*
	lockout	interface	F(1,10) = 7.96	p=0.0026*
	continuous rate	interface	F(1,10) = 17.81	p=0.0018*
	4hr limit	interface	F(1,10) = 243.25	p=0.0181*

4.3.2 Mental Workload

The average mental workload reported by the nurses was approximately 12% (standard deviation=0.08) for the new interface, which was slightly lower than that for the old interface (14%, standard deviation=0.10). However, an ANOVA showed that this difference was not statistically significant (F(1,10)=1.06, p=0.33). A χ^2 test (refer to Table 4.1) also confirmed that the tally of subjects rating a higher workload with the old interface (6) was not significantly greater than that of subjects rating a higher workload with the new interface (4). These results, however, should be interpreted within the context of experience. Within only several minutes of practice with the new interface, the nurses reported workloads not different from that associated with the old interface with which they have had several years of experience (see Figure 4.3(a)).

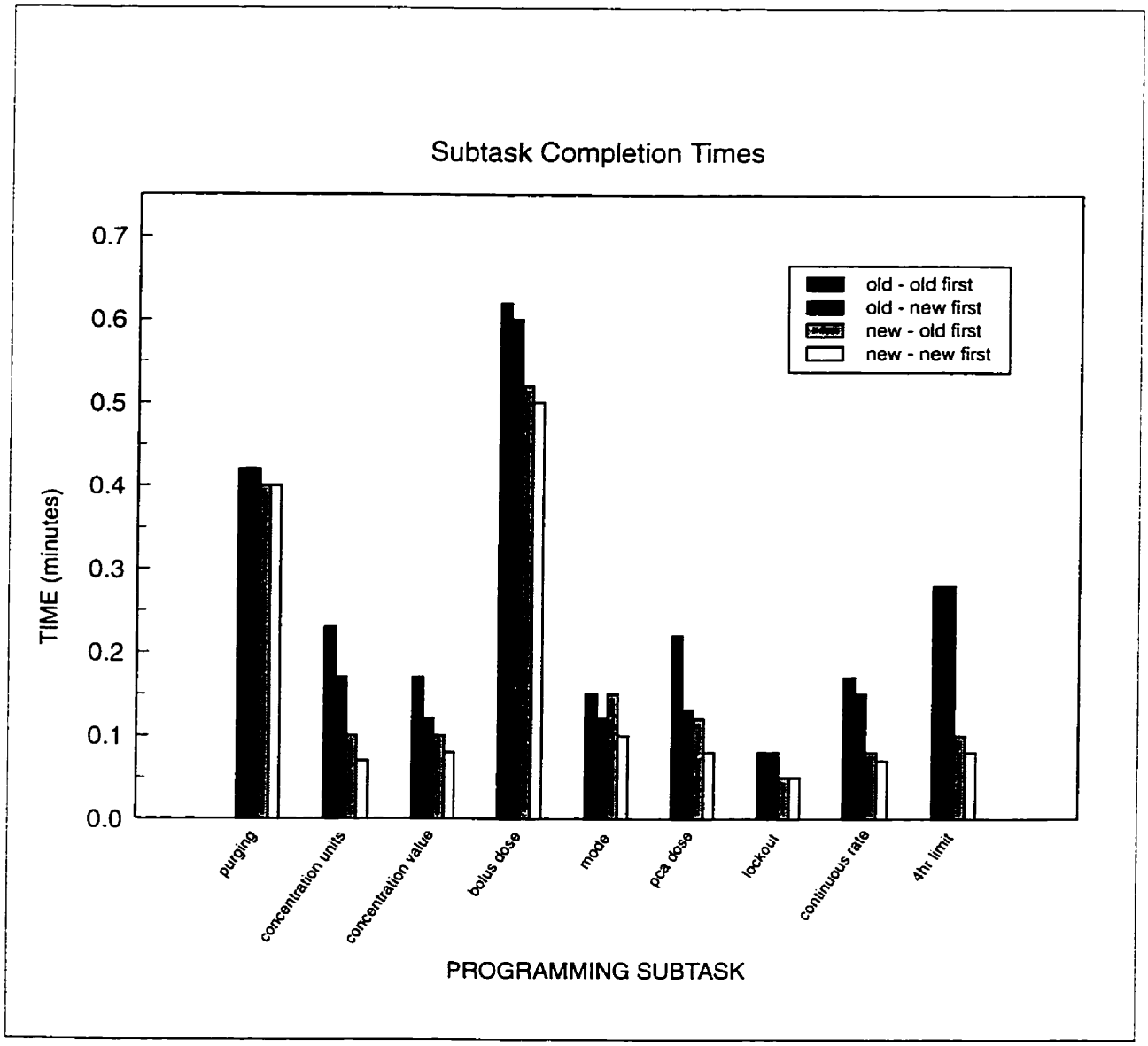


Figure 4.2. Completion times of each of the programming subtasks for experienced users. Illustrates differences in performance time between interfaces at the level of subtasks.

Workload ratings for the second repetition of the programming tasks were significantly lower than that for the first repetition ($F(1,10)=6.27$, $p=0.03$). Although there was no evidence of a mode x repetition effect on workload corresponding to that observed with overall task completion times, there was an interface x mode x repetition effect in which there was a more substantial reduction in reported workload for the new interface with the C and P+C modes over two repetitions compared to that for the old interface ($F(2,20)=8.62$, $p=0.002$). This suggests that with the new interface, nurses found it progressively easier to program these unfamiliar modes. Meanwhile, the workload reduction was less pronounced with the old interface (and even increased for the C mode of programming), despite greater familiarity and more experience with that interface (refer to Figure 4.3(b)).

4.3.3 Programming Errors

A total of 13 programming errors were made with the new interface, while almost three times as many, 34, were made with the old interface (refer to Table 4.3 and Figure 4.4(a)). A χ^2 test showed that this difference is statistically significant (χ^2 (1)=9.38, $p<0.0025$, one-tailed).

Further analysis of error severity did not reveal any significant differences between the interfaces in terms of their proneness to critical errors. There were 14 critical errors made with the old interface, 6 of which were a result of setting the wrong concentration. In comparison, there were 7 critical errors made with the new interface, none of which were concentration errors. All of the critical errors with the new interface were related to the omission of administering a bolus dose. Administering a bolus dose during programming is a practice that is unfamiliar to the nurses (see Table 4.6 for nurses' comments). Hence, it is likely that many of the bolus errors made with the new interface were caused in part by the incongruity of experimental conditions with standard clinical practice at the Toronto Hospital.

The incidence of errors was equally distributed among the three modes of programming. However, as many as 12 errors made with the old interface were associated with selection of the wrong mode. With the new interface, only 2 errors were related to the mode. That the new interface appears to be less prone to mode errors is likely attributable to the local menu displays, providing the user with a global view of choices for mode selection.

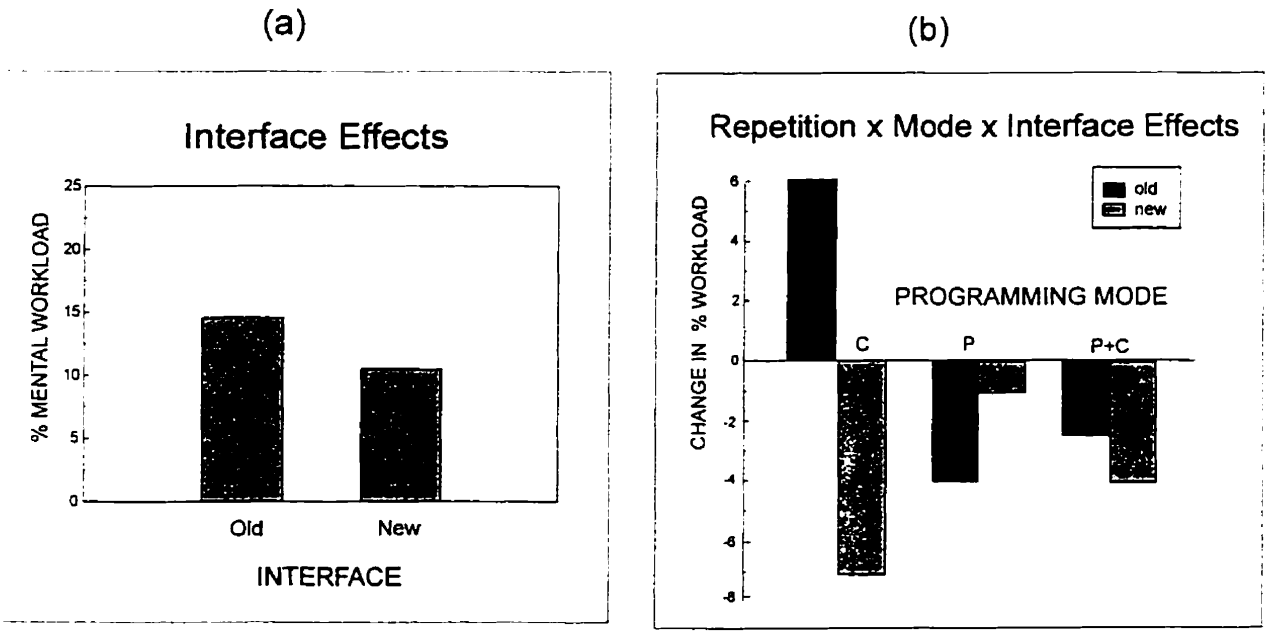


Figure 4.3. Mental workload ratings from experienced users for the old and new interfaces. (a) Workload for each interface. (b) Change in % mental workload over two repetitions (repetition 2 - repetition 1).

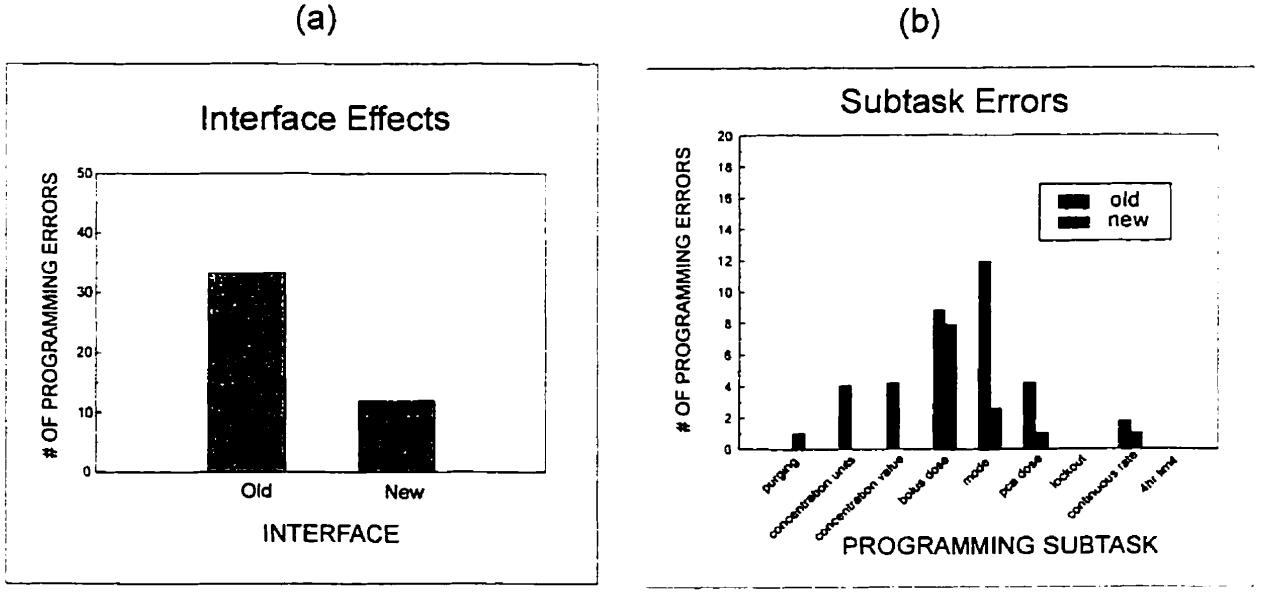


Figure 4.4. Programming Errors. (a) Total number of programming errors for each interface. (b) Distribution of errors across subtasks.

Table 4.3. List of programming errors made by experienced users.

Subject	Order*	Interface	Trial	Mode	Error Category	Consequence to Patient?	Description of Error(s)	Consequences
MM	1	old	1	c	mode	no	mistakenly answered NO to continuous selection, later corrected	
MM	1	new	1	c	mode	no	selected p instead of c, later corrected	
MM	1	new	2	p	bolus	no	pressed START instead of YES to set bolus dose	
SK	1	old	2	c	purge	no	mistakenly pressed PURGE instead of YES to Purge? prompt	
SE	1	old	1	p	conc units	no	mistakenly answered NO to MG/ML? prompt, later corrected	
SE	1	old	1	p	conc	yes	entered 1mg/ml (YES to default) instead of 2mg/ml	each dose 2 mg instead of 1 mg 4 hr limit 60 mg instead of 30 mg
SE	1	old	1	p	mode	no	selected c instead of p mode, later corrected	
SE	1	old	1	p	mode	no	attempted to correct previous error in mode selection but selected P+C instead of p mode, later corrected	
SE	1	old	1	p	pca dose	yes	entered 2mg instead of 1 mg (compounded with earlier conc error)	each dose 4 mg instead of 2 mg
SE	1	old	2	c	conc units	no	mistakenly answered NO to MG/ML?, later corrected	
SE	1	old	3	p+c	conc units	no	mistakenely set units to UG/ML instead of MG/ML, later corrected	
SE	1	old	3	p+c	mode	yes	selected p instead of P+C mode	no conintuous infusion (0 instead of 1mg/hr)
SE	1	old	3	p+c	pca dose	yes	entered 2 5mg instead of 1 5mg	each dose 2 5 mg instead of 1 5 mg
SE	1	old	3	p+c	continuous rate	yes	failed to set continuous rate as a result of earlier error in mode selection	no conintuous infusion (1 5mg/hr)
SE	1	old	4	p	conc value	no	entered 1 5mg/ml instead of 2 5mg/ml, later corrected	
SE	1	old	4	p	pca dose	no	entered 2 5mg instead of 1 mg, later corrected	
SE	1	old	4	p	bolus	yes	infused a second bolus dose	bolus dose 3mg instead of 1 5mg
SE	1	old	5	c	mode	no	selected p instead of c mode, later corrected	
SE	1	old	6	p+c	mode	yes	selected p instead of p+c mode	no continuous infusion (0 instead of 1 mg/hr)
SE	1	old	6	p+c	pca dose	no	entered 0 2mg instead of 1mg, later corrected	
SE	1	old	6	p+c	continuous rate	yes	failed to set continuous rate as a result of earlier error in mode selection	no continuous infusion (0 instead of 1mg/hr)
SE	1	new	2	c	mode	no	selected p+c instead of c mode, later corrected	
SE	1	new	3	p+c	bolus	yes	did not administer bolus dose	bolus dose 0mg instead of 1 5mg
SE	1	new	4	p	pca dose	no	entered 2 5mg instead of 1mg, later corrected	
SE	1	new	5	c	mode	no	selected p instead of c mode, later corrected	
MH	1	old	1	c	mode	no	selected p+c instead of c mode, later corrected	
MH	1	old	5	p+c	mode	no	mistakenly answered NO to p+c mode, used REVIEW in an attempt to correct, but reverted back to conc subtask, eventually corrected	
MH		new	1	c	continuous rate	no	entered 1mg/hr instead of 1 5mg/hr, later corrected	
HT	1	old	1	p	bolus	yes	did not administer bolus dose	bolus dose 0mg instead of 1 5mg
HT	1	old	3	c	mode	no	mistakenly answered NO to c mode, later corrected	
HT	1	old	5	p+c	bolus	no	mistakenly pressed LOAD instead of YES to set bolus dose	
LB	2	old	2	c	bolus	no	entered 1 6mg instead of 1 5mg	
ET	2	old	5	p+c	conc value	yes	entered 1 5 instead of 2 0mg/ml	each dose 3 mg instead of 1 mg continuous rate 1 3 mg/hr instead of 1 5 mg/hr 4 hr limit 35 mg instead of 28 mg
ET	2	new	1	p	bolus	yes	did not administer bolus dose	bolus dose 0mg instead of 1 5mg
ET	2	new	2	p+c	bolus	yes	administered second bolus dose	bolus dose 3mg instead of 1 5mg
ET	2	new	3	c	bolus	no	mistakenly pressed START instead of YES to set bolus dose	
ET	2	new	4	p	bolus	yes	did not administer bolus dose	bolus dose 0mg instead of 1 5mg
ET	2	new	5	p+c	bolus	yes	did not administer bolus dose	bolus dose 0mg instead of 1 5mg
ET	2	new	6	c	bolus	yes	did not administer bolus dose	bolus dose 0mg instead of 1 5mg
WD	2	old	1	p+c	conc units	no	mistakenly answered NO to MG/ML, later corrected	
SH	2	old	2	c	conc	yes	entered 1mg/ml (YES to default) instead of 1 5mg/ml	continuous rate 2 25 mg/hr instead of 1 5 mg/hr 4 hr limit 45 mg instead of 30 mg
SH	2	old	6	p	mode	no	mistakenly answered NO to p mode, used REVIEW in an attempt to correct, but reverted back load dose subtask, later corrected	
SG	2	old	1	c	4 hr limit	no	mistakenly answered NO to "4hr limit set?", later corrected	
SG	2	old	4	c	mode	no	mistakenly selected p instead of c mode, later corrected	

* Order 1 = old first
2 = new first

This feature seems to be particularly useful in overcoming subjects' habitual tendencies to program in the P mode.

Interpretation of Errors as Symptoms

In the experiment with the novice subject population, errors were analyzed and treated as symptoms of interface design problems. Below, the same approach is taken to examine the programming errors with the old interface made by the experienced population of users. Each design issue is supported by an example of a common programming error:

* Several errors in the mode subtask occurred with the old interface. Subjects made incorrect selections of operating mode options which were presented in serial. This is partially attributable to the custom clinical practice of selecting the P mode of operation. However, the frequency of this error with new interface is much lower, suggesting that errors in mode selection with the old interface may also be partly attributable to programming sequence complexity.

* While attempting to recover from mode selection errors, subjects either scrolled through the loop to find the correct mode, or used REVIEW/CHANGE, which sent them back to the concentration subtask. This illustrates the tedious error recovery and inconsistent control functions of the old interface.

4.3.4 Subjective Preference

Out of the 12 nurses in the experiment, 9 preferred the new interface over the old, while 1 preferred the old interface, and 2 were impartial between the two interfaces. A χ^2 test conducted on the proportion of subjects expressing preference for one interface over the other showed that the number of subjects preferring the new interface is significantly greater than the number preferring the old interface ($\chi^2(1)=6.4$, $p<0.012$).

Comments made by the nurses throughout the experiment are documented in Table 4.6. Generally, nurses found the new interface easier to use than the old interface. Although many of the comments suggest that the nurses also found the new interface easier to *learn* than the old interface, it is important to consider the contrasting training approach and training environment they received with each interface.

Table 4.6. Comments obtained from recovery room nurses in the experiment. Table continues on next page.

GENERAL COMMENTS
- bolus dose is not usually infused during programming; bolus is usually done by another nurse after programming is all finished
- much easier to program without distractions: in the recovery room, rarely uninterrupted
- bolus is usually not written on the order form; it's usually left to the discretion of the nurse, and is normally set anywhere from 1mg to 4mg
- not accustomed to seeing continuous mode, or concentrations other than 2mg/ml
- better interface would make a bigger difference for someone who doesn't use it often.
- when you use (a device) often, the interface doesn't matter as much
- the machine is usually hooked up to the patient after programming is completed
- thinks that someone should've asked their opinions when the old PCA first came out
- all PCA's used to be set up the night before all at once (e.g., 18 of them were set up for the next 18 patients who were coming in the next morning.)
- not as much pressure as programming in the recovery room because of all the distractions
- purge is usually done manually before programming is started
- difficult to compare one machine which you've had a lot of experience with to one which you don't
OLD INTERFACE
- originally given demonstration and some hands-on practice, but the PCA machines were brought into the hospital for clinical use long enough after training that we forgot how to program the PCA; we had to program the PCAs for patients without any further training or practice. We had to figure it out right on the spot
- when (Lifecare) PCA machines were first brought in to the recovery room, they were set up hours before or even the night before patients were scheduled for surgery, because the setup was so cumbersome. So several of them would be set up all at once and lined up for the next day. After the hospital realized that so many drug cartridges were going to waste when surgeries were canceled/postponed, they (nurses) had to start programming the devices when they received the patient or order form.
- mental demand for the old interface was probably higher (than it is now) during the first few times using the device
- purging on the old interface was too slow so they no longer do it; now purging is done manually because it's quicker.
- the manufacturers should have asked our opinion about the (old) PCA machine before they brought it in and made the us use them
- workload (for the old PCA) was much, much higher than it is now. When we first got the device, it was so difficult to learn, even though all of us are really good at it now. It was so difficult partly because we had to learn it while also doing other things
- "the old interface isn't so bad if the nurse uses it every day, but it would probably be very difficult if it was someone on the floors who doesn't program it everyday."
- liked the buttons on the old interface because they are closer together
- both interfaces are quite easy, but prefers old because she is already so used to it and is already accustomed to the screens. However, with a little more practice, subject thinks she'd like the new interface just as much
- seems like old interface has less buttons
- towards the end of the session with the old interface, the subject is frustrated and remarks, "they (the designers) are doing this on purpose to catch us off guard. Why don't they make it easier for us?"

Table 4.6. Comments obtained from recovery room nurses in the experiment (continued from previous page).

NEW INTERFACE
- likes the idea of the summary showing all the settings
- new interface doesn't take long to learn
- new interface is easier than current (old) interface
- liked the menu system for choosing mode, and concentration units
- likes the speed of the increment/decrement touch switches;
- summary showing all the settings is nice
- programming the new interface is a little bit faster; defaults make it faster than the old
- "took a while to learn and get used to the new interface"
- liked the menu on the new interface and the indicator box showing the selected choice
- bolus is not shown in the summary of the new interface (it should be shown)
- new interface is "more fun to program"
- buttons on new interface have a "good arrangement"
- simple, user friendly, easier to learn than old interface

Many nurses stated in interviews that they found PCA programming difficult and lengthy at the time PCA pumps were first introduced to the recovery room at The Toronto Hospital. During that time, the pumps were set up hours before the patient's scheduled surgery to accommodate the difficult and time consuming programming procedure. Several factors instigated a change in this practice; one was improved proficiency in programming, and the other was related to unnecessary waste of the PCA drug cartridges that resulted when surgical procedures were rescheduled or canceled. Currently, nurses program the pumps "on the spot" when they receive the patient and the PCA order form.

One reason for the difficult and extended learning period with the old interface is that nurses were originally given instruction several weeks before devices were actually brought into the hospital for clinical use. The intervening time period was sufficient to allow for the material to be forgotten, forcing the nurses to relearn what had been taught in their initial training session. Moreover, nurses had to cope with a busy and demanding environment while relearning how to program the PCA pumps for patients. A common statement from nurses was that it took them several months to become proficient or confident with programming under these circumstances.

In summary, experienced PCA programmers performed significantly faster with the new interface compared to the old, and made fewer errors. Mental workload was found to be lower with new interface than the old interface, but the difference was not statistically significant. Subjective preferences obtained from interviews following the experiment showed that a majority (90%) of nurses who expressed a preference, preferred the new interface over the old interface.

5. GENERAL DISCUSSION

The empirical evaluations have provided evidence of improved performance with the redesigned interface over the existing Lifecare interface for both novice and experienced populations of PCA users. Taken collectively, the results of the experiments indicate that, regardless of the level of user expertise in PCA programming, the new interface is superior in efficiency and safety over the old interface. Furthermore, mental workload associated with the new interface is the same or lower than that associated with the old interface.

Interestingly, there were few differences in performance between experienced and novice subjects. In an attempt to confirm that experience did not play a significant role in influencing performance, 'experience' was added to the list of independent variables (used in the experiments) and an ANOVA was conducted on the combined data from the two experiments. The results of the ANOVA showed that there was indeed no significant difference in task completion times and workload between the two populations ($F(1,10)=0.09$, $p=0.765$ and $F(1,10)=3.84$, $p=0.0641$, respectively).

There are several possible explanations as to why experience provided very little performance advantage. The first pertains to the use of a software prototype rather than a physical prototype. Recovery room nurses, having had previous experience on the actual device, may have experienced a negative transfer of training such that their performance with the software prototype was negatively affected. Furthermore, the input modality of the simulations may have also been a source of hindrance to the nurses' performance since they were observed to be slightly less skilled at using a mouse than were the students. Future studies would be required to ascertain the fidelity of the simulations and input modality.

A second explanation as to why students performed equally well as the nurses is that previous programming experience does not lead to better programming times. This, however, contradicts statements from nurses suggesting that improvement in programming proficiency had occurred over months of repeated practice.

A final, more plausible explanation as to why nursing students were able to attain a comparable level of proficiency as their more experienced cohorts relates to the environment in which the experiment took place. Nursing students were given individual instruction in a quiet, segregated environment, whereas the training that recovery room nurses originally underwent for PCA certification was immensely different. The comparatively favourable environment in which nursing students were trained likely had an impact on accelerating learning. This offers support for the view that the environment in which the device is used plays an important role in shaping performance. Future studies should investigate how learning and performance are affected by environmental stressors such as noise, time pressure or interruptions.

In addition to comparable programming times, programming errors made by novice and experienced subjects were also similar in nature. Both the experienced and novice subjects were observed to have particular a proneness to errors associated with the decision points in the decision/action structure of the programming sequence. Figure 5.1 highlights the sections of the programming sequence where programming errors most frequently occurred. The two most frequent errors made with the old interface resulted from either an incorrect concentration setting or an incorrect mode selection (totaling 13 and 15 errors, respectively). With the new interface, the two most common errors were related to bolus dose, 8 involved administering an incorrect dose (e.g., infusing 2 doses instead of a single dose), and 6 resulted from failing to administer a bolus dose at all. As was mentioned in earlier discussions on programming errors, many of the errors observed with the new interface may have been caused by factors such as the students' lack of clinical training in medical fluid administration, misreading the order form, or conflicts between experimental tasks and standard clinical procedures in the recovery room. Further studies would be required in order to determine whether the error rates observed in the experiments generalize to a field setting. The issues raised in this discussion may be areas for future investigations.

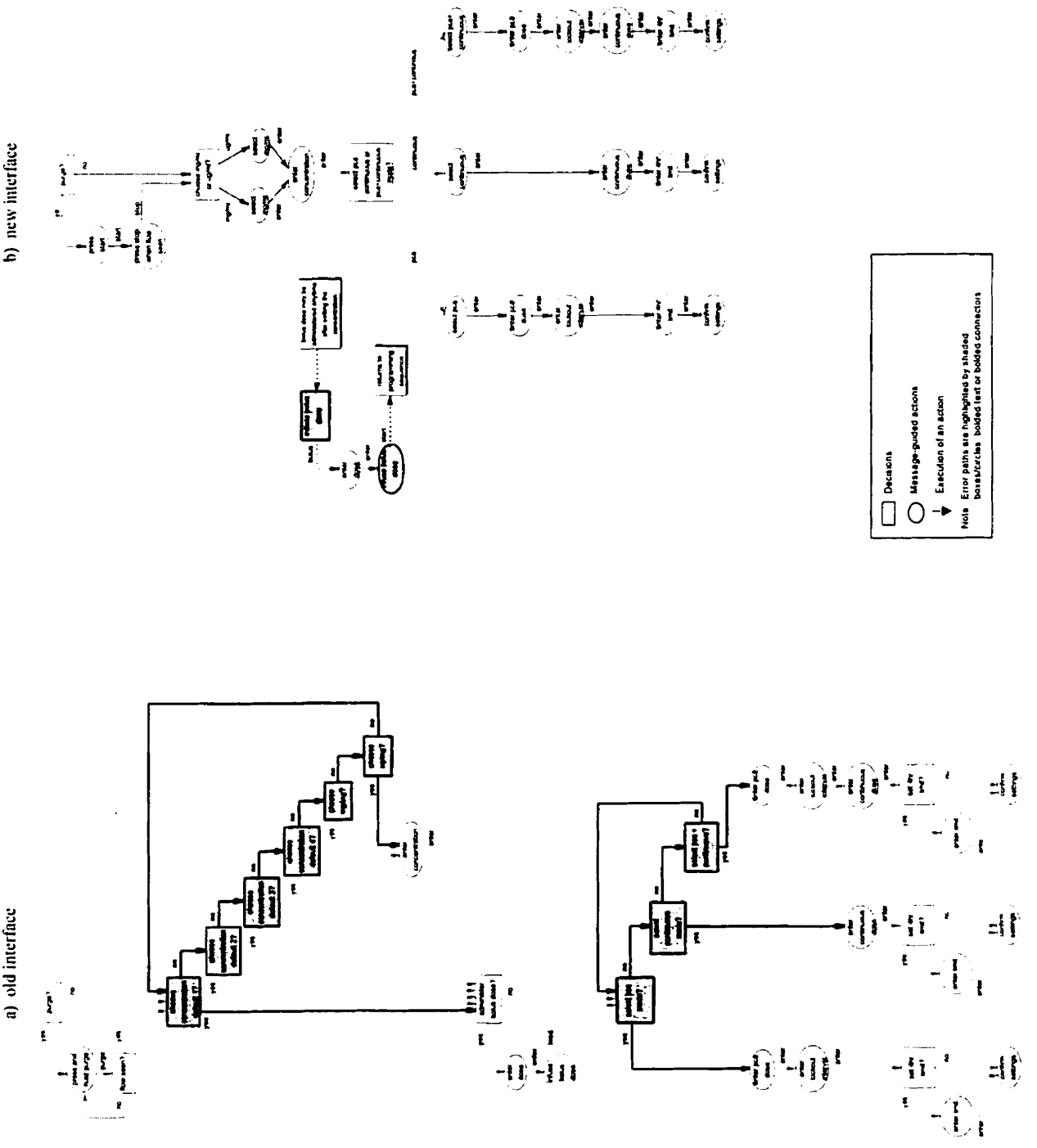


Figure 5.1. Most frequent error paths along the decision/action structure of PCA programming. (a) Programming errors with the old interface. (b) Programming errors with the new interface.

Recommendations for Future Studies

The preceding discussion brought attention to several recommendations for future studies. These recommendations will be recapitulated and elaborated upon below.

The first recommendation is to test the fidelity of the software simulations in order to investigate whether performance with the software simulations is an accurate predictor of performance with the actual device. The testing should compare programming performance on the actual PCA device with that on the simulated device. Also, fidelity tests should include a comparison of various input devices, e.g., mouse and touch screen, to test whether input modality has any impact on performance.

The second recommendation is to test the efficacy of different PCA order forms in order to determine whether programming performance can be further improved. It is expected that improving the design of the order form would help to solve problems arising from misreading the order form, and therefore help to prevent resulting programming errors.

The third recommendation is to conduct an evaluation of the interfaces with clinically trained nurses who have little or no PCA programming experience (e.g., operating room nurses or floor nurses). Comparing performance of these nurses with that of the recovery room nurses would help distinguish the effects of clinical experience from programming experience, and would also help to establish whether the performance differentials observed between the interfaces generalize to less experienced or less frequent PCA programmers.

The final recommendation for future studies is to conduct similar empirical evaluations of the interfaces under more representative environmental conditions. This would entail using controlled distractions or interruptions to examine whether performance differentials between the interfaces magnify or diminish, or whether the rate of learning with the two interfaces are differentially affected by interruptions.

6. CONCLUSION

This thesis has delineated the implications of a poorly designed PCA device from the perspective of its users: unnecessary mental workload for the clinician, and increased risk of injury or death for the patient. Clearly, methods which reduce morbidity and mortality rates

are of significant importance. Motivated by these observations, this thesis has demonstrated the notion that safety, efficiency, and ease of use of PCA machines can be improved by adopting a human factors approach to interface design. The findings lend support to the broader prospect of applying these methods to other devices to improve medical device safety.

7. REFERENCES

Abbott Laboratories (1989). *Lifecare 4100 PCA Plus II Infuser System Operating Manual.* North Chicago: Abbott Laboratories

Aucella, A.F., Kirkham, T., Barnhart, S., Murphy, L. & LaConte, K. (1994). Improving Ultrasound Systems by User-Centered Design. In *Proceedings of the Human Factors and Ergonomics Society* (pp. 705-709). Santa Monica, CA: Human Factors and Ergonomics Society

Bancroft, M.L. (1989). Accidental Breathing System Disconnection in a Modern Intensive Care Unit: Human Factors Solutions. In *Proceedings of the Human Factors Society* (p. 1170). Santa Monica, CA: Human Factors Society

Betram, D.A. (1991). Measures of Physician Mental Workload. In *Proceedings of the Human Factors Society* (pp. 1293-1294). Santa Monica, CA: Human Factors Society

Bogner, M.S. (1995a). Human Factors and Medical Devices: Lack of Feedback. *FDA User Facility Reporting Bulletin, 14,* 1-8

Bogner, M.S. (1995b). Technology, Human Error, and Standards. In *Proceedings of the Meeting for Health Care Technology Policy II* (pp. 374-384). Arlington, VA: The International Society for Optical Engineering

Botney, R. & Gaba, D. M. (1995). Human Factors Issues in Monitoring. In C. D. Blitt, & R. L. Hines (Eds.), *Monitoring in Anesthesia and Critical Care Medicine 3rd Edition.* Churchill Livingstone, New York

Burlington, B. (1995). Human Factors and the FDA's Goals: Improved Medical Device Design. In *Proceedings of the AAMI FDA Conference: Human Factors in Medical Devices: Design, Regulation, and Patient Safety* [Online]. Arlington: Association for the Advancement of Medical Instrumentation. Available URL: http://www.fda.gov/cdrh/ humfac/ hufacimp.html #AAMI/FDA

Callan, C.M. (1990). An Analysis of Complaints and Complications with Patient-Controlled Analgesia. In Ferrante, F.M., Ostheimer, G.W. & Covino, B.G. (Eds.), *Patient-Controlled Analgesia* (pp. 139-150). Boston: Blackwell Scientific Publications

Callan, J.R., Gwynne, J.W., Kelly, R.T., Muckler, F.A, & Schoenfeld, I. (1991). Task Analysis of Remote Afterloading Brachytherapy Cancer Treatment. In *Proceedings of the Human Factors Society* (pp. 675-678). Santa Monica, CA: Human Factors Society

Carstensen, P.B. (1995). Overview of FDA's New Human Factors Plan: Implications for the Medical Device Industry. In *Proceedings of the AAMI FDA Conference: Human Factors in Medical Devices: Design, Regulation, and Patient Safety* [Online]. Arlington: Association for the Advancement of Medical Instrumentation. Available URL: http://www.fda.gov/cdrh/ humfac/ hufacimp.html#AAMI/FDA

Center for Devices and Radiological Health (1996). *Human Factors Implications of the New GMP Rule* [Online]. Available URL: http://www.fda.gov/cdrh/humfac/hufacimp.html

Charante, E. M., Cook, R. I., Woods, D. D., Lue, Y., & Howie, M. B. (1992). Human-Computer Interaction in Context: Physician Interaction with Automated Intravenous Controllers in the Heart room. In H. G. Stassen (Ed.), *Analysis, Design, and Evaluation of Man-Machine Systems*. (pp. 263-274). London: Pergamon Press

Cohen, M. R. (1993). Preventing Errors Associated with P.C.A. Pumps. Nursing, 23, 17

Cook, R. I., Potter, S. S., Woods, D. D., & McDonald, J. S. (1991). Evaluating the Human Engineering of Microprocessor Controlled Operating Room Devices. *Journal of Clinical Monitoring*, *7*, 217-226

Cook, R. I., Woods, D. D., & Howie, M. B. (1990). The Natural History of Introducing New Information Technology into a High-Risk Environment. In *Proceedings of the Human Factors Society Annual Meeting* (pp. 429-433). Santa Monica, CA: Human Factors Society

Cook, R. I., Woods, D. D., Howie, M. B., Horrow, J. C., & Gaba, M. D. (1992). Unintentional Delivery of Vasoactive Drugs with an Electromechanical Infusion Device. *Journal of Cardiothoracic and Vascular Anesthesia*, *6*, 238-244

Cooper, J. B., Newbower, R. S. & Kitz, R. J. (1984). An Analysis of Major Errors and Equipment Failures in Anesthesia Management: Considerations for Prevention and Detection. *Anesthesiology*, *60*, 34-42

Cooper, J. B., Newbower, R. S., Long, C. D. & McPeek, B. (1978). Preventable Anesthesia Mishaps: A Study of Human Factors. *Anesthesiology*, *49*, 399-406

Doniz, K., & Harkness, H. (1994). *Interface for Patient Controlled Analgesia Machine*. Unpublished B.A.Sc. Thesis, University of Toronto, Toronto, Ontario

Effken, J.A. Kim, N., Kadar, E. & Shaw, R.E. (1992). Enhancing Computer Displays to Support Coordination of Hemodynamic Monitoring and Treatment. In *Proceedings of the Human Factors Society* (pp. 1522-1525). Santa Monica, CA: Human Factors Society

Ferrante, F.M. & Covino, B.G. (1990). Patient-Controlled Analgesia: A Historical Perspective. In Ferrante, F.M., Ostheimer, G.W. & Covino, B.G. (Eds.), *Patient-Controlled Analgesia* (pp. 3-9). Boston: Blackwell Scientific Publications

Gaba, D. M., Maxwell, M. & DeAnda, A. (1987). Anesthetic Mishaps: Breaking the chain of accident evolution. Anesthesiology, 66, 670-676

Hart, S.G. & Staveland, L.E. (1988). Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. In Hancock, P.A. & Meshkati, N. (Eds.), *Human Mental Workload.* North Holland: Elsevier Science Publishers B.V.

Henriksen, K., Kaye, R.D. & Morisseau, D. (1991). Toward Conceptualization of Human Error in Teletherapy. In *Proceedings of the Human Factors Society* (pp. 670-673). Santa Monica, CA: Human Factors Society

Hyman, W.A. (1994). Errors in the Use of Medical Equipment. In Bogner (Ed.), *Human Error in Medicine* (pp.327-347). New Jersey: Erlbaum Associates

Ilsley, A. H., Owen, H., Plummer, J. L., Mackey, N. A., & Roberts, D. R. D. (1994). A System for Standardized Evaluation of Patient-Controlled Analgesia Devices: Design, Construction, and Engineering Aspects. *Journal of Clinical Monitoring, 10*, 194-200

Isla, R., & Lin, L. (1993). *Investigation of the Effectiveness of the User Interface of a Patient Controlled Analgesia Machine*. Unpublished B.A.Sc. Thesis, University of Toronto, Toronto, Ontario

Isley, A.H., Owen, H., Plummer, J.L., Mackey, N.A. & Roberts, D.R.D. (1994). A System for Standardized Evaluation of Patient-Controlled Analgesia Devices: Design, Construction, and Engineering Aspects. *Journal of Clinical Monitoring, 10(3)*, 194-199

Leape, L. L. (1994). Error in Medicine. *Journal of the American Medical Association, 272(23)*, 1851-1857

Lin, L., Isla, R., Doniz, K., Harkness, H., Vicente, K. J., & Doyle, D. J. (1995). *Applying Human Factors to the Design of Medical Equipment: Patient-Controlled Analgesia* (CEL 95-06). University of Toronto, Cognitive Engineering Laboratory, Toronto

McConnell, E.A. (1995). Using Medical Equipment: Risky Business. *Nursing, 25*, 62-4

McConnell, E.A. (1995). American Registered Nurse Medical Device Education: A Comparison of Simple and Complex devices. *Medical Instrumentation, 29(6)*, 520-526

Meadows, S. (1989). Human Factors Applications to Health Care Systems. In *Proceedings of the Human Factors Society* (p. 1167). Santa Monica, CA: Human Factors Society

Meadows, S. & Kelly, R. (1989). Human Factors of Blood Glucose Monitoring. In *Proceedings of the Human Factors Society* (p. 1168). Santa Monica, CA: Human Factors Society

Medical Devices Review Committee (1992). *Direction for Change: Report of the Medical Devices Review Committee*. Ottawa: Minister of Supply and Services Canada

Molich, R., & Nielson, J. (1990). Heuristic Evaluation of User Interfaces. In *Proceedings of CHI '90* (pp. 249-256). New York: ACM Press

Moll van Charante, E.M., Cook, R.I., Woods, D.D., Lue, Y., & Howie, M.B. (1992). Human-Computer Interaction in Contest: Physician Interaction with Automated Intravenous Controllers in the Heart Room. In H.G. Stassen (Ed.), *Analysis, Design, and Evaluation of Man-Machine Systems* (pp.263-274). London: Pergamon Press

Obradovich, J.H. & Woods, D.D. (1996). Users as Designers: How People Cope with Poor HCI Design in Computer-Based Medical Devices. *Human Factors, 38(4)*, 574-592

Ostrander, L. E. (1986). Attention to the Medical Equipment User in Biomedical Engineering. *Medical Instrumentation, 20*, 3-5

Owen, H., Glavin, R. J., Reekie, R. M., & Trew, A. S. (1986). Patient-Controlled Analgesia: Experience of Two New Machines. *Anaesthesia, 41*, 1230-1235

Owen, H., Thomas, D.W. (1988). Patient-Controlled Analgesia - The Need for Caution. *Anaesthesia, 43*, 770-772

Patt, R. B., Wu, C., Bressi, J., & Catania, J. A. (1993). Accidental Intraspinal Overdose Revisited. *Anesthesia and Analgesia, 76*, 202

Rachlin, J.A. (1995). Human Factors and Medical Devices. *FDA User Facility Reporting Bulletin, 12*, 1-4

Ready, L.B. (1995). How Many Acute Pain Services Are There in the United States, and Who Is Managing Patient-Controlled Analgesia? *Anesthesiology, 82(1)*, 322

Riley, C. (1991). Infusion Pumps: User Friendlies? Nursing, 21, 31

Sawaki, Y , Parker, R. K., & White, P. F. (1992). Patient and Nurse Evaluation of Patient-Controlled Analgesia Delivery Systems for Postoperative Pain Management. *Journal of Pain and Symptom Management, 7*, 443-453

Sawyer, D. (1995). CDRH's Approach to Providing Human Factors Information. In *Proceedings of the AAMI/FDA Conference: Human Factors in Medical Devices: Design, Regulation, and Patient Safety* [Online]. Arlington: Association for the Advancement of Medical

Instrumentation. Available URL: http://www.fda.gov/cdrh/humfac/ hufacimp.html#AAMI/ FDA

Serig, D.I. (1991). Medical Misadministrations and "Peeping Tom" Research. In *Proceedings of the Human Factors Society* (pp. 679-681). Santa Monica, CA: Human Factors Society

Smythe, M. (1992). Patient-Controlled Analgesia: A review. *Pharmacotherapy, 12,* 132-143

Van Cott, H.P. (1993). Human Error in Health Care Delivery: Cases, Causes, and Correction. In *Proceedings of the Human Factors and Ergonomics Society* (pp. 430-434). Santa Monica, CA: Human Factors and Ergonomics Society

Vicente, K. J., & Rasmussen, J. (1992). Ecological Interface Design: Theoretical Foundations. *IEEE Transactions on Systems, Man, and Cybernetics, SMC-22*, 589-606

Voelker, R. (1996). 'Treat Systems, Not Errors,' Experts Say. *Journal of the American Medical Association, 276(19),* 1537-1538

Weinger, M.B., Herndon, O.W., Zornow, M.H., Paulus, M.P., Gaba, D.M., & Dallen, L.T. (1994). An Objective Methodology for Task Analysis and Workload Assessment in Anesthesia Providers. *Anesthesiology, 80*, 77-92

Weisner, S.J. (1988). A Touch-Only User Interface for a Medical Monitor. In *Proceedings of the Human Factors Society* (pp. 435-439). Santa Monica, CA: Human Factors Society

White, P. F. (1987). Mishaps with Patient-Controlled Analgesia. Anesthesiology, 66, 81

Wickens, C. D. (1992). *Engineering Psychology and Human Performance. 2nd ed.* New York: Harper-Collins

Wiklund, M.E. (1989). Human Factors in Medical Product Design. In *Proceedings of the Human Factors Society* (p. 1169). Santa Monica, CA: Human Factors Society

Wiklund, M.E. & Hoffman, L.R. (1988). When Electronic Devices Outnumber Flower Bouquets in the Hospital Room. In *Proceedings of the Human Factors Society* (pp. 430-434). Santa Monica, CA: Human Factors Society

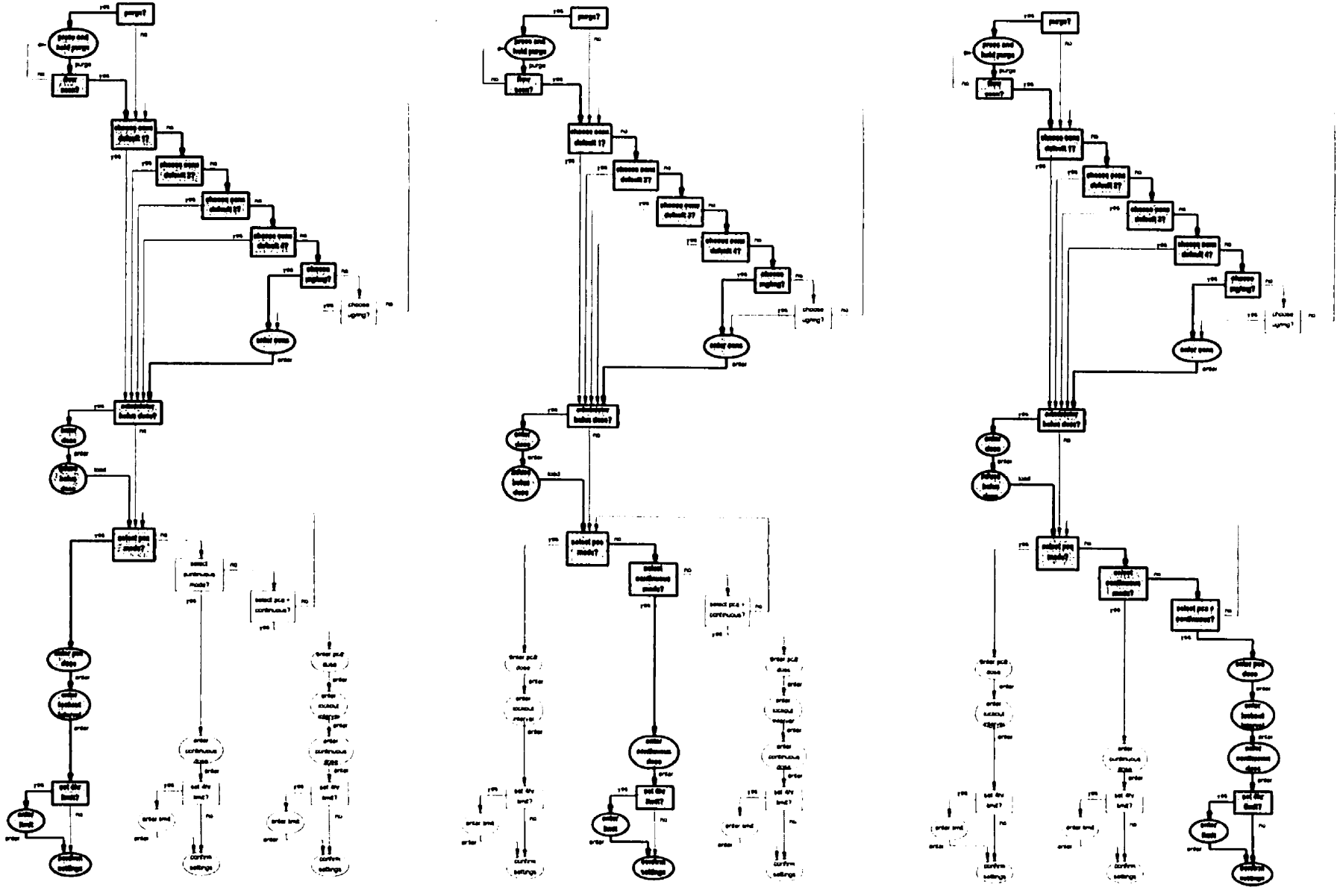
Yue, L., Woods, D. D., & Cook, R. I. (1992). *Cognitive Engineering of the Human-Computer Interface: Re-Design of an Infusion Controller in Cardiac Anesthesiology* (CSEL Report TR-01-92). Columbus, OH: Cognitive Systems Engineering Laboratory, Ohio State University

APPENDIX A

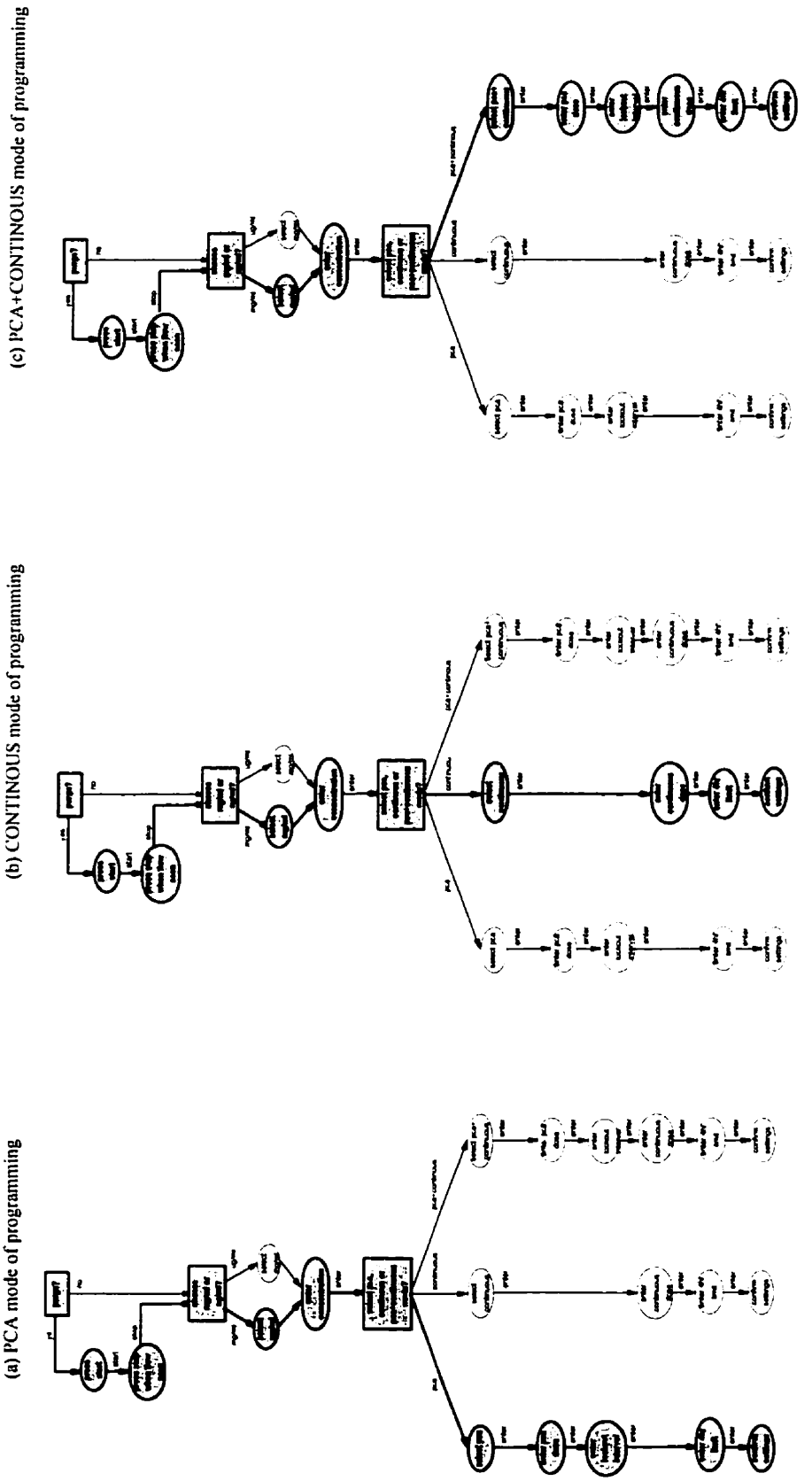
(a) PCA mode of programming

(b) CONTINOUS mode of programming

(c) PCA+CONTINOUS mode of programming

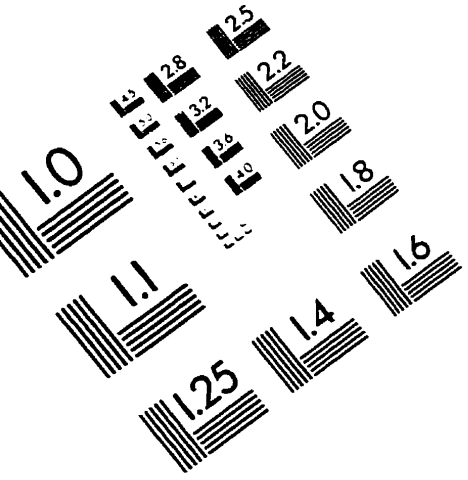


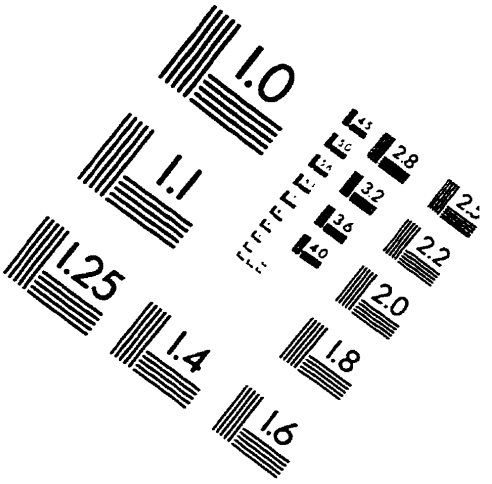
Appendix A - Figure A1: Correct programming path in decision/action structure of experimental tasks for the old interface

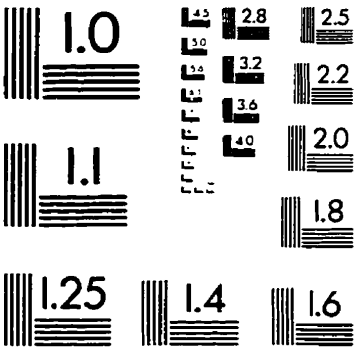


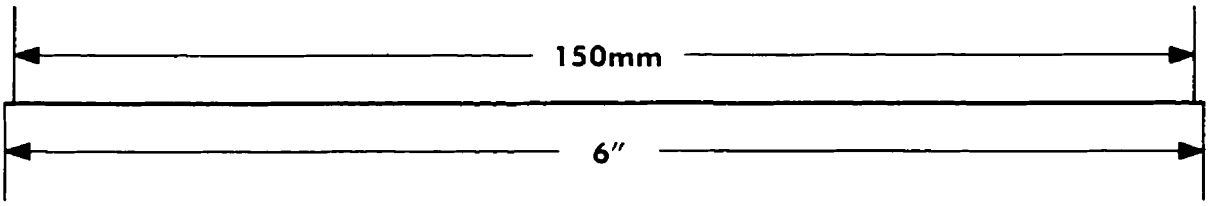
Appendix A - Figure A2: Correct programming path in decision/action structure of experimental tasks for the new interface

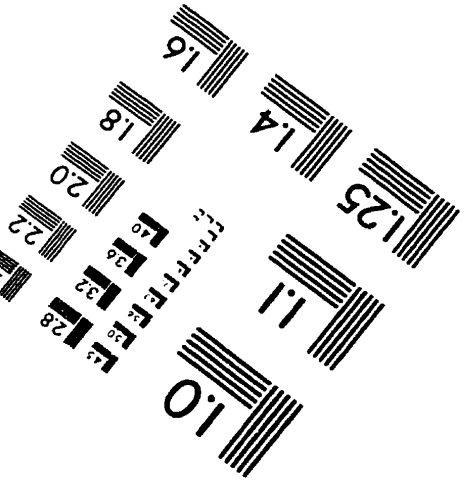
IMAGE EVALUATION TEST TARGET (QA-3)

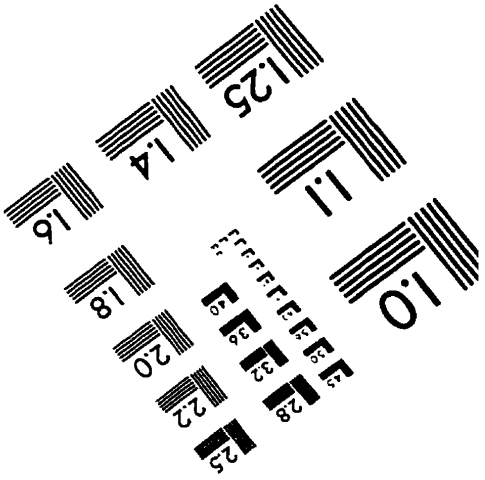












APPLIED IMAGE, Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989

© 1993, Applied Image, Inc., All Rights Reserved