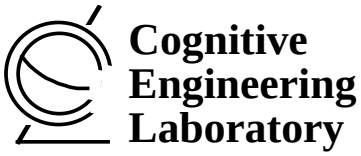
Applying Human Factors to the Design of Medical Equipment: Patient-Controlled Analgesia

Laura Lin, Racquel Isla, Karine Doniz, Heather Harkness, Kim J. Vicente, and D. John Doyle

CEL 95-06

Cognitive Engineering Laboratory University of Toronto Department of Industrial Engineering
4 Taddle Creek Rd. Toronto, Ontario, Canada M5S 1A4
phone: (416) 978-0881 email: benfica@ie.utoronto.ca fax: (416) 978-3453



Director: Kim J. Vicente, B.A.Sc., M.S., Ph.D.

The Cognitive Engineering Laboratory (CEL) at the University of Toronto (U of T) is located in the Department of Industrial Engineering, and is one of three laboratories that comprise the U of T Human Factors Research Group. The CEL began in 1992 and is primarily concerned with conducting basic and applied research on how to introduce information technology into complex work environments, with a particular emphasis on power plant control rooms. Professor Vicente's areas of expertise include advanced interface design principles, the study of expertise, and cognitive work analysis. Thus, the general mission of the CEL is to conduct principled investigations of the impact of information technology on human work so as to develop research findings that are both relevant and useful to industries in which such issues arise.

Current CEL Research Topics

The CEL has been funded by Atomic Energy of Canada, Ltd., Natural Sciences and Engineering Research Council, Defense and Civil Institute for Environmental Medicine, Japan Atomic Energy Research Institute, Westinghouse Electric Company, and Asea Brown Boveri Corporate Research - Heidelberg. The CEL also has collaborations and close contacts with the Mitsubishi Atomic Power Industries and Toshiba Nuclear Energy Laboratory. Current projects include:

- Studying the interaction between interface design and skill acquisition in process control systems.
- Understanding control strategy differences between people of various levels of expertise within the context of process control systems.
- Developing a better understanding of the design process so that human factors guidance can be presented in a way that will be effectively used by designers.
- Evaluating existing human factors handbooks.
- Developing advanced interfaces for computer-based anesthesiology equipment.

CEL Technical Reports

For more information about CEL, CEL technical reports, or graduate school at the University of Toronto, please contact the address printed on the front of this technical report, or send email to Dr. Kim J. Vicente at <benfica@ie.utoronto.ca>.

EXECUTIVE SUMMARY

Compared to other complex work environments (e.g., aviation, nuclear power), the medical domain has paid little attention to human factors. Lack of attention to user characteristics in the design of medical equipment leads to human errors, and potentially, life-threatening incidents. The primary hypothesis explored in this paper is that, by adopting human factors principles, the use of medical equipment can be made safer and more efficient. We have selected a commercially available patient-controlled analgesia (PCA) machine, the Abbott Lifecare 4100 PCA Plus II infuser, as a vehicle to test this hypothesis. A cognitive task analysis was conducted to understand the context in which PCA machines are used. This analysis, combined with a set of human factors design principles, led to a redesigned interface. The major changes in this new design were: a simpler dialogue structure with less steps, an overview of this structure showing where users are in the sequence, better feedback regarding the commands already entered, easier error recovery, and improved labels and messages. An experimental evaluation was conducted, comparing a software prototype of this new interface with a simulation of the existing interface for the Lifecare 4100. The results show that the new interface leads to significantly faster, less effortful, and more reliable performance. These findings have important implications for improving the design of other medical equipment.

INTRODUCTION

Human error plays a crucial role in the safety of medical equipment. For example, 60% of the deaths and serious injuries reported to the Medical Device Reporting system of the Food and Drug Administration's Center for Devices and Radiological Health have been attributed to operator error (Bogner, 1994). This figure is comparable to those reported in other domains, such as the nuclear, aviation, and maritime industries (Perrow, 1984). In-depth analyses of incidents in those domains have shown that, in the vast majority of these instances, the operators are not to blame. Rather, human errors can frequently be traced back to deficiencies in the design of the human-machine interface (broadly interpreted to include organizational issues as well). Because the interface was not designed with human capabilities and limitations in mind, operators are being placed in situations where the demands imposed on them are unrealistic from a psychological point of view. As a result, errors are inevitable.

The discipline of human factors, or ergonomics, deals with these issues by designing interfaces that take into account human capabilities and limitations. Lack of attention to human factors during design seriously jeopardizes safety by increasing the probability of human errors. In high hazard industries such as aviation and nuclear power, where the cost of an error can be enormous, human factors has become a significant design consideration (although there is still a great deal of room for improvement). In the medical domain, however, human factors has received very little attention (Bogner, 1994), despite the fact that nurses and doctors, like the operators in the industries just mentioned, are also dealing with complex situations where human lives are at stake.

By not taking into account operator characteristics, designers are creating deficient interfaces that will infrequently, but inevitably, lead to human errors and, potentially, life-threatening accidents. We are not alone in this opinion. For example, one researcher in the medical field has predicted that: “Our profession is likely to replicate the automation-related problems previously seen in other industries despite the warnings of the dedicated professionals, inside and outside of medicine” (Gaba, 1994, p. 62). Other researchers have begun to document cases where poor human factors has led to critical incidents (e.g., Cook, Woods, Howie, Horrow, & Gaba, 1992). The primary hypothesis explored in this paper is that, by adopting human factors principles, the use of medical equipment can be made safer and more efficient as well. We have selected patient-controlled analgesia (PCA) machines as a vehicle to illustrate these points. However, the design methods and conclusions presented here are believed to generalize to many different medical devices, not just PCA machines.

The remainder of this paper is organized as follows. First, a brief introduction to PCA will be provided. Next, the interface for the PCA machine that served as the focus of this study will be described. A cognitive task analysis was conducted to understand the context in which PCA machines are used. This analysis, combined with a set of human factors design principles, led to a redesigned interface. An experimental evaluation was conducted, comparing this new interface with the existing PCA interface. To anticipate, the results show that the new interface leads to faster, less effortful, and most importantly, more reliable performance. Implications for improving the design of other medical equipment will be drawn.

PATIENT-CONTROLLED ANALGESIA

Patient-controlled analgesia (PCA) has quickly become an effective and popular means of providing analgesia in patients with acute postoperative pain, as well as in other patients (White, 1985). This technology was developed in response to a well-recognized problem of pain undertreatment in hospitalized patients. For example, studies have shown that in the majority of postsurgical inpatients, parenteral narcotics given for moderate to severe pain provide inadequate analgesia (Angell, 1980). Also, nurses frequently under-administer narcotic analgesics postoperatively because of concerns about side-effects and narcotic addiction (Marks & Sachar, 1973). As a result of these attitudes, many patients receive only a small fraction of analgesic needed for adequate pain control (Sriwantanakul, et al., 1983). Complicating matters further is the fact that there is tremendous variability, both between patients and within an individual patient over time, with respect to the analgesic dose and the clinical response (Austin, Stapleton, & Mather, 1980). Part of this variability may be due to the fact that highly anxious patients have a lower pain tolerance (Scott, Clum, & Peoples, 1983). Another source of variability may be related to the finding that there is frequently a poor correlation between nursing assessments of patient pain and reports from the patients themselves (Teske, Dant, & Cleeland, 1983). These and other findings provide support for the idea of having patients control their own analgesic use, i.e., PCA.

From the patient's perspective, PCA is easily understood. Whenever they are in pain, or are planning to do something likely to be painful (such as getting out of bed), they push the PCA push-button (called an "owie" button in pediatric wards). If the patient is eligible to get the requested drug, as determined by the computer inside the pump, analgesic is given into the patient's IV over a few seconds. If there has not been enough

elapsed time from the time of the last dose (the "lockout period"), the computer denies the request.

The action of the PCA machine is governed by a computer program which, in turn, is based on a PCA prescription. PCA prescriptions are written into the patient's chart using a standardized order form. Floor nurses use these orders to program the PCA machine. Typically, this involves opening up the machine (a special key is needed) and entering a series of numbers and other data through a keypad. The key elements in a PCA prescription are illustrated in Table 1.

Safety

Safety problems related to PCA use generally fall into one of three groups (White, 1987): (1) operator errors (e.g., incorrectly entering a drug cartridge concentration of 1 mg/ml instead of 10 mg/ml, resulting in the computer administering a 1000% overdose), (2) patient errors (e.g., misunderstanding the PCA pump device), and (3) mechanical problems (e.g., "siphoning"). If human error plays as an important a role as we have suggested, then one would expect that operator errors are the most common type of problem. This is indeed the case (White, 1987). In this paper, we will focus on operator errors caused by incorrectly programming the PCA machine.

Why are human errors, such as programming mistakes, the most common type of problem with PCA machines? One potential reason is that nurses are often poorly trained in entering PCA prescriptions into the pump, either because of limited time and resources available for training, or because nurses may not have had to care for many patients on PCA pumps in recent weeks. A second reason could be that nurses are not selected based on an interest or an aptitude for mathematical or mechanical aptitudes, nor is technology a traditional part of nursing (Golonka, 1986). Thus, to require them to perform such tasks is to invite errors. A third, related reason for these errors is that the interfaces for PCA machines are poorly designed from a human factors perspective. That is, their interfaces are not based on a comprehensive understanding of the characteristics of their users and the context in which they are used. The first and second reasons suggest that intervention should be focused on training, whereas the third reasons suggests that the interfaces of PCA machines should be redesigned. Clearly, both training and design changes can help reduce human errors but how can one achieve the most amount of leverage?

Interestingly, with very few exceptions, the possibility of redesign is never acknowledged in the PCA literature. Instead, the most frequently mentioned solution to the problem of operator errors is to increase the amount of training (Smythe, 1992; Cohen, 1993). For example, White (1987) states: “If the nursing staff and patients are properly instructed in the use of the PCA device, these problems are preventable” (p. 82;

see also Smythe, 1992). While training can certainly help, there are very strong reasons to believe that training alone will not substantially reduce the frequency of operator error.

The "training solution" reflects a mechanistic attitude towards human performance which views people as automatons that can repeatedly perform the same task in the same

TABLE 1. Elements of a PCA prescription.

- Route of administration (usually IV)
- Loading dose (drug quantity) and interval of administration (e.g., morphine, 4 mg, IV bolus)
- Incremental dose administered at patient's discretion (e.g., morphine, 1 mg, IV)
- Lock-out interval - time interval after last dose during which a patient request is denied (e.g., eight minutes)
- Cumulative dose limit (e.g., maximum dose for a four hour period; morphine, 20 mg, IV)
- Background infusion rate (e.g., morphine, 1 mg/hr, IV) [Many physicians set this to zero for safety reasons]

manner, once given the appropriate skills (Ostrander, 1986). In contrast, everything we know about human performance points to a very different picture. Human performance is subject to inherent variability, which means that mistakes will periodically be made, no matter how well trained and well intentioned the operator. This point has been acknowledged by some in the medical community:

> Physicians and nurses need to accept the notion that error is an inevitable accompaniment of the human condition, even among conscientious professionals with high standards. Errors must be accepted as evidence of system flaws not character flaws. Until and unless that happens, it is unlikely that any substantial progress will be made in reducing medical errors (Leape, 1994, p. 1857).

It is also important to note that human performance is greatly influenced by the context in which behavior takes place. If a nurse is in a hurry and is programming a PCA machine in the dark so as not to wake up a patient, then errors are more likely to occur if the interface was not designed for such a context, regardless of how much training was provided. Similarly, human performance can be degraded by what are known as performance-shaping factors, such as fatigue (e.g., long shifts) and circadian rhythms (e.g., performing a task at 4 am). The "training solution" fails to take all of these factors

into account, and thus its impact on reducing errors is limited. A complementary approach that acknowledges and accommodates the variability of human performance is needed to minimize the impact of factors that lead to errors, and to allow operators to easily detect and recover from the errors that will inevitably occur.

Rather than increasing the training required to use a poorly designed interface, one can try to redesign the interface to make it simpler to use. Nurses' responsibility is to care for patients. Programming a PCA machine is a means to that end, and should not be an end in itself. Thus, the task of dealing with the machine should be made as transparent as possible (Vicente & Rasmussen, 1992) so that nurses can focus on the task at hand, i.e., patient comfort. Achieving this design goal requires an in-depth knowledge of human factors principles. But since medical equipment has not traditionally been based on such principles, one would expect that existing equipment does not satisfy the requirements outlined above. It is not surprising then, that several cases of PCA machine mis-programming have been reported in the medical literature (White, 1987; Cohen, 1993). This finding suggests that ease of use has not been an important criterion in the evaluation of PCA machines.

Previous Research

Some comparative evaluations of PCA machines have been reported in the literature. For example, Sawaki, Parker, & White (1992) evaluated the usage of 5 different PCA machines in postsurgical wards, using questionnaires given to nurses and patients. While there are several positive features of this evaluation (e.g., it was conducted in a clinical setting; the sample size was large), there are also some limitations with regard to the concerns of the present paper. First, ease of use was not the primary focus of the study. Only a few questions addressed usability, and even these were vaguely defined. For example, one of the questions was: How easy was it to learn how to program and use each of these devices? Because of the absence of the definition of an explicit criterion, such questions are likely to lead unreliable responses. Second, this study only measured the opinions of users, not actual performance. While subjective ratings are a useful source of data, it is well known that subjective opinion need not correlate with objective performance (Rouse, 1984). It is possible for subjects to express a preference for one device, when performance data show no difference or even an advantage for a different device. For both of these reasons, the conclusions one can derive from this study are very limited.

Another comparative PCA machine evaluation focused on mechanical properties, and therefore conducted no formal evaluation of ease of use (Owen, Glavin, Reekie, & Trew, 1986). Nevertheless, the authors claimed that both machines tested were "user

friendly". The fact that most problems experienced with PCA machines are related to human errors suggests that the authors' conjecture is unlikely to be correct.

More recently, Ilsley et al. (1994) have developed an automated system to conduct standardized evaluations of PCA machines. However, the focus of this system is on measuring the reliability and accuracy of the amount of analgesia delivered by the PCA machine. While this is certainly a significant measure of safety, it does not address problems due to inadequate interface design.

In summary, the comparative evaluations of PCA machines that have been reported have focused primarily on mechanical properties. It is important to note that this is a necessary but not sufficient condition for overall safety. A PCA machine that reliably delivers the exact amount of analgesia upon demand can still lead to catastrophe if it has a poorly designed interface that induces programming errors. Clearly, there is a strong unexplored need for evaluating PCA machines from a human factors perspective.

This paper addresses this need by presenting a design and evaluation of a PCA interface based on human factors principles. Before this interface and experiment are described, the current interface that served as the focus for this work will be described.

ABBOTT LIFECARE 4100 PCA PLUS II: CURRENT INTERFACE

The Abbott Lifecare 4100 PCA Plus II infusion pump was selected as the object of this research, primarily because it is widely used at Toronto General Hospital (TGH). This section provides a brief description of this device, particularly its interface.

Operating Parameters

Preparing the PCA machine for therapy involves programming the machine with specific parameter settings. The operating parameters are prescribed by a physician on a PCA order form and given to a nurse to program into the machine before the PCA therapy begins. As already mentioned, the parameters include: mode, dose, rate of infusion, drug concentration, lockout interval, and four hour limit (see Table 1).

The Abbott Lifecare 4100 PCA Plus II Infusion pump is capable of administering an analgesic in a discrete or continuous mode or both. In the PCA mode (discrete delivery), the patient can request a dose of analgesic by pressing a button on a handset. Depending on the time that has elapsed since the last request and the cumulative amount delivered since commencement of therapy, the PCA machine delivers a dose through an intravenous line. The lockout period specifies the minimum time period between patient requests before the machine will comply with the request. The 4 hour limit specifies the total amount of drug a patient can receive within 4 hours. In the CONTINUOUS mode, analgesic is administered at a continuous rate without PCA doses. This baseline infusion

can be combined with self-administered discrete doses (PCA mode), and is termed PCA+CONTINUOUS mode.

Physical Interface/Programming

The interface of the Abbott Lifecare 4100 is shown in Figure 1. The functional features of this PCA machine include the touch switch panel and the liquid crystal display (LCD). The LCD provides instructional messages which prompt the programmer (usually, a nurse) to enter the parameter settings. These are entered using the touch switches on the control panel. Each sequential screen is composed of a short message to which the user responds with "yes" or "no". The user is also asked to specify values by either accepting a default value, or incrementing/decrementing it to the desired value. "Enter" advances the user to the next parameter. The Lifecare 4100 control panel consists of twelve (soft) touch switches, most of which have multiple functions. The control panel is visually grouped, however, in a manner that suggests there are only six touch switches; as shown in Figure 1, each physical touch switch has two labels (two soft touch switches).

Error Recovery

During the process of programming, several types of errors may occur. An incorrect setting may be entered due to inexperience, haste, or any number of other reasons. This will require the user to detect and recover from the error. This includes correcting errors during the programming sequence and after the programming sequence is completed. With the current design, changing a previously set parameter while in the programming mode requires the user to scroll backwards through the sequence to locate the parameter to be changed. This is different from changing a parameter after programming is complete; in this case, the user is prompted to indicate which parameter he/she wants to change.

The Lifecare PCA infuser allows for error recovery tasks by provision of a "REVIEW/CHANGE" touch switch. When pressed during the programming sequence, this touch switch brings the user to the previous screen to allow changes to be made. The programming sequence then resumes. Once the entire programming sequence has been completed and the user wishes to make changes, pressing the "REVIEW/CHANGE" touch switch will prompt the user as to which setting he/she wishes to amend.

COGNITIVE TASK ANALYSIS

Redesigning an interface to maintain functionality and conform to user needs, requires a thorough understanding of both the system and the demands it places on the

user. To acquire this understanding, cognitive task analysis (CTA) was employed as a systematic method of studying the psychological requirements of programming the PCA machine. Establishing an understanding of the programming task also served to guide observations of the machine in context. The CTA methodology that was used was based on previous research (Charante, Cook, Woods, Yue, & Howie, 1992; Yue, Woods, & Cook, 1992; Cook, Potter, Woods, & McDonald, 1991). The method will be briefly described first, followed by the results of the analysis.

Data Collection Methods

The methods used to collect the information necessary for the CTA included: bench tests (i.e., hands on expert review), field observations, and comments from recovery room nurses based on a presentation of the proposed interface. A detailed description of how these methods were employed, and the insights they led to can be found in Lin & Isla (1993) and Doniz & Harkness (1994), so only a brief overview will be provided here.

The bench tests aided in identifying the characteristics of the device which make its operation prone to errors: poor assembly mechanisms, faulty operations, and device failures. The goal was to analyze any system failures or problems so that modifications to the current interface could justifiably prevent future problems or malfunctions. A summary of the bench tests results can be found in Isla and Lin (1993). These results lead to a thorough mapping of internal system structure (see below), which was instrumental in identifying interface design deficiencies.

The field study took place in the recovery room of the TGH. This involved observing and interviewing nurses responsible for programming the PCA machines. The study was carried out over eight visits, permitting us to interview nine different nurses, and to survey the programming of thirty PCA machines. Observing the machine and its users in context drew attention to the added demands that the environment placed on the users. The distractions and noise in the recovery room during peak times appeared to increase the nurses' mental workload. This underscores the gravity of the problems encountered during the bench tests, inasmuch as the bench tests were performed under comparatively favourable conditions.

Nurses were questioned in the interviews on both the merits and shortcomings of the current PCA machine. Their responses reiterated the problem areas identified in the bench tests, namely the incompatible defaults, complex or lengthy programming procedures, tedious error recovery process, and indistinguishable touch switches. Doniz & Harkness (1994) provide a summary of these and additional grievances recorded in the field study. A large proportion of the amendments proposed to address these problems were incorporated in the redesigned interface.

Development of a new design followed the bench tests and field study. The proposed design was demonstrated to TGH nurses during an oral presentation. The purpose of the presentation was twofold: first, to solicit feedback which would aid in further refining the design; and second, to foster user acceptance of the new design ideas. Involving the intended users of the device in the design process thereby played a crucial role in the redesign of the PCA interface.

During the presentation, nurses were shown the programming sequence for setting the machine up for PCA+Continuous operation (the most complex mode) using the new proposed interface. Throughout the presentation, questions and comments were made, bringing to light new issues and recommendations. Several design changes were directly instigated by the feedback from the presentation (see Doniz & Harkness, 1994). In general, the presentation and feedback provided a preliminary assessment of the redesigned interface and a compilation of practical suggestions from end users. A greater understanding was gained of what the nurses would like to see altered and why these changes would be beneficial. In addition, the presentation to the nurses, the intended users of the device, helped to promote their acceptance of the proposed design ideas.

The remainder of this section presents the results of the CTA.

Results

The bench testing led to a state transition diagram analysis which summarizes the structure of the tasks that users are required to perform with the existing interface. State transitions at three levels of detail were examined (see Isla & Lin, 1993):

1) general flow of activities (function flow diagram of the "subtasks" performed to complete the task of programming - see Figure 2),

2) decisions and actions required of each subtask (refer to Figures 3 to 10), and

3) detailed mapping of sequential LCD messages and user input.

The state transitions at the second level of detail proved to be the most instrumental in the development of the new design; the general structure of many subtasks in the programming sequence were unnecessarily complex (refer to Figures 3 to 10). To take but one example, the subtask of choosing the mode, shown in Figure 6, involves 3 separate decisions. Not only is this procedure time consuming, but it also neglects the need for a global view of the decision. With the current design, a choice between three related alternatives is treated as three unrelated choices. Other examples of unnecessarily complex task structure were also identified (see Isla & Lin, 1993). These observations anticipate the need for significant changes to the interface.

Having determined the structure of the dialogue that users are required to follow, an analysis of the information requirements associated with each dialogue step was

performed. This aspect of CTA identifies the information or knowledge needed to perform a task, and therefore, the information that should be included in an interface to perform the previously identified tasks.

At this point, it is worthwhile to introduce the distinction between deep control and surface control of a system (Vicente & Rasmussen, 1992). If the user relies on perceptual cues provided by the interface in order to decide what actions are appropriate, then they are relying on surface control. In contrast, if they have to go beyond the perceptual features and instead have to rely on their conceptual understanding (i.e., a “deep” model) of the internal structure of the device in order to determine what to do, then they are relying on deep control. Although both control modes can be useful, surface control tends to be faster and less effortful 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 on working memory load. Deep control is slower and more effortful because it requires users to engage in analytical reasoning. Furthermore, there is a great deal of evidence to indicate that people have a tendency to engage in surface control, even when it is not appropriate to do so (Vicente & Rasmussen, 1992). Thus, an important question is: How well does the current interface support surface control?

Analysis of the Abbott Lifecare PCA Plus II machine showed that surface control is supported by the sequential display of instructional messages. The user is guided through the programming sequence allowing the user to "perceive and act" and thus reducing dependency on working memory. The effectiveness of this programming cue, however, is less than optimal because other potential cues are neglected. One such cue which was identified as needing improvement was the spatial organization of the touch switch panel (see Figure 1). Lack of a meaningful grouping (i.e. procedural order) and the 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: lack of external representation of the system structure. Although the user is guided through a sequence of tasks (Figure 2), 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. Finally, an inconsistent cue can also pose cognitive demands on working memory. Bench tests yielded several inconsistencies in touch switch labeling. The functions of the touch switches are misrepresented by the wording of their labels. For instance, the RESET/ START touch switch does not reset the machine, it stops and starts infusion; and

REVIEW/ CHANGE does not review the settings, the HISTORY touch switch performs this function. Touch switch functions and labels are further discussed in the following section.

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. Two groups of tasks were analyzed: programming tasks and error recovery tasks. Programming tasks include: purging, administering loading dose, selecting drug concentration, selecting mode, setting lockout interval, setting continuous rate and setting 4 hr. limit. Error recovery tasks include: changing previously set parameters during programming mode and after programming completed.

Figures 11 to 13 show the information that the user is required to know for decision making in order to accomplish each subtask. This was compared with the information that was actually provided on the interface. This analysis showed that the information required to perform error recovery tasks is not supported effectively by the guidance messages (insufficient cues to aid user in performing the task). Provision of information for the programming tasks is comparatively more adequate. Thus, in performing the error recovery tasks, the user must rely on his/her own knowledge of the system. Due to the lack of external representation of the system structure, the user's interpretation of the system may be incomplete or erroneous, making the task susceptible to error or excessive time consumption. This motivated the need for a new interface which provides the appropriate external cues. To anticipate, the redesign should simplify tasks most dependent on deep control (error recovery tasks in particular), reduce cognitive demands, and encourage effective surface control of the system.

In summary, conducting a CTA provided us with a solid understanding of the demands that this task imposes on users, the context in which these tasks must be performed, as well as an indication of how the current interface does not provide the resources to help users effectively and reliably deal with task demands. The deficiencies of the current interface were analyzed in more detail by determining how well they measured up to human factors design principles.

HUMAN FACTORS DESIGN PRINCIPLES: ANALYSIS OF CURRENT INTERFACE

This section describes how human factors design principles were adopted in evaluating and subsequently redesigning the interface to the Abbott Lifecare PCA Plus II. Extensive use was made of several existing human-computer interaction principles (Yue, Woods, & Cook, 1992; Cook, Potter, Woods, & McDonald, 1991), including:

1. Provide users with prompt, salient feedback after each action
2. Make the functions of the various controls clear and obvious
3. Make the displayed messages easy to understand
4. Minimize the load on the users' memory
5. Provide users with shortcuts to increase efficiency
6. Provide clearly marked exits for the user to leave the system.

These principles, as they apply to the Abbott PCA machine, will now be discussed in turn.

Provide Feedback

The user should be given feedback when changes occur in the behaviour of the device (Cook, Woods, & Yue, 1992). Feedback is used to tell the user what operation has been executed, as well as the results of the operation. In this way, the user is always aware of the status of the system. Feedback is especially critical in both detecting and recovering from errors.

The Abbott PCA machine provides a limited amount of feedback to the user through a very small LCD screen. For example, when a loading dose is being administered, a message is displayed indicating that the infusion is in progress. While programming, however, the user receives no feedback on which parameters have already been set or how many are left to program. This can be quite frustrating for a new user of the device because programming is a fairly lengthy procedure. The user may not feel in control of the system and will rely on it to guide him/her through the programming sequence.

Make Functions of Controls Clear

A control which is used to carry out several functions is called multifunctional. With multifunctional controls, the user must remember what operations they perform in different contexts and also when they can be used. This increases the user's mental working load and chances of making errors (Cook, Woods, & Yue, 1992).

Several touch switches on the PCA machine are multifunctional. These touch switches can be grouped into three categories. A touch switch that performs two similar

(or identical) functions belong to the first category. Touch switches that fall under this category include the YES/ENTER, REVIEW/CHANGE, and OFF/RECHG touch switches. A touch switch that performs two opposite functions, such as RESET/START, belongs to the second category. The third category consists of touch switches that perform unrelated functions (SILENCE/NO).

A user who relies solely on surface control may find that the labels on different touch switches have similar meanings. For example, the user must use 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-start (retains parameter settings, if the PCA machine has been off for less than an hour) or a cold-start (clears parameter settings, if the PCA machine has been off for more than an hour). The RESET/START touch switch is used to begin or to stop infusion in the Continuous and PCA+Continuous modes. It also restarts infusion after the machine has been temporarily turned off. However, it is not used to reset parameters as the name implies.

The REVIEW/CHANGE and HISTORY labels appear to have similar meanings which may lead to confusion over their actual functions. REVIEW/CHANGE is used to revert to the previous screen during programming (to change or view a setting) or to reset a parameter once programming is completed. HISTORY is used to display the history of all the doses delivered, as well as the event log (a chronological record of events occurring during infusion). It is also used to display currently set parameters at the end of the programming sequence. Lack of intuitive labeling forces the user to commit to memory that HISTORY, and not REVIEW/CHANGE, must be pressed to view the parameter settings.

The REVIEW/CHANGE touch switch takes the user either forward or backward in the programming sequence, depending on when it is pressed. During programming, it can be used to change a previous setting. Therefore, to go to a previously set parameter, the user presses REVIEW/CHANGE in order to scroll backwards through the programming sequence until he/she reaches the desired parameter. On the other hand, if programming has been completed (all parameters have been set), then REVIEW/CHANGE takes the user forward through a series of displays similar to the programming sequence, and allows the user to make changes to any of the settings.

An inconsistency was found with the first REVIEW/CHANGE function mentioned above. It is unclear how many steps backward will result from one press of the touch switch during programming. The following two examples illustrate the inconsistency found with this REVIEW/CHANGE function:

Example 1

The user is presently setting the drug concentration and reads the message:

Rx CONCENTRATION
TO SET USE
0.1 MG/ML
PRESS ENTER

The user realizes that he/she must set the concentration in the units μG/ML, not MG/ML. He/she presses REVIEW/CHANGE and is returned to the message:

DRUG
Rx CONCENTRATION
MILLIGRAM / ML ?
YES OR NO

and proceeds to change the units to micrograms/ml.

Example 2

The user has already set the loading dose and is about to infuse it:

LOADING DOSE
TO INFUSE
5.0 MG
PRESS LOAD DOSE

but realizes that the value has been incorrectly set and wants to return to the previous screen which is:

LOADING DOSE
TO SET USE
XX.X MG
PRESS ENTER

The user simply wishes to change the amount he/she has set for the infusion. However, after pressing REVIEW/CHANGE the user is taken back to the screen used to set the drug concentration:

DRUG
Rx MORPHINE
1 MG/ML ?
YES OR NO

In the first example, REVIEW/CHANGE takes the user back to a display which permits him/her to set the drug concentration in different units. The user remains in the same subtask (select drug concentration). However, in the second example,

REVIEW/CHANGE does not permit the user to change the loading dose which is performed in the preceding screen. Instead, it removes him/her from the active subtask (administering a loading dose) and returns him/her to a completely different subtask (selecting drug concentration). These examples show the need for improved consistency in touch switch functioning.

The LOAD DOSE touch switch performs two functions. Its first function is to bring the system to the loading dose subtask so that the user can set the amount to be administered. Once the amount has been set, the LOAD DOSE touch switch is pressed again to start the infusion. Since users rarely administer a loading dose to a patient, making this touch switch multifunctional may cause confusion. The touch switch is used so seldomly that users may easily forget that pressing the touch switch will allow them to set the loading dose and in the other case begin infusion of the loading dose.

Make Displayed Messages Clearly Understandable

Displayed messages should tell the user what is detected by the device. They serve to help the user 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 of the current machine are awkwardly worded and sometimes even ambiguous. For instance, the following message may be easily misinterpreted:

ADMINISTER
LOADING DOSE
NOW?
YES OR NO

Pressing YES/ENTER to answer this query will not begin infusion of a loading dose, as the wording suggests. The message is only asking if the user intends to infuse a loading dose so it may then proceed to a screen to allow the user to set the dose.

Another example of an awkwardly worded display is the message that appears when the REVIEW/CHANGE touch switch is pressed at the end of the programming sequence:

CHANGE?
ANY SETTING:/
SELECT MODE
YES OR NO

The message is intended to query the user as to whether or not he/she would like to change the mode or any of the settings. It is not apparent from this display that the settings are dependent on the mode: which parameters must be set depends on the mode

one chooses. The wording of the display also implies that a mode has not yet been selected.

Yet another awkwardly worded message is the one which asks the user:

4 HOUR DOSE LIMIT
SET?
YES OR NO

This query 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. To recover from this error, the user must press the REVIEW/CHANGE touch switch and go through the entire sequence until he/she is prompted again to set a 4 hour dose limit.

Minimize User Memory Load

The interface should minimize the amount of information that the user must store in short term memory to operate the device. Perceptual cues should be provided to relax the user's reliance on memory. 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).

Since the user is led through the programming procedure by the sequential messages, it is not necessary for the user to remember what parameters must be programmed in each of the three modes with the current interface. For example, the dose and lockout interval parameters must be set in the PCA mode. In the CONTINUOUS mode, however, they are not part of the operating parameters; a continuous rate and 4 hr limit are set instead. The interface dialogue ensures that the relevant displays are brought up, as a function of mode.

Another example of how the current interface reduces memory load occurs when the user has finished programming the device and presses REVIEW/CHANGE to alter one of the settings. After changing the desired parameter, the user is forced to proceed through all the subsequent settings in the programming sequence. This is important when a different mode is selected since the parameters that need to be entered depend on the mode chosen. Since the user is required to program all the parameters for the newly selected mode before exiting the error recovery task, they do not have to remember which additional parameters must be reprogrammed.

Provide Shortcuts

When the user is comfortable with the system, shortcuts in the programming procedure allow him/her to progress through it much more quickly. Shortcuts should

only eliminate the steps that are unnecessary in a particular task (Molich & Nielsen, 1990), and the device should operate as normal afterwards.

One method of providing shortcuts is to set appropriate machine defaults. The default values provided on the current PCA interface design conflict with the standard operating values used at the TGH. One example of a conflicting default value is the drug concentration. The most widely used concentration at this hospital is 2.0 mg/ml. Programming this parameter requires the user to toggle through four screens of default values before being able to enter this concentration. Making suitable default settings available would eliminate many of these unnecessary programming steps.

Provide Clearly Marked Exits

Exits permit the user to leave a system or subsystem. By providing these exits, the user can prevent errors from occurring (Molich & Nielsen, 1990). He/she can leave the system without having affected any settings in the device.

There are several programming tasks in which the user is not provided with an adequate exit. When setting the concentration, the user must select one of a series of screens with different drug concentrations or choose to set the concentration manually. If, for example, the user accidentally skips the desired concentration setting, he/she cannot use REVIEW/CHANGE to go back to it. Instead, the user must toggle through all the screens (in a loop) and start again from the beginning (see Figure 2). Similarly, in the sequence for selecting the mode, the user is prompted sequentially by the LCD messages to select one of the three modes. These screens occur in a loop without an exit.

In summary, several areas where the current interface can be improved were identified. A summary of the problems that were identified from the application of design guidelines and the CTA is provided in Table 2, along with the effects of each problem, its severity, and the proposed solution. These proposed solutions were integrated into a redesigned interface that will be described next.

TABLE 2. 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 recover 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 user	Med	- provide labels that are user-driven
Misleading & confusing messages	- hard to interpret - slows down user	Med	- provide messages that are user-driven
Review/Change inconsistent	- disorienting & confusing	Low	- make consistent
Defaults not appropriate	- many steps - takes time	Low	- use TGH standard

REDESIGNED INTERFACE

As illustrated in Table 2, the bench tests and field study provided the necessary evidence to motivate changes to the PCA machine's interface. Results from the bench tests showed that the system structure is not adequately visible. Lack of external cues forces users to rely on deep control to manipulate the system, and poor communication of information in screen messages and touch switch labels cause confusion over the perceived system structure. The context of machine usage also necessitates changes to the interface. As observed in the field study, nurses often perform the task of programming the PCA machine under a demanding and stressful environment. The physical and cognitive demands imposed on the user by a poor interface can increase the chance of error especially when compounded by an intense environment. This accentuates the need for a PCA interface which minimizes mental effort and the time to perform the programming task. Thus, the redesigned interface was intended to reduce the likelihood of programming errors and to facilitate the recovery from errors when they do occur.

It is important to note that the proposed interface maintains roughly similar technological constraints since modifications were made to the current interface (as opposed to the conception of an entirely new physical format). This approach served to prevent extensive deviation from current manufacturing costs of the PCA machine.

Physical Interface

The physical design of the new interface is shown in Figure 14. It consists of an enlarged LCD screen to house an indicator field and menu display, in addition to the message field. The indicator field at the top of the screen displays the programming task that is being performed. Below this display field is a status display of the three stages of programming: concentration, mode, and settings. As the program leads the user through each stage, the corresponding item is highlighted by an indicator box. The redundancy of having two sections of the LCD dedicated to tracing the programming sequence ensures that the system structure is constantly visible. This also allows for a global view of the programming sequence. With the existing design, each programming task lacks context, giving the user only a highly limited view of the system structure at each programming stage. The confusion caused by this lack of awareness is minimized in the new design by the sequence indicators, which show where the users are, where they have been, and where they are going in the programming sequence.

Figure 15 lists the touch switch functions on the redesigned interface. The touch switch control panel on the new interface has been modified in several ways. First, the number of touch switches was reduced. Because of changes to the programming sequence, the PURGE SYSTEM touch switch was eliminated. The purging task now prompts the user to press START to begin purging the system, and to press STOP once the flow is seen. Second, the double touch switches of the current machine have been separated into single touch switches on the new interface. Also, grouping of the touch switches separates the ones used for programming from the special-purpose touch switches (BOLUS DOSE, REVIEW, HISTORY, and ON/OFF). The upper group contains the programming touch switches and is spatially organized in a logical manner: YES above NO, START above STOP, and the up arrow ( ) above the down arrow(¬). As the labels imply, each touch switch performs only one function (one physical touch switch represents one soft touch switch). The same is true for the special-purpose touch switches in the bottom grouping, except for the ON/OFF touch switch which serves two related purposes.

The functions of the touch switches differ slightly from the current interfaces due to modifications to the system structure. The BOLUS DOSE on the new design replaces the LOAD DOSE touch switch. This new label more accurately describes its function, as indicated by the nurses at TGH. Since a bolus dose can be administered at any time during and after programming (provided that the concentration has been set), the prompt which asks the user whether to administer one has been eliminated from the programming sequence. This makes BOLUS DOSE a special-purpose touch switch, simplifying the programming sequence and reducing the number of programming touch switches.

REVIEW/CHANGE has been relabelled REVIEW. It maintains the same purpose, allowing the user to review the settings and to make changes where necessary. When pressed during programming, it performs identically as the current machine but with greater consistency. The only modification to the error recovery task is that the sequence indicators on the new design can inform the user of their location within the sequence.

The HISTORY touch switch performs the same functions as on the current machine with the added function of providing an updated summary of settings at any stage of programming. This supplemental function enhances the user's sense of control over the system and aids in detection of errors. As an additional preventive measure against overlooking incorrect settings, a summary of the programmed settings is automatically displayed before the user locks the door and hooks the machine up to the patient.

The modification of touch switches and their functions are intended to produce a more efficient interface for communication of information. By confining the user

response to a small set of clear alternatives (yes/no, start/stop, up/down), it was expected that decision making would be simpler. Also, the organization of the touch switches into two distinct groupings should provide for quicker response execution.

System Structure

Accompanying the changes to touch switch functions are modifications to the programming sequence. The function flow diagram for the redesigned interface is shown in Figure 16 (Figures 17 to 24 show the decisions and actions required for each subtask). To fulfill the objective of reducing programming time, the sequence was simplified by minimizing the number of message display screens. The first approach used to accomplish this simplification was to incorporate a menu system for the stages that included choices. For example, when selecting the mode, the three options are listed on the screen simultaneously and the user is prompted to use the ¯ or ¬ touch switch to select the desired mode. This achieves two goals: it decreases the number of screens the user must toggle through (thus reducing programming time), and it provides the user with a more global view by displaying all the decision options on a single screen.

The second strategy employed in simplifying programming was to provide the appropriate default settings at every stage. The most common settings and values used at TGH, shown in Table 3, were employed as the system defaults to avoid unnecessary programming steps.

Finally, message screens for tasks which were extraneous to programming (e.g., bolus dose) or fixed in the protocol of machine setup (purging and setting 4 hr limit) were either condensed or eliminated from the programming sequence. For instance, in the previous interface, the user is queried as to whether or not a bolus dose is to be administered. Since the user has the option to use the BOLUS DOSE touch switch at any time, it is questionable to place this message in the sequence or even to justify placing it at one location within the sequence as opposed to another. The prompt which asks the user: "Set 4 hour limit? yes or no" was also eliminated because the 4 hour limit must always be set for all modes. The sequence on the new design will go directly to instructing the user to input the limit. Lastly, the purging task was revised to employ less message displays. By granting the user control over when to stop purging, the task should become faster.

The combined effect of these minor changes to the programming sequence is a profound transformation in the perceived complexity of the system structure. A comparison of the maximum number of screens shows the new design to be more efficient. The previous design 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 this reduction in number of programming screens, but also because of the added visibility of the system provided by the global and local status displays (sequence indicators and decision options, respectively).

TABLE 3. Typical parameter settings used by nurses at TGH that serves as the default settings in the new interface.

PARAMETER	DEFAULT
Concentration Units	mg/ml
Drug Concentration	2.0 mg/ml
Mode	PCA
PCA Dose	1.0 mg
Lockout Interval	6.0 min
Continuous Rate	1.0 mg/ml
4 Hour Limit	30.0 mg

EXPERIMENTAL EVALUATION

This section describes an empirical evaluation comparing the existing and redesigned PCA interfaces. Previous studies assessing the usability of PCA machines have been based solely on subjective measures, such as questionnaires. As it is difficult to make conclusive statements about performance from these measures alone, direct performance measures are also needed to reliably determine ease of use. In this study, three primary measures were considered: task completion time, number of errors, and subjective ratings of mental workload. It was expected that the redesigned interface would lead to better performance on all three measures, compared to the existing interface.

Method

Subjects. The principle focus of the experiment was to test both interfaces on novice users. Accordingly, the selection of subjects for the experiment was based on two important criteria. First, subjects had to have a background that was representative of the background of professional nurses. Second, subjects also had to have no experience with the current Abbott Lifecare Plus II interface, so as to eliminate any potential transfer

effects. Thus, subjects were university students with medical professional backgrounds, including nursing, pharmacy, and rehabilitation medicine. A total of 24 participants volunteered to partake in the experiment, ranging in age from 18 to 45.

Experimental design. A 2 x 3 x 2 mixed design was adopted for the experiment with Interface (Old vs. New) and Programming Task (PCA, Continuous, and PCA + Continuous) as within-subjects factors, and Interface Order (New First vs. Old First) as a between-subjects factor. The order of presentation of the interface and the tasks were counterbalanced. Thus, each subject performed a total of 6 trials.

Programming tasks. For each trial, the subjects were given a copy of the TGH doctor's order form filled in with the requested values to be programmed. The values requested depended on the task being performed. The subjects then proceeded to program the machine by following the directions presented to them on the respective interface.

Apparatus. A graphical simulation of both interface designs was developed using the Toolbook Openscript software package. The simulations ran on an IBM-compatible PC equipped with a mouse and a MegaImage colour monitor. Input data for the programming task were provided to subjects on standard PCA order forms used at the TGH.

Procedure. First, the purpose of the experiment was explained to the subject. Also, background information on the PCA machine and the tasks that subjects would be performing were explained. The subject was then provided with six PCA order sheets and was asked to begin the programming tasks. Subjects then proceeded through each stage, programming the required values. After each trial, subjects completed a mental workload rating scale (see below), and provided any comments they might have had on the preceding trial. An experimenter was present during the entire experiment. At the end of the experiment, informal comments were solicited from subjects to determine which of the two interfaces they preferred.

Performance measures. There were three dependent variables. First, the total time to successfully complete each trial was recorded. Second, the number of errors made in completing each trial was also recorded. Both of these measures were collected by the experimenter, who observed the subjects as they completed the required tasks. Third, subjective ratings of mental workload were also collected from subjects. The NASA-TLX method, a well accepted measure of subjective mental workload, was used for this purpose (Wickens, 1992).

Results

The results for task completion time will be examined first. A three-way ANOVA with Interface, Task, and Order as the main factors was conducted. The Interface effect was highly significant ($F(1,22) = 54.71$, $p < 0.0001$). As shown in Figure 25, the mean time with the New interface was 3.9 minutes, whereas that with the New interface was 5.2 minutes, 33% slower. However, this main effect can only be meaningfully interpreted within the context of the Order x Interface interaction, which was also statistically significant ($F(1,22) = 13.90$, $p < 0.0012$). As shown in Figure 26, the mean programming time on the new interface is always faster, but there is an asymmetrical transfer effect from one interface to the other. There are several ways to look at this effect. First, transferring from the New to the Old interface causes a larger performance change than going from the Old to the New (differences of +1.9 minutes and -0.6 minutes, respectively). Second, those subjects who have already had some experience at the task with the Old interface are slower with the New interface than subjects who are doing the task for the first time with the New interface. This suggests that subjects who have been exposed to the Old interface acquire behaviours that do not allow them to fully exploit the benefits of the New interface. Third, those subjects who have already had some experience at the task with the New interface are much slower with the Old interface than subjects who are doing the task for the first time with the Old interface. This suggests that subjects who have been exposed to the benefits of the New interface then have a more difficult time compensating for the deficiencies of the Old interface. Taken together, these results show the general superiority of the New interface, but they also point to the value of being exposed to such an interface right from the start.

The mental workload ratings were transformed into percentages, and a similar ANOVA was conducted. Again, the main effect for Interface was significant ($F(1,22) = 16.09$, $p < 0.0006$). As shown in Figure 27, the mean workload rating for the New interface was 10.8%, whereas that for the Old interface was more than twice as high, 23.8%. No other effects were statistically significant. Thus, the New interface led to significantly less workload than the Old interface.

A non-parametric statistical analysis was conducted on the number of errors. As shown in Figure 28, the New interface led to 28 errors, whereas the Old interface lead to a total of 43 errors, a difference of 54%. A χ^2 test indicated that this difference was statistically significant ($\chi^2(1) = 3.1619$, $p < 0.05$, one-tailed). Thus, the New interface led to significantly more reliable performance than the Old interface.

The informal comments expressed by subjects at the end of experiment were consistent with the results just presented. Twenty-three out of the twenty-four

participants ($p < 0.001$, binomial test) expressed a strong preference for the New interface design. The one participant who did not prefer the new design recommended a completely different type of interface, similar to the doctor's order sheet, and therefore did not favour the old design either. A summary of the most common comments and suggestions provided by subjects can be found in Doniz & Harkness (1994).

Discussion

This experimental evaluation sought to compare the performance obtained with the existing and redesigned interfaces for the Abbott Lifecare PCA Plus II. The informal comments obtained from subjects clearly show that the new interface was preferred by the participants. While it is certainly important to consider how the users felt about the interface, it is necessary to complement this anecdotal evidence with the results of the performance measures. All three of the other measures showed a statistically significant advantage for the New interface; the redesigned interface lead to faster programming times, lower ratings of mental workload, and fewer programming errors. These results strongly support the hypotheses that motivated the study.

CONCLUSIONS

This paper was motivated by the fact that lack of attention to human factors in the design of medical equipment can lead to deficient interfaces that induce human errors with potentially life-threatening consequences. The design of PCA machines was chosen as a focus for research, and several reports of mishaps associated with the use of PCA machines due to human error were cited. It is important to point out, however, that these problems are not specific to PCA machines. Incidents with other types of medical devices that can be traced back to poor interface design have also been reported (e.g., Cook, Woods, Howie, Horrow, & Gaba, 1992). The fact that the degree of automation in medical equipment is rapidly increasing suggests that such errors are likely to increase in the future. Clearly, the development of alternative design techniques is essential if safety is to be improved, or at the very least, maintained at the current level.

A review of the literature indicates that most authors recommend more training or simply “being more careful” as ways of reducing errors. Very rarely is redesign of the device mentioned as a way to improve safety. This shows a total lack of awareness of the impact of human factors on system safety (cf., Leape, 1994). PCA is not atypical in this regard. For example, in discussing several incidents of accidental intraspinal overdose with an infusion device, one paper states: “This potentially devastating event is not the result of faulty design or functioning of the system, but, in the opinion of these authors, is

directly attributable to human error. As such, it can be reliably prevented by careful observation of the precautions and guidelines" (Patt, Wu, Bressi, & Catania, 1993, p. 202). From a human factors viewpoint, this is an astounding statement! The vast majority of researchers in the medical domain do not seem to appreciate the fact that interface design deficiencies can lead to human errors, and therefore, that device design and human error are not independent. For example, the authors just quoted go on to conclude: "While several regrettable incidents of patient injury have occurred in association with the use of the intraspinal infusion pump, it appears to be a reliable and safe device" (Patt et al., 1993, p. 202)! Until the medical community realizes that safety is an emergent property which results from the interaction between characteristics of equipment, human operators, and the environment in which the equipment is used, inadequate interfaces will continue to be designed and used.

This research has shown that the application of human factors design principles can lead to a PCA machine interface that is faster, less effortful, and more accurate to operate than a commercially available device. As far as we know, this is the first controlled study to empirically demonstrate this fact. Going beyond PCA machines in particular, this study also makes a significant contribution to the study of medical devices in general. Many of the studies in the literature looking at human factors of medical devices have either critiqued an existing device but have not proposed a new design (e.g., Cook, Woods, Howie, 1990; Cook, Potter, Woods, & McDonald, 1991), or have proposed a new design but have not evaluated it on a controlled manner (e.g., Yue, Woods, & Cook, 1992).

Nonetheless, the work presented here is not without its limitations. First, this study did not look at the issue of learning. It is not known how extended practice will affect the relative performance differences observed between the two devices. Second, the experimental evaluation did not directly address the issue of transfer from one interface to another. None of the subjects had ever programmed a PCA machine, so the results generalize most readily to novices nurses, not nurses who have extensive practice with an existing PCA interface. This is an important issue that needs to be addressed if the design proposed here is to fully realize its benefits in an operational setting in the short term. Finally, the design proposed here has only been implemented and evaluated as a prototype. Further work must be done to implement a fully functional device, and evaluate it with professional nurses in context .

The method of analysis and the design principles adopted here were based on previous work and have been shown to be applicable to different types of medical devices (Cook, Potter, Woods, & McDonald, 1991; Charante, Cook, Woods, Yue, & Howie,

1992; Yue, Woods, & Cook, 1992). It is important to point out that these principles go well beyond the human factors guidelines compiled by the Association for the Advancement of Medical Instrumentation guidelines (AAMI, 1988). Research has shown that those guidelines do not reveal many of the deficiencies in computer-based devices (Cook, Potter, Woods, & McDonald, 1991). Thus, if we are to go beyond the deficient interfaces currently in operation, medical equipment manufacturers will need to adopt human factors analysis and design methods similar to those used in this research. Hopefully, providing empirical evidence of the benefits that can be realized by such methods, as we have done in this paper, will serve as a catalyst for change.

ACKNOWLEDGEMENTS

This research was sponsored by Natural Sciences and Engineering Research Council of Canada research and equipment grants to Dr. Kim J. Vicente. We would like to thank Theresa Kinninmout, Nurse Clinician in the Acute Pain Service at TGH, and Greg Doris and the recovery room nurses at the TGH for their invaluable input and cooperation. Finally, we are also indebted to Bill Muto and Jayne Schurick for their comments on an earlier draft, and David Gaba, M.D., of Stanford University, for his encouragement and help with this work.

REFERENCES

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

Angell, M. (1980). The quality of mercy. New England Journal of Medicine, 306, 98.

Association for the Advancement of Medical Instrumentation (1988). Human factors engineering guidelines and preferred practices for the design of medical devices (AAMI HE-1988). Arlington, VA: AAMI.

Austin, K. L., Stapleton, J. V., & Mather, L. E. (1980). Multiple intramuscular injections - a major source of variability in analgesic response to meperidine. Pain, 8, 47.

Bogner, M. S. (1994). Medical devices and human error. In M. Mouloua and R. Parasuraman (Eds.), Human performance in automated systems: Current research and trends (pp. 64-67). Hillsdale, NJ: Erlbaum.

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 1992 (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., Woods, D., & Yue, L. (1992). Cognitive Engineering of the Human Computer Interface: Redesign of an Infusion Controller in Cardiac Anesthesiology.

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.

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

Gaba, D. M. (1994). Automation in anesthesiology. In M. Mouloua and R. Parasuraman (Eds.), Human performance in automated systems: Current research and trends (pp. 57-63). Hillsdale, NJ: Erlbaum.

Golonka, L. (1986). Trends in health care and use of technology by nurses. Medical Instrumentation, 20, 8-10.

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 Patient Controlled Analgesia Machine. Unpublished B.A.Sc. Thesis, University of Toronto, Toronto, Ontario.

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

Marks, R. M., & Sachar, E. J. (1973). Undertreatment of medical inpatients with narcotic analgesics. Annals of Internal Medicine, 78, 173.

Molich, R., & Nielson, J. (1990). Heuristic evaluation of user interfaces. In Proceedings of CHI '90 (pp. 249-256). New York: ACM Press.

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.

Patt, R. B., Wu, C., Bressi, J., & Catania, J. A. (1993). Accidental intraspinal overdose revisited. Anesthesia and Analgesia, 76, 202.

Perrow, C. (1984). Normal accidents: Living with high-risk technologies. New York: Basic Books.

Rouse, W. B. (1984). Computer-generated display system guidelines, volume 2: Developing an evaluation plan (EPRI NP-3701). Palo Alto, CA: EPRI.

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.

Scott, L. E., Clum, G. A., & Peoples, J. B. (1983). Preoperative predictors of postoperative pain. Pain, 15, 283.

Smythe, M. (1992). Patient-controlled analgesia: A review. Pharmacotherapy, 12, 132-143.

Sriwantanakul, K., Weis, O. P., Alloza, J. L., et al. (1983). Analysis of narcotic usage in the treatment of postoperative pain. Journal of the American Medical Association, 250, 926.

Teske, K., Dant, R., & Cleeland, C. S. (1983). Relationship between nurses observations and self-report of pain. Pain, 16, 289.

Yue, L., Woods, D. D., & Cook, R. I. (1992). Cognitive engineering of the human-computer interface: Re-design of an infusion control in cardiac anesthesiology

(CSEL Report TR-01-92). Columbus, OH: Cognitive Systems Engineering Laboratory, Ohio State University.

Vicente, K. J., & Rasmussen, J. (1992). Ecological interface design: Theoretical foundations. IEEE Transactions on Systems, Man, and Cybernetics, SMC-22, 589-606.

White, P. F. (1985). Patient-controlled analgesia: A new approach to the management of postoperative pain. Seminars in Anesthesia, 4, 255.

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.