APPLYING HUMAN FACTORS TO THE DESIGN OF MEDICAL EQUIPMENT: PATIENT-CONTROLLED ANALGESIA

Laura Lin, MASc,[1,2] Racquel Isla, BASc,[1] Karine Doniz, BASc,[1] Heather Harkness, BASc,[1] Kim J. Vicente, PhD,[1,2] and D. John Doyle, MD[2,3]

Lin L, Isla R, Doniz K, Harkness H, Vicente KJ, Doyle DJ. Applying human factors to the design of medical equipment: patient-controlled analgesia.

J Clin Monit 1998; 14: 253–263

ABSTRACT. Objective. Medical instruments commonly have poorly designed user interfaces that promote human errors with life-threatening consequences. The primary hypothesis of this study was that a specific user interface could be made safer and more efficient if redesigned using human factors techniques and principles. **Methods.** The user interface of a commercially available patient-controlled analgesia (PCA) pump, the Abbott Lifecare 4100 PCA Plus II infuser, was evaluated using a cognitive task analysis of bench tests and field observations. Based on this analysis, the user interface was redesigned. Important elements of the new design include a dialog structure with fewer steps, a dialog overview showing the user's location in the programming sequence, better command feedback, easier error recovery, and clearer labels and messages. The changes were evaluated by comparing a computer prototype of the new interface with a computer simulation of the old one. Twelve student nurses performed six programming tasks with each interface. Task completion time, number of errors, and subjective mental workload were collected for each trial. **Results.** The results showed significantly faster programming times ($F(1,11) = 6.85$, $P < 0.025$), lower mental workload ratings ($\chi^2(1) = 4.45$, $p < 0.025$, one-tailed), and fewer errors ($\chi^2(1) = 3.33$, $p < 0.05$, one-tailed) with the new interface. **Conclusion.** Adopting a human factors approach to redesigning the PCA interface led to significantly faster, easier, and more reliable performance. These findings have important implications for improving the design of other computer-based medical equipment.

KEY WORDS. Medical equipment, interface design, human factors, task analysis, patient-controlled analgesia, human-computer interaction.

INTRODUCTION

Human error plays a crucial role in the safety of medical equipment. Sixty percent 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 [1]. This figure is comparable to those reported in other domains (e.g., nuclear, aviation) [2]. Experience in those domains has shown that errors can be reduced through human factors design techniques that take into account human capabilities and limitations. In the medical domain, however, human factors issues have received comparatively little attention [1, 3], despite the fact that nurses and doctors are also dealing with complex situations where human lives are at stake. The primary hypothesis explored in this paper is that,

From the [1]Cognitive Engineering Laboratory, Department of Mechanical & Industrial Engineering, University of Toronto, [2]Institute of Biomedical Engineering, University of Toronto, and [3]Department of Anaesthesia, University of Toronto, Canada.

Received Apr 8, 1997. Accepted for publication Feb 14, 1998.

Address correspondence and requests for reprints to Dr Kim J. Vicente, Department of Mechanical & Industrial Engineering, University of Toronto, 5 King's College Road, Toronto, Ontario, Canada M5S 3G8.
E-mail: benfica@mie.utoronto.ca

© 1998 *Kluwer Academic Publishers. Printed in the Netherlands.*

by adopting human factors design principles, medical equipment can be made safer and more efficient.

Patient-Controlled Analgesia (PCA) pumps were selected as a context for testing this hypothesis. There are three types of safety problems related to PCA use [4]: (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 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 [4].

With few exceptions, the possibility of reducing human errors by applying human factors design techniques is never acknowledged in the PCA literature. Instead, the most frequently mentioned intervention is to increase the amount of training [4–6]. But rather than increasing the training required to use a poorly designed interface, it seems more logical to redesign the interface to make it simpler to use. The task of interacting with the machine should be made as transparent as possible [7] so that nurses can focus on the task at hand, i.e., patient comfort. Several cases of PCA machine misprogramming reported in the literature [3, 6] suggest that this goal has not yet been achieved. Ease of use does not seem to have been an important criterion in the design and evaluation of existing PCA machines.

Several comparative evaluations of PCA machines have been reported in the literature. Sawaki et al. [8] evaluated 5 different devices in postsurgical wards, using questionnaires given to nurses and patients. However, only a few questions addressed usability, and even these did not provide explicitly defined criteria (e.g., How easy was it to learn how to program and use each of these devices?). Thus, the validity of the responses is questionable. Also, the study only measured the opinions of users, not performance. It is well known that people may express a preference for one device, when performance data show no difference or even an advantage for another device [9]. As a result, the conclusions one can derive from this study regarding usability are limited. A second PCA machine evaluation focused on mechanical properties, and did not evaluate ease of use [10]. Nevertheless, the authors claimed that the machines tested were "user friendly." Ilsley et al. [11] developed an automated system to conduct standardized evaluations of PCA machines. However, this system only measures the reliability and accuracy of the amount of medication delivered by the PCA pump, not the adequacy of the interface design. In summary, there is a strong need for evaluating PCA machines from a human factors perspective, a pump that reliably delivers the required amount of medication can still lead to catastrophe if it has a poorly designed interface that induces programming errors.

METHODS

This study consisted of a cognitive task analysis (CTA) of the interface to the Abbott Lifecare 4100 PCA Plus II infuser, an analysis and redesign of that interface based on human factors design guidelines, and an experiment comparing the old and new interfaces.

Cognitive task analysis

CTA was employed as a systematic method of studying the psychological requirements of programming the PCA machine. The CTA methodology used was based on previous research [12–14]. Data were collected using: bench tests (i.e., hands-on expert review), field observations, and comments from recovery room nurses after a presentation of the proposed interface. A detailed description of this procedure can be found in Isla & Lin [15] and Doniz & Harkness [16], so only a brief overview will be provided here.

The bench tests aided in identifying the characteristics of the device that make its operation prone to errors: poor assembly mechanisms, faulty operations, and device failures. Also, a thorough mapping of internal system structure was conducted to identify design deficiencies [15].

The field study took place in the recovery room of the Toronto General Hospital (TGH). It involved observing and interviewing nurses responsible for programming the current PCA machine. The study was carried out over 8 visits during which 9 different nurses were interviewed, and the programming of 30 PCA machines was observed.

Development of a new design followed the bench tests and field study. The proposed design was demonstrated to TGH nurses during an oral presentation to solicit feedback which would aid in further refining the design, and to foster user acceptance of the new design ideas. Several design changes were directly motivated by the feedback from the users [15].

Human factors design principles

The human factors design principles [13, 14] that were used to evaluate and subsequently redesign the interface to the Abbott Lifecare PCA Plus II were:

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 reliable shortcuts to increase efficiency.
6. Provide clearly marked exits for the user to leave the system.

Experimental evaluation

Twelve subjects who had no prior experience with the current Abbott Lifecare PCA interface were selected from University of Toronto students enrolled in the nursing program. A 2 × 3 × 2 × 2 mixed design was adopted with Interface (Old vs. New), Programming Task (PCA, Continuous, and PCA + Continuous), and Repetitions (first and second repetition of each programming task) as within-subjects factors, and Interface Order (Old First vs. New First) as a between-subjects factor.

Graphical simulations of both interface designs were developed using the Toolbook software development package (Asymetrix Corporation, Bellevue, WA). The simulations ran on an IBM-compatible (PC 486-DX, 33 MHz) equipped with a mouse and a SVGA color monitor. Input data for the programming task were provided to subjects on standard PCA order forms used at the TGH.

Subjects were briefed on the purpose of the experiment, and the tasks they would be performing. They then received instructions on how to program the device and hands-on-practice as well. For each trial, subjects were provided with a PCA order form and were asked to program the machine according to the values on the form. After each trial, subjects completed a mental workload rating scale. At the end of the experiment, informal comments were solicited from subjects to determine which of the two interfaces they preferred.

The total time to successfully complete each trial and the number of errors were recorded by computer. Subjective ratings of mental workload were also collected using the NASA-TLX method, a well accepted measure of subjective mental workload [17].

Completion time data were analyzed with ANOVA, the number of errors was analyzed using a χ^2 test, and workload data were analyzed using both ANOVA and χ^2.

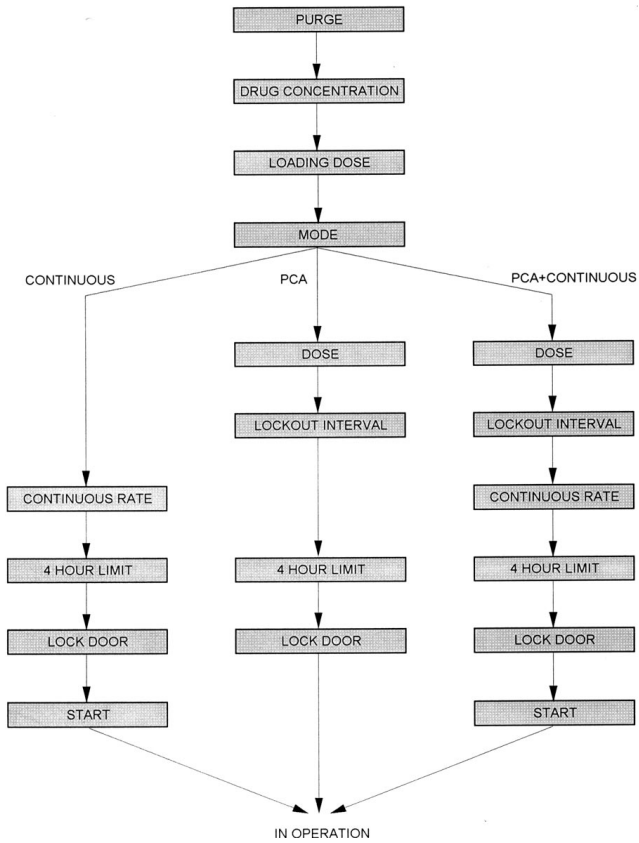

Fig. 1. Function flow diagram of the programming sequence for the Abbott PCA.

RESULTS

Cognitive task analysis

Bench testing led to three state transition diagram analyses that summarize the structure of the tasks that users are required to perform with the existing interface [15]:

1. general flow of activities (function flow diagram of the "subtasks" performed to complete the task of programming – see Figure 1);
2. decisions and actions required of each subtask; and
3. detailed mapping of sequential LCD messages and user input.

The results showed that the structure of many subtasks in the programming sequence was unnecessarily complex. For example, the mode selection subtask, shown in Figure 2, displays a choice between three related alternatives as three unrelated decisions. This procedure is time consuming, and neglects the need for a global view of the decision options. Problems such as this

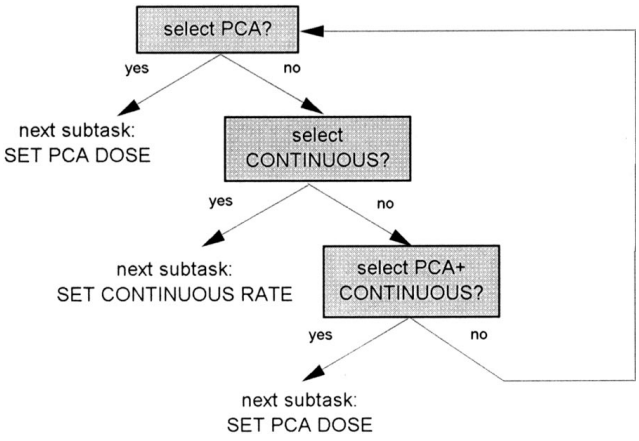

Fig. 2. Decisions and actions for the subtask of selecting operating mode.

anticipate the need for significant changes to the interface.

An analysis of the information requirements associated with each dialogue step was performed next. The information users needed to perform the tasks was compared to the information directly provided by the interface. Any mismatch represents a design deficiency. On the positive side, the interface guides users through the programming sequence, allowing them to "perceive and act" and thus reducing dependency on working memory. On the negative side, there is a lack of external representation of the dialog structure. There is no overview to remind users how many parameters there are to program, what order the sequence follows, what steps in the sequence have been completed, and what steps remain to be performed. Because the interface does not provide this information, an unnecessary memory load is placed on users. Furthermore, the touch switch panel (see Figure 3) provides a misleading visual grouping of the touch switches that fails to reinforce the structure of the programming sequence. Finally, there is also a problem with the labeling of the touch switches. 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. These problems can be observed in detail in Figure 4 which shows the information required by the user and the information provided by the device for each programming and error recovery subtask. Various mismatches between what is required and what is provided can be identified, indicating a need for redesign.

Observing the machine and its users in context drew attention to the added demands that the hospital environment placed on users. The distractions and noise in

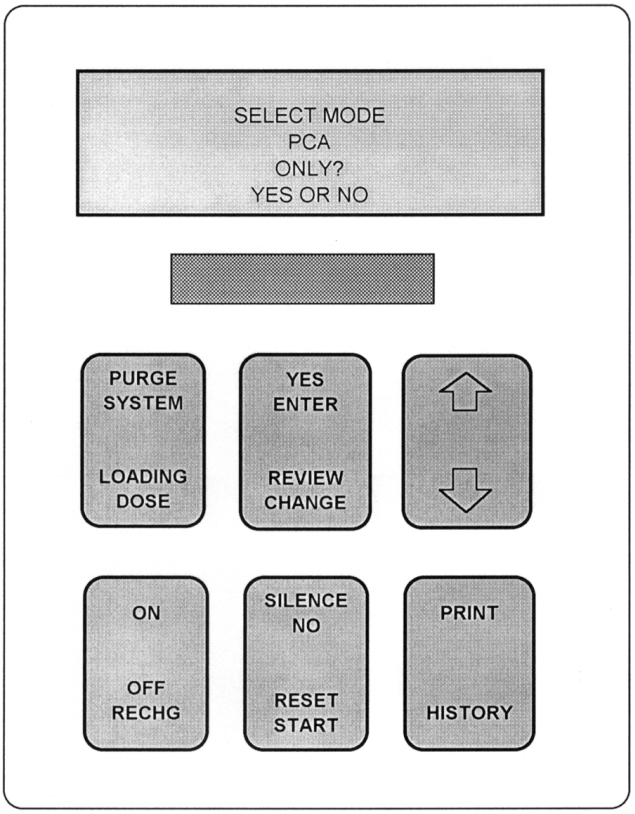

Fig. 3. Current interface for the Abbott PCA.

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, since those tests were performed under comparatively favorable conditions.

User responses reiterated the problem areas identified in the bench tests, especially the complex or lengthy programming procedures, tedious error recovery process, and indistinguishable touch switches.

The CTA results provided us with a solid understanding of the demands that these tasks impose on users, the context in which these tasks must be performed, as well as an indication of how the current interface fails to provide the resources to help users effectively and reliably deal with these task demands.

Analysis of existing interface using human factors guidelines

The user should be given feedback about what operation has been executed, as well as the results of the operation [13]. Feedback allows users to always be aware of the status of the system, and to detect and recover

SUBTASK	INFORMATION REQUIRED BY USER	
	PROVIDED BY DEVICE	NOT PROVIDED BY DEVICE
PURGE	‡ Disconnect patient from PCA machine before purging * Use PURGE touchswitch to begin purging the system	|| Purge is successful; slack is cleared from tubing ** Can purge system any time during programming procedure ** Purging clears slack from tubing
SELECT DRUG CONCENTRATION	* Use arrow keys to set value † Press ENTER after setting value § Min & max limits	** No need to enter drug name ** Defaults presented sequentially in a loop; can return to a particular default if skipped
LOAD DOSE	* Use arrow keys to set dose value † Press ENTER after setting value * Press LOAD DOSE to infuse	†† Loading dose is not infused immediately after answering YES to "LOAD DOSE NOW?" ** Can press LOAD DOSE anytime during programming procedure
SELECT MODE	* Press YES to select	** Mode options presented sequentially in a loop ** "PCA" and "CONTINUOUS" are different from "PCA+ CONTINUOUS"
SET PCA DOSE, LOCKOUT INTERVAL, CONTINUOUS RATE	* Use arrow keys to set value * Units of parameter setting † Press ENTER after setting value § Min & max limits	
SET 4 HOUR LIMIT	* Use arrow keys to set the limit † Press ENTER after setting value § Min & max limits	†† Answering YES to "4 HR DOSE LIMIT SET?" allows user to set the limit as opposed to confirming that it has already been set
CHANGE A SETTING (in programming mode)	* Value of original setting	** Use REVIEW/CHANGE to change a setting ** Press REVIEW/CHANGE x times to go back x screens ** Cannot use REVIEW/CHANGE to return to go back to PURGE screen
CHANGE A SETTING (in infusion mode; after programming completed)	† Value of original setting ¤ If one setting changed, other settings may also require change	** Press REVIEW/CHANGE to change settings ** REVIEW/CHANGE will begin new procedure; will NOT take user back to last setting unlike procedure for changing a setting during programming mode

Legend:

* Given on LCD
† Given on LCD but troublesome to view or obscured because of large amount of information presented or
‡ Flashed on LCD for several seconds
§ Beeps when limits are reached
¤ Device guides user through procedure
|| Not a device function
** Must learn from experience or training
†† Wording is unclear

Fig. 4. Information requirements for programming subtasks.

from errors. The Abbott PCA machine provides a limited amount of feedback to users 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 frustrating for a new user because programming is a fairly lengthy procedure.

The task of keeping track of the functions of different controls is affected by the choice of labels and the number of functions for each control. Multifunctional

controls can be used to carry out several functions, requiring users to remember the different operations that a particular control performs in different contexts. This increases the user's mental workload, and thus the chances of making errors [13].

In the case of the Abbott PCA pump, the function of the various controls is not very clear. For example, the user must remember that the ON and RESET/START touch switches have different effects, despite the similar labels. 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.

Confusion over control function also stems from the multifunctional nature of some touch switches. For instance, the REVIEW/CHANGE touch switch is multifunctional: it 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, users press REVIEW/CHANGE to scroll backwards through the programming sequence until they reach the desired parameter. However, if programming has been completed (i.e., all parameters have been set), then REVIEW/CHANGE takes the user forward through a series of displays similar to those in the programming sequence, and allows the user to make changes to any of the settings.

The first REVIEW/CHANGE function also appears to suffer from inconsistency. It is unclear how many steps backward will result from one press of the touch switch during programming, as evidenced by the following examples:

Example 1. The user is setting the value for drug concentration and realizes that he/she has mistakenly set the concentration to μg/ml instead of mg/ml. Pressing REVIEW/CHANGE returns the user to the previous screen for selecting the concentration units, and allows him/her to correct this setting.

Example 2. The user has already set the loading dose and is about to infuse it but realizes that the value has been incorrectly set and wants to return to the previous screen to correct it. However, after pressing REVIEW/CHANGE, the user is taken back to the screen used to set the drug concentration.

In the first example, REVIEW/CHANGE takes users back to a display which permits them to set the drug concentration in different units. Users remain in the same subtask (selecting drug concentration). However, in the second example, REVIEW/CHANGE does not permit users to change the loading dose selected in the preceding screen. Instead, it removes them from the active subtask (administering a loading dose) and returns them to a completely different subtask (selecting drug concentration). Collectively, these examples show the need for improved design in touch switch functioning.

Displayed messages should be clearly understandable to the user. Messages which are unclear may themselves be sources of error [13]. With the current machine, several LCD messages are awkwardly worded and sometimes even ambiguous. For instance, the message below 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 ask users if they would like to change the programming mode or any of the settings. It is not apparent from this message 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.

The interface should minimize the amount of information that users must remember 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 [17]. 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 h limit are set instead. The interface dialogue ensures that the relevant displays are brought up, as a function of mode. However, the CTA results from bench testing and analysis of information requirements have indicated that there are several other features of the interface which unnecessarily increase the burden on users' memory, making errors more likely.

When users are comfortable with the system, short-

Table 1. 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 fuction
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	Medium	Provide functional grouping
Misleading and confusing labels	Hard to interpret Slows down user	Medium	Provide labels that are user-driven
Misleading and confusing messages	Hard to interpret Slows down user	Medium	Provide messages that are user-driven
Review/change inconsistent	Disorienting and confusing	Low	Make consistent
Defaults not appropriate	Many steps Takes time	Low	Use TGH standard

cuts in the programming procedure allow them to progress through it much more quickly. Shortcuts should only eliminate the steps that are unnecessary in a particular task [18], and the device should operate as normal afterwards. By systematically supporting reliable shortcuts, improvements in efficiency may be obtained without incurring errors due to oversights.

One method of providing shortcuts is to set appropriate machine defaults. The default values provided on the current interface conflict with the standard operating values used at TGH. For example, the most widely used concentration is 2.0 mg/ml. Users must currently 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.

Providing appropriate exits can facilitate error recovery [18] by allowing users to leave the system after having inadvertently entered incorrect commands. In several programming tasks, 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 users accidentally skip the desired concentration setting, they cannot use REVIEW/CHANGE to go back to it. Instead, they must toggle through all the screens (in a loop) and start again from the beginning. Similarly, in the sequence for selecting the mode, users are prompted sequentially by the LCD messages to select one of the three modes. These screens occur in a loop without an exit.

In summary, the Abbott PCA Plus II interface does not conform with existing human factors guidelines, thereby unnecessarily increasing both the demands on users and the probability of human errors. The design deficiencies that were identified are summarized in Table 1, along with the effects of each problem, its severity, and a proposed solution.

Redesigned interface using human factors guidelines

The proposed solutions to the design problems in Table 1 were integrated to create a redesigned interface that obeys human factors design guidelines. It is important to note that the new 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 was taken to prevent extensive deviation from current manufacturing costs of the PCA machine.

The new interface, shown in Figure 5, consists of an enlarged LCD screen to house an indicator field and menu display, in addition to the message field. The indicator field displays the programming task that is being performed, while the menu display shows the three stages of programming: concentration, mode, and settings. As the program leads the user through each

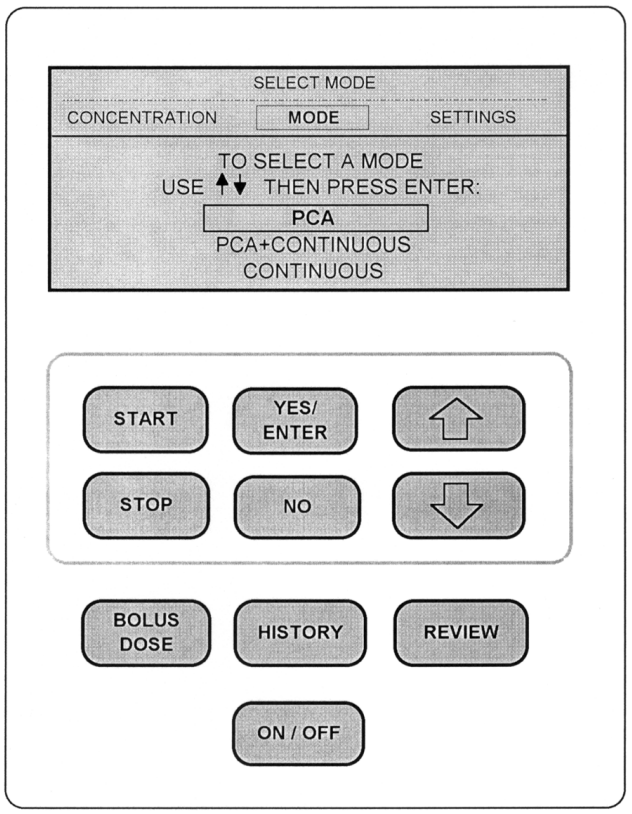

Fig. 5. Redesigned PCA interface.

stage, the corresponding item on the menu display 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.

The touch switch control panel on the new interface was 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 those that are special-purpose (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 those of the current interface due to modifications to the system structure. The BOLUS DOSE on the new design replaces the LOADING DOSE touch switch. This new label more accurately describes its function, as indicated by the nurses at TGH.

REVIEW/CHANGE has been relabeled REVIEW. It maintains the same purpose, allowing users to review the settings and to make changes where necessary. When pressed during programming, it performs similarly to 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 users' 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 users lock the door and hook the machine up to the patient.

The system structure was simplified by minimizing the number of message display screens. The major change was an introduction of a menu system that presented related choices in parallel rather than serially. 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 modification decreases the number of screens users must toggle through, and provides users with a more global view of the dialog structure by displaying all related decision options on a single screen. Simplification was also achieved by providing the most common settings and values used at TGH as defaults to avoid unnecessary programming steps. Finally, message screens for tasks that were fixed in the protocol of machine setup (purging and setting 4 h limit) or extraneous to programming (e.g., bolus dose) were either condensed or eliminated from the programming se-

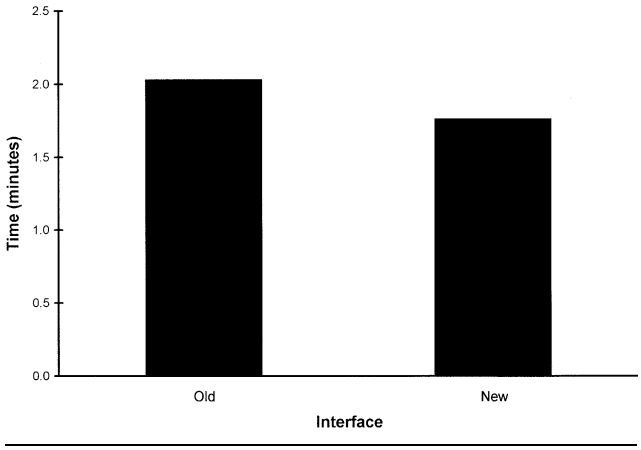

Fig. 6. Mean programming time.

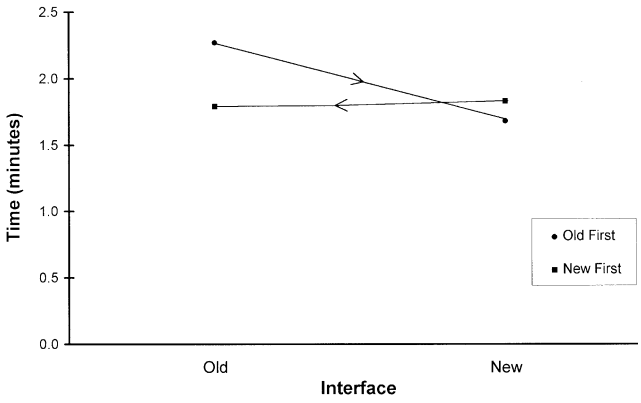

Fig. 7. Mean programming time as a function of interface and order of presentation.

quence. For instance, in the existing interface, the prompt that asks the user: "SET 4 HOUR LIMIT? YES OR NO" was eliminated because the 4 hour limit must always be set for all programming modes. The sequence on the new design will go directly to instructing the user to input the limit.

The combined effect of these changes to the programming sequence is a significant transformation in the complexity of the dialog structure. 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%.

Experimental evaluation

Because of the aforementioned changes, the new interface was expected to be more efficient, less demanding, and more reliable than the old interface. A four-way ANOVA was conducted with Interface, Task, Repetitions, and Order as the main factors and task completion time as the dependent variable. The Interface effect was significant ($F(1,11) = 7.23$, $p < 0.025$). As shown in Figure 6, the mean programming time with the Old interface was 2.03 minutes, whereas that with the New interface was 1.76 minutes, 15% faster. However, this main effect can only be meaningfully interpreted within the context of the Order × Interface interaction, which was also statistically significant ($F(1,11) = 9.76$, $p < 0.01$). As shown in Figure 7, there is unequal transfer from one interface to the other. There are several ways to look at this effect. First, transferring from the Old to the New interface causes a larger performance change than going from the New to the Old (differences of −0.59 minutes and −0.05 minutes, respectively). Second, those subjects who have already had some experience at the task with the New interface are moderately faster, as a result of learning, with the Old interface than subjects who are doing the task for the first time with the Old interface. Third, those subjects who have already had some experience at the task with the Old interface are much faster 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 first benefit from a transfer of training to the New interface, whereas the Old interface seems to inhibit the benefits of practice and training for the group exposed to the New interface first.

The mental workload ratings were transformed into percentages, and a similar ANOVA was conducted. There was no statistically significant difference in workload between the two interfaces ($F(1,11) = 2.2$, $p = 0.12$). However, the average workload rating for the Old interface was more than twice as high as that for the New interface (15% and 7%, respectively), as shown in Figure 8. This result may have been caused by the significantly lower workload ratings of the subjects who were first exposed to the Old interface compared to those who were first exposed to the New interface ($F(1,11) = 8.36$, $p = 0.016$). Subjects in the former group confined their workload ratings (for 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. A χ^2 test was conducted to determine whether more subjects experienced higher workload ratings with the Old or New interface. The results showed that 82% of the subjects had a higher workload with the Old interface ($\chi^2(1) = 4.45$, $p < 0.025$, one-tailed).

As shown in Figure 9, the New interface led to 10

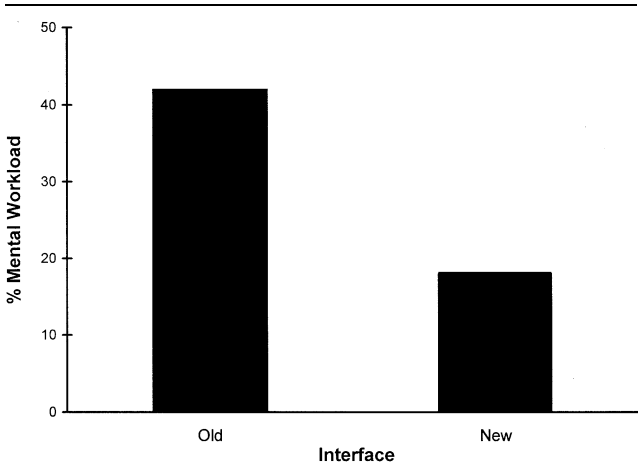

Fig. 8. Mean mental workload.

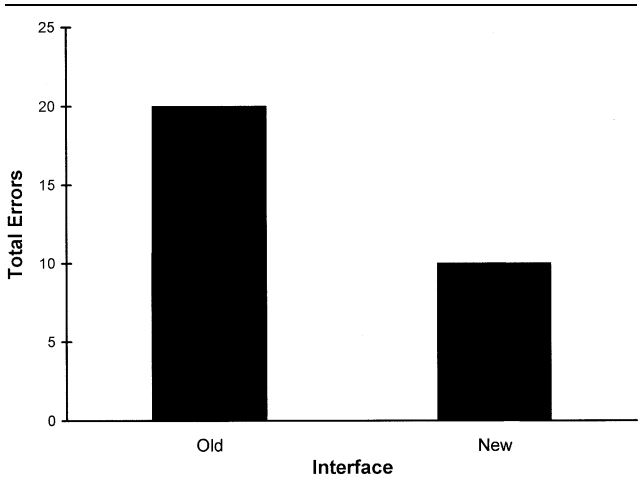

Fig. 9. Total number of programming errors.

errors, whereas the Old interface led to a total of 20 errors, a difference of 100% ($\chi^2(1) = 3.33$, $p < 0.05$, one-tailed). Thus, the New interface led to a significantly more reliable performance than the Old interface.

The informal comments expressed by subjects at the end of the experiment were consistent with the results just presented. All twelve participants expressed a strong preference for the New interface design.

DISCUSSION

This research has shown how human factors techniques can be used to improve the design of medical equipment. A CTA of the Abbott PCA Plus II infuser provided a systematic means of identifying the information users require to program the machine effectively. A review of the existing interface using human factors design guidelines revealed many deficiencies that make the users' task more difficult, and thus more error-prone, than it needs to be. The same set of guidelines were systematically used to design a new interface that remedied these deficiencies. An empirical evaluation of the old and new designs showed that the redesigned interface lead to faster programming times, lower ratings of mental workload, and fewer programming errors. These results strongly support the hypothesis behind this study, namely that the use of medical equipment can be made safer and more efficient by adopting human factors design principles. As far as we know, this is the first controlled study to empirically demonstrate this fact.

Going beyond PCA machines in particular, this research also makes a significant contribution to the study of medical devices in general. Many of the studies in the literature on human factors of medical devices have either critiqued an existing device but have not proposed a new design (e.g., [14, 19]), or have proposed a new design but have not evaluated it in a controlled manner (e.g., [13]). While we have selected PCA machines in this study, the design methods and conclusions presented here are believed to generalize to other computer-based medical devices.

Nonetheless, the work presented here is not without its limitations. First, this study did not systematically examine learning. It is not known how extended practice will affect the relative performance differences observed between the two interfaces. Second, the experimental evaluation did not comprehensively 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 novice nurses, not to nurses who have extensive practice with the existing PCA interface. This is an important issue that needs to be addressed if the design proposed here is to be introduced into an operational setting where nurses have experience with other interfaces. Third, the design proposed here has only been implemented and evaluated as a prototype using a computer screen and mouse. Further work must be done to implement a fully functional physical device with a keypad, and evaluate it with professional nurses in context. Finally, we have been told that the interface for the Abbott PCA Plus II infuser has been redesigned by the manufacturer with different buttons and software since our study was conducted. We do not know how the newer design compares with the two that we evaluated here.

The method of analysis and the design principles adopted in this research were based on previous work and have been shown to be relevant to different medical devices [12–14]. 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 [20]. Research has shown that those guidelines do not reveal many of the deficiencies in computer-based devices [14]. 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 article, will serve as a catalyst for change.

This research was sponsored by research and equipment grants from the Natural Science and Engineering Research Council of Canada 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. We are also indebted to Bill Muto and Jayne Schurick for their comments on an earlier draft, and David Gaba, MD, of Stanford University, for his encouragement and help with this work. Finally, thanks to Jeffrey M. Feldman, MD, and the reviewers for exceptionally thorough and constructive comments.

REFERENCES

1. Bogner MS. Medical devices and human error. In: Mouloua M, Parasuraman R, eds. Human performance in automated systems: Current research and trends. Hillsdale, NJ: Erlbaum, 1994: 64–67
2. Perrow C. Normal accidents: Living with high-risk technologies. New York: Basic Books, 1984
3. Leape LL. Error in medicine. J Am Med Assoc 1994; 272 (23): 1851–1857
4. White PF. Mishaps with patient-controlled analgesia. Anesthesiology 1987; 66: 81
5. Smythe M. Patient-controlled analgesia: A review. Pharmacotherapy 1992; 12: 132–143
6. Cohen MR. Preventing errors associated with P.C.A. pumps. Nursing 1993; 23: 17
7. Vicente KJ, Rasmussen J. Ecological interface design: Theoretical foundations. IEEE Transactions on Systems, Man, and Cybernetics 1992; SMC-22: 589–606
8. Sawaki Y, Parker RK, White PF. Patient and nurse evaluation of patient-controlled analgesia delivery systems for postoperative pain management. J Pain Symptom Management 1992; 7: 443–453
9. Rouse WB. Computer-generated display system guidelines, volume 2: Developing an evaluation plan (EPRI NP-3701). Palo Alto, CA: EPRI, 1984
10. Owen H, Glavin RJ, Reekie RM, Trew AS. Patient-controlled analgesia: Experience of two new machines. Anaesthesia 1986; 41: 1230–1235
11. Ilsley AH, Owen H, Plummer JL, Mackey NA, Roberts DRD. A system for standardized evaluation of patient-controlled analgesia devices: Design, construction, and engineering aspects. J Clin Monit 1994; 10: 194–200
12. Charante EM, Cook RI, Woods DD, Lue Y, Howie MB. Human-computer interaction in context: Physician interaction with automated intravenous controllers in the heart room. In: Stassen HG, ed. Analysis, design, and evaluation of man-machine systems 1992. London: Pergamon Press, 1992: 263–274
13. Yue L, Woods DD, Cook RI. 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, 1992
14. Cook RI, Potter SS, Woods DD, McDonald JS. Evaluating the human engineering of microprocessor controlled operating room devices. J Clin Monit 1991; 7: 217–226
15. Isla R, Lin L. Investigation of the Effectiveness of the User Interface of Patient Controlled Analgesia Machines. Unpublished B.A.Sc. Thesis, University of Toronto, Toronto, Ontario, 1993
16. Doniz K, Harkness H. Interface for Patient Controlled Analgesia Machine. Unpublished B.A.Sc. Thesis, University of Toronto, Toronto, Ontario, 1994
17. Wickens CD. Engineering psychology and human performance. 2nd ed. New York: Harper-Collins, 1992
18. Molich R, Nielson J. Heuristic evaluation of user interfaces. In Proceedings of CHI'90. New York: ACM Press, 1990: 249–256
19. Cook RI, Woods DD, Howie MB. The natural history of introducing new information technology into a high-risk environment. In Proceedings of the Human Factors Society Annual Meeting. Santa Monica, CA: Human Factors Society, 1990: 429–433
20. Association for the Advancement of Medical Instrumentation. Human factors engineering guidelines and preferred practices for the design of medical devices (AAMI HE-1988). Arlington, VA: AAMI, 1988