Usability Assessment of Pacemaker Programmers

CHRISTINE C. CHIU, KIM J. VICENTE,* ILAN BUFFO-SEQUEIRA, ROBERT M. HAMILTON, and BRIAN W. McCRINDLE

From the Division of Cardiology, The Hospital for Sick Children, and the *Cognitive Engineering Laboratory, Department of Mechanical and Industrial Engineering, University of Toronto, Toronto, Ontario, Canada

CHIU, C.C., ET AL.: Usability Assessment of Pacemaker Programmers. *There is a perception among clinicians of usability differences in the user interface of pacemaker programmers, but there is an absence of literature in this area. The purpose of this study was to describe usability differences in pacemaker programmers. Forty-two programmer users completed self-administered questionnaires and two usability experts independently performed heuristic evaluation to identify features that violated general usability principles. Programmers from seven manufacturers (coded A–G) were evaluated. There was a balanced representation of users: nurses (58%) versus technologists (40%) who are employed in community (50%) versus academic (45%) hospitals, novice versus expert users based on the median users' programming experience of 60 months (range 1–300 months). Significant differences between programmers were found in overall user satisfaction and ease of programmer use ($P < 0.0001$) in the display, controls, operation, and physical dimension of the programmers ($P < 0.05$). Heuristic evaluations showed frequent violations of usability principles in all programmers. Problematic areas include reliance on user recall, inconsistency in operation of critical controls, poor readability, and not anticipating user wants or action. Programmer interface designs do not consistently meet user needs or general usability principles. This impacts on the safe and effective use of programmers. Guidelines in programmer design should be established, particularly with respect to labeling, location, and operation of critical controls. (PACE 2004; 27:1388–1398)*

pacemaker, implantable defibrillator, programmer, usability, user interface, heuristic evaluation, medical device design

Introduction

Implantable pacemakers and defibrillators are used by clinicians for management of patients with heart rhythm disorders.[1–3] The follow-up assessment of these implanted devices are conducted through special laptop computers called "pacemaker programmers." A pacemaker programmer is made up of computer hardware called the *platform* and a software interface from which users operate the controls and perform their follow-up tasks. Broadly speaking, one may conceptualize the programmer user interface as having input, output, and processing components within the human operator element and the computer element (Fig. 1) The relationship between the design of any user interface and ease of use for the human operator is critical to the overall usability. *Usability* is generally defined as the effectiveness, efficiency, and user satisfaction in the design of a product for a specific use.[4,5] Poor user interface designs can lead to errors in operation, inefficiencies, and user frustrations.[6] Usability testing plays an important role in unmasking design flaws leading to improvements to meet user needs.[7]

There are no guidelines in the user interface design of pacemaker programmers. General human to computer interface guidelines are probably adopted to varying degrees in most designs.[6,8] Personal communication with engineers from pacemaker manufacturers has shown varying involvement of human factors specialists in the design of programmers. These issues may be responsible for the perception amongst clinicians of usability differences between programmers from different manufacturers, although there is an absence of literature in this regard. The purpose of this study was to describe the usability differences and identify problematic areas in the user interface of pacemaker programmers.

Methods

Data were collected from users and usability experts to ensure accuracy and confidence in interpreting the findings of this study.[9] Programmer users were surveyed by anonymous completion of a self-administered questionnaire. Two undergraduate engineering students who specialized in human factors and usability research on medical equipment independently performed heuristic evaluation of the programmers. This study was

Supported in part by the Cardiology Advisory Council at the Hospital for Sick Children, Toronto, Canada.

Address for reprints: Christine Chiu, BSc, Div. of Cardiology, The Hospital for Sick Children, 555 University Ave., Toronto, Ontario, Canada M5G-1X8. Fax: 416-813-7547; e-mail: christine.chiu-man@sickkids.ca

Received February 3, 2004; revised June 9, 2004; accepted July 8, 2004.

approved by the Hospital for Sick Children Research Ethics Board.

Pacemaker Programmers

Manufacturers of pacemaker programmers were coded by letters A–G . The platforms assessed in this study are those used in device clinics as of November 2002.

Participants

Volunteers were recruited from the allied professional participants of a regional conference on electrophysiology and device management in November 2002. It was estimated that about 50 of 90 participants were involved in pacemaker follow-up from Canadian provinces of Ontario, Quebec, and Nova Scotia. Observers with no hands-on programming experience or employees of pacemaker manufacturers were excluded. Additional volunteers were recruited by telephoning ten pacemaker clinic sites whose staff did not attend the November 2002 conference.

User Questionnaire

The questionnaire is comprised of three sections: (1) cover sheet explaining the study, (2) user information data form, and (3) user opinion data form to complete for each programmer type used in their practice.

On the user opinion form, there were 20 Likert scale items with 4–6 discrete response options (e.g., ranked from 1–6, very easy to very difficult) and open-ended items to allow respondents to write comments in their own words: what they like the most and dislike the most about the programmer, what they would like to change about the programmer, and to write down any problems they have encountered.

Heuristic Evaluation

This is a methodology in human factors engineering whereby clinical or human factors expert(s) evaluates user interface system to determine how well it complies with established usability principles like the 15-point checklist[8,10] (Appendix 1). This technique is useful in identifying problematic areas and allows comparison of interface designs.

Two usability experts independently worked with the principle investigator to evaluate six pacemaker programmers by a walk-through of a typical operator task list:[11]

- Communication or interrogation of the implanted device with the programmer
- Obtain parameter settings, lead impedance, battery data, histogram counter data
- Obtain other diagnostic data like trends, arrhythmia events, etc.
- Perform stimulation and sensing threshold tests
- Print a copy of the electrogram (EGM) /electrocardium (ECG)
- Identify lead or battery failure if present
- Optimize any parameter settings by reprogramming
- Obtain final interrogation and printout of parameters

The usability experts commented on violations of the user interface to established heuristic principles and made suggestions for improvements. Usability experts did not evaluate programmer E since there were only six user evaluations received and their devices were no longer implanted.

Data Analysis

Questionnaire responses were coded using the ranking scheme established for the response options of the Likert scale items on the data forms (e.g., from 1, very easy, to 6, very difficult). Missing data was not imputed but reported as missing for that category. Programmer performances were reported by comparison of their features rather than comparison of manufacturers. In view of the small number of evaluations (< 10) received for programmers E–G, these were not included in the quantitative analysis but their differences were described.

Descriptive statistics (median and range) were used to summarize the characteristics of the users and the questionnaire items. The Kruskal-Wallis test, a nonparametric test, was used to test for differences in the rank of median responses for three or more groups, in this case, between programmers A–D. A P value < 0.05 was defined as statistically significant. The mean responses were ranked from the questionnaire items to discriminate the differences between these four programmers. Textual comments/keywords from users and usability experts were sorted into the components of the user interface model depicted in Figure 1. From this, the authors identified and summarized major patterns in which differences exist between the user interface of programmers.

Results

Forty-two users (38 recruited from a regional conference, 4 from telephone solicitation) completed the questionnaires and provided assessment on seven different pacemaker programmers. There was a balanced representation of users in terms of user education, place of employment, and programming experience (Table I). The profile was skewed in only two areas: (1) Users were potentially frequent users of pacemaker

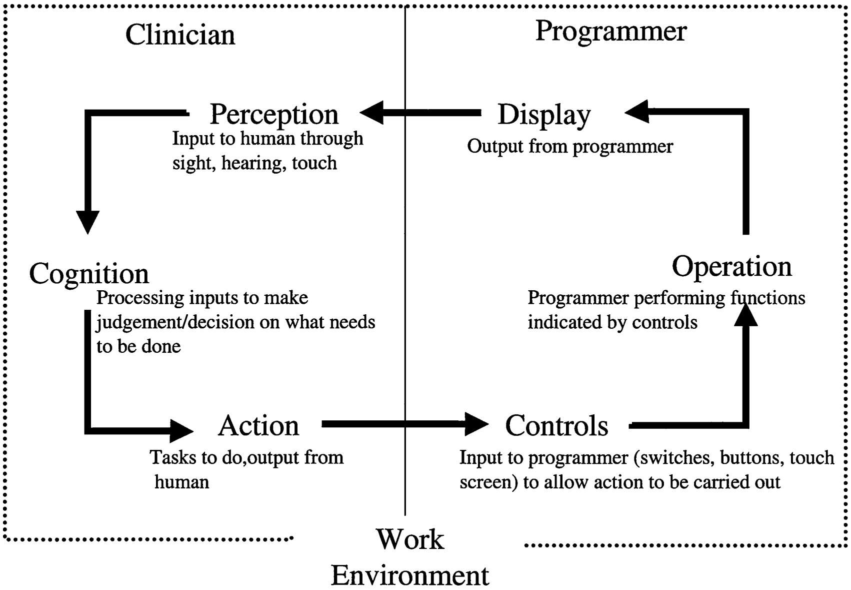


Figure 1. *Conceptual model of the clinician-programmer elements as adapted from a model of human-computer system.[4] This model helps us to conceptualize the relationship that exists between the clinician operator and the programmer. Designs in user interface of the programmer must adopt a systems approach. It must not only address the programmer aspect but also the limitations posed by the clinician aspect (human factor) and the work environment in which the interaction takes place. There is an analogous relationship between the programmer and the clinician in terms of input (controls, perception), processing (operation, cognition), and output (display, action).*

programmers in terms of clinic frequency (≥1 clinic per week) and programming volume (conduct programming >20 times per month). (2) Users have other clinical responsibilities that are non-device related, like heart failure clinic, vascular clinic, catheter laboratory intensive care unit or chemotherapy nursing, noninvasive ECG testing (stress, Holter, ECG, tilt test), research, billing, administration, and triage. The physical attributes of these seven programmers are listed in Table II along with the number of users who provided assessment.

Significant differences between programmers were found in overall user satisfaction, ease of programmer use, and in the various components of the user interface model (Table III). Some of the user comments were listed per verbatim in Appendix 2 as they provided some insight into the user's

Table I.

Profile of Users who Completed the Questionnaire

Characteristic	N	Category	n	%
Education	40	Nurse	23	58%
		Tech	16	40%
		Fellow	1	3%
Workplace	38	Community hospital	19	50%
		Academic hospital	17	45%
		Private clinic	2	5%
Responsibilities	36	Performs other duties	26	72%
		Dedicated to devices	10	28%
Device clinic volume	39	Weekly	34	87%
		Biweekly	4	10%
		Monthly or less	1	3%
Programming volume	39	>21/month	31	79%
		11–20/month	6	15%
		<10/month	2	5%
	N	**Median (range)**		
		Mean ± SD		
Programming experience	40	60 months (1–300 months)		
		88 ± 75 months		

Table II.

Programmer Attributes and Number of User Assessments

Programmer	Weight (kg)	Dimensions (L × W × T) (cm)	Display Screen (L × W) (cm)	Navigation Tool	Multiple-Color Display	Ability to Print Full Size Printouts	Year Launched In market	# User Evaluations Received
A	11.1	55 × 40 × 12	23 × 30.5	touch pen	yes	yes	2002	34
B	9.4	45 × 36 × 12	24.5 × 18.5	touch pen / screen	limited (black/red/blue on white)	yes, fanfold sheets	2001	23
C	8.5	45 × 34 × 9	16 × 21.5	touch pen / screen	limited (black/red/ on white)	yes	2000	17
D	9.9	30 × 40 × 15	12.5 × 19.5	touch pen / screen	no (yellow on black)	no	1995	10
E	8.8*	38 × 31 × 9	14 × 19	hard key buttons	no (black on white)	no	late 1980s	6
F	8.5*	41 × 31.5 × 9	14.5 × 19.5	touch pen / screen	no (black on white)	yes	1994	4
G	9.0	45 × 41.5 × 12	18.5 × 24.5	touch pen / screen	yes	yes	2000	4

* includes carrying bag and accessories.

frustration or dissatisfaction. There is a loss of the richness of information if the comments were summarized or grouped together. The authors incorporated keywords extracted from user comments and from expert assessments into the human to computer interface model which expanded this model by helping to identify relevant themes within each component as illustrated in Figure 2. These themes depicted not only what is important in each component but also where improvement efforts may be targeted. Furthermore, users did not always report important usability issues identified by usability experts through heuristic evaluation (Table IV).

In the section below, the authors elucidated some of the important differences between programmers by stating what the issue or problem is, where the differences are between programmers, and linking these problems to the usability principles that were violated. These problems likely accounted for the user's perception of usability differences between programmers.

Controls

Touch Pen/Touch Screen

Users preferred the flexibility of using a touch pen or touch screen to perform their follow-up tasks. Users and usability experts reported problems with the touch pen on programmer A, yet the screen cannot be activated by finger touch. Other programmers B, C, D, F, G have the flexibility of both touch pen and screen (Table II).

Emergency Button

There are differences in the labeling, location, and operation of the emergency button (Table V). Despite users performing their follow-up tasks on the touch screen, four of seven programmers do not have a software icon on the screen to allow activation of an emergency setting. This is a violation against the heuristics of: (1) The system should anticipate user action and user efficiency. In an emergency, the user will attempt to activate emergency settings through the screen from which they are working. (2) Recognition rather than recall, that is, the user has to remember that the emergency setting is activated by a hard key located on the console. Activation of the emergency setting should be available as a software icon on the screen and as hard button on the programmer console. Other potential error prone problems include users accidentally hitting the emergency button in programmer A, emergency setting that is not the maximum output, 1-step versus 2-step activation, and the presence of other similar buttons that may confuse the user like "PC shock," "Abort therapy" or "Stat shock," "Divert therapy" in programmers B, C.

Table III.

User Opinion of Programmer Performance: Differences Between Programmers A, B, C, D

Questionnaire Item	Theme	Ranked Mean Response Best			Worst	P Value
Ability to see the emergency button	Controls	A	C	B	D	0.0002
Ability to quickly activate emergency settings	Controls	A	C	B	D	<0.0001
Clearness of information displayed	Display	A	C	B	D	0.0007
Clearness of alert messages	Display	A	C	B	D	0.002
Is there too much information displayed?	Display	No	No	No	No	0.51*
Ability to find what you need on the screen	Display	A	C	B	D	0.0002
Ability to find what you need on the printouts	Display	A	B	C	D	0.009
Ability to interrogate pacemaker settings	Operation	A	B	C	D	< 0.0001
Ability to obtain relevant telemetry/counter data	Operation	A	C	B	D	0.0001
Ability to perform automatic threshold tests	Operation	A	C	B	D	< 0.0001
Ability to use demonstration software on programmer	Operation	A-C	N/A in *B D*			0.006
Ability to recognize and correct errors by the programmer	Operation	A	C	B	D	0.03
Ability to complete follow-up tasks on programmer within 10 minutes	Operation	A	C	B	D	0.003
Frequency of computer failure during patient follow-up	Operation	C	A	B	D	0.03
Frequency of communication/telemetry problems	Operation	A	C	B	D	0.003
Ability to carry the programmer as needed	Physical	C	B	A	D	0.003
Ability to use the programmer anywhere as needed	Physical	C	A	B	D	0.36*
Overall satisfaction with using programmer		A	C	B	D	< 0.0001
Overall ease of programmer use		A	C	B	D	< 0.0001

Ranking was based on mean responses from users, statistical analysis was based on the median responses. NA = no data; * = no significant difference; - = tied ranks.

Control to Restore Initial Pacemaker Settings

Users preferred the "Initial values" button on programmer B. This control enhances user efficiency by restoration of the initial parameter settings in one step. There is trouble locating this button or icon and its labeling is inconsistent between programmers: "Load Initial Values" in programmer C, "Recall" in programmer D or hidden under a submenu of the "GET," "Initial/Start of Session" in programmer A. Similar controls like the "GET" icon is next to a "SAVE. . ." icon which does not retrieve saved parameter settings but saves the current parameter setting. These differences in terminology and location of the icon placed heavy reliance on user recall and decreased user efficiency.

Display Screen

Users preferred a larger screen size. Smaller size made it difficult to select options and cluttered the information displayed. Display screen dimensions are shown in Table II. Information can be better organized by the use of popdown menus for extended parameter settings and these popdown menus should not be remotely located as they are in programmer C. Usability experts did not like the display screen in programmers B, D, their closely spaced options caused users to make wrong selections and redoing the task again (inefficiency).

Printout

Users and usability experts liked the full size printout of programmers A, B due to its aesthetic layout. Most programmers can be connected to an external printer for full size printing. However, some users were not aware of this capability. Printouts were hard to read on programmers C, D, and G due to poor lay out and small font size. Programmers C, F, G are particularly bad because they print over the perforating margins on small squares of paper. Programmer A prints to the same reading level as the user (i.e., horizontal printout). All other programmers (B–G) print vertically requiring users to strain their neck to read or to pull off the strips each time they want to read the printout. These problems violated usability principles by not anticipating user action; that is, users will want to read or look at the data as it prints without tearing it from the printer. Redundant information is printed that competes and decreases visibility of relevant information. For example, programmer

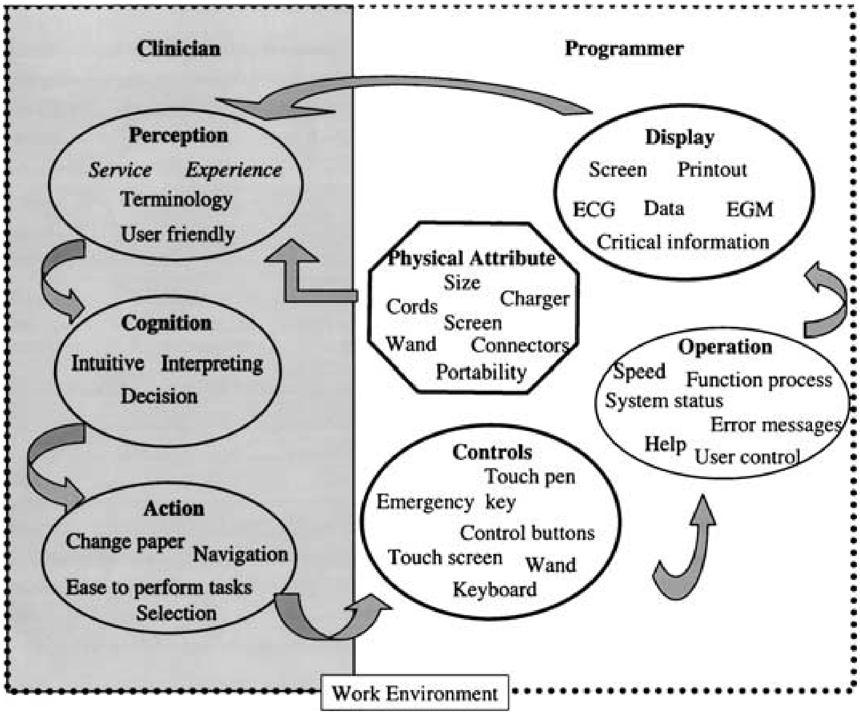


Figure 2. *The conceptual user interface model from from Figure 1 is expanded by inclusion of the keywords identified from users and usability experts. An additional component called Physical Attribute of the programmer element was fitted into the model based on user comments. This expanded model illustrated important themes and areas that can be targeted for improvement.*

A pinted out all the rate responsive parameter settings even when this feature was turned off. Users did not like the printout in small squares of paper as this presented inefficiency in terms of filing (cutting and pasting into patient chart).

Operation

Slow programmer operation or frequent telemetry errors may pose a safety issue, decrease user efficiency, and increase user frustration. Users commented that programmer B was slow in programming, retrieving data, and printing. While it was printing, users could not perform other tasks. Likewise, programmer D was slow in its response and frequently has telemetry errors. This may be a safety issue and inefficient, for example, if a user can not reprogram a setting because the programmer is still busy. Other programmers did not have problems in this regard. Programming and telemetry are nearly instantaneous and they can perform printing in the background while the user moves on to perform other tasks with the programmer.

Process of Conducting Tests

Complete instructions on how to conduct tests must be fully visible to the user before starting the test. The user's attention should be focused to one area of the display during the test. Usability experts commented that programmers B, C, D, F, G did not provide sufficient instruction to the user on conducting tests. This is a problem identified by usability experts and not reported by users likely because users have learned or trained on how to initiate the threshold tests. Instructions must indicate how to start and end the test before the user initiates the test. Critical information such as the voltage or pulse width decrement during threshold test should be displayed at the ECG level where user's attention is directed during the test. Otherwise, these issues are violating the heuristics of user efficiency, help and documentation.

Discussion

The present study identified differences in the user interface design of programmers impacting on user satisfaction and the perceived overall ease of programmer use. This translates to the user perception that certain programmers are easier to use than others. Further research is needed to determine if these differences affect patient safety, patient call-back, and user performance in conducting follow-up, error rate, or follow-up time.

From this study, there many examples of the user having to "make do" or adapt to the poor

Table IV.

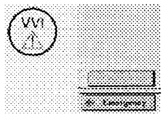
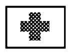
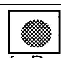
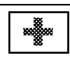
Some of the Usability Issues Identified by Heuristic Evaluation

Usability Issue	Rule Violation	Suggestion
Terminology: various terms, acronyms that are not consistent between manufacturers, e.g., AR vs APVS or AR vs ARS vs Ar; min^{-1}, ppm, bpm; [Program] vs [Programs]; Number vs No.	Match between system and world, recognition rather than recall	Establish standards across industry or provide definitions with each use to avoid user confusion.
Date format: poor in *A* and *F*, e.g., 02/04/03 or 02.04.2003 or 02.04.	Readability, Match between system and world	Establish standard convention of e.g., dd-month-year as 02-April-2003.
Inconsistent location of controls: e.g., [PRINT] in *B* only as hard key on console, no software icon, while in *A* it is only in software icon and in *D*, both hard and soft keys are present.	User efficiency, recognition rather than recall, anticipate user action, consistency and standards	Establish standard convention in location of critical controls such as the [Interrogation], [Emergency], [Print], [Program], on/off switch, restore initial settings. They should be visible and not hidden inside submenus.
A control with more than one function: e.g., [Start test] changes to [End test] after its activation, user may not look again or may not be aware how to end test until they've started it.	Recognition rather than recall, system should anticipate user action	Controls should serve only one function and its function should be clearly marked and stated.
More than one control that performs the similar function: in *G*, there are three places within a screen that has printing function; in *A*, there is [Save to disk], [Device Memory Report], [Save...].	Consistency and standards	Controls should be precisely labeled, there should not be more than one control within any active screen that does the same thing.
Inappropriate use of control size: confuses user as to which one to press to start threshold test or to activate emergency setting e.g., in *F*, [Decrease test Amplitude] is bigger than [Start test]; in *C*, [Divert therapy] is bigger than [Stat Pace] or [Stat Shock].	Fitt's Law	Large visible buttons should be used to draw user's attention to complete an immediate task.
Indiscriminate use of color in coding controls: in *C*, [Divert therapy] is the larger red button, while [Stat Pace] is small orange and [Stat Shock] is yellow; in *A*'s ICD emergency screen..[Deliver] shock or [Program] pacing are blue instead of red; in *G*, yellow was used for action controls...six definitions of their own color code that users are not likely to remember.	Fitt's Law, consistency and standards, match between system and world	Consider standardizing color of controls to convention (Occupational Safety and Health Agency): red-danger, orange-warning, yellow-caution, blue-notice, green-safety; Avoid overuse as it may cause more confusion.
Inconsistent feedback of system status to users: programmers to various degrees are not as good at informing users when commands have been executed or inadequate messages to inform users of problems but without info on where problem is or steps that needs to be taken e.g., in *F*.	Visibility of system status, help users diagnose and recover from errors	Some programmers use a combination of messages and audio tone to let user know whenever commands, selection, are executed. Error messages should identify problem and provide instruction to user on what to do next or propose solution.
Screen savers: in *D* and *G*, confuses user on what to do, may cause panic in thinking data is lost or result in inadvertant programming; does not serve any useful purpose in patient follow-up	Error prevention	Avoid use of screen savers

See text for further discussion. Italicized letters referred to programmers. ICD = implantable cardioverter defibrillator.

Table V.

Emergency Controls on Programmers to Activate Emergency Pacing in Pacemakers

Programmer	Location of Hard Button	Color of Button	Label	Hard Button on Wand	Software Emergency Icon	Activation via Hard Button
A	Bottom left side of display screen	Red		No	Yes (bottom left side of screen)	1-step immediate activation with confirmation message.
B	Top right side of console	Red	Emergency VVI	No	No	1-step activation with confirmation message.
C	Top right side of console	Orange	Stat Pace	No	No	2-step with message requiring user to depress key again. ** Not max output of device, only to 5 V.
D	Bottom left side of console	Red	 Stat VVI	Yes	No	1-step immediate activation with confirmation message and parameters page displayed.
E	Mid top left side of console	White with red cross		No	No	1-step immediate activation by press and hold hard key for 2 seconds, followed by confirmation message.
F	Bottom left side of console	Red	 Safe Program	Yes	Yes	1-step immediate activation and parameters page displayed.
G	Top right side of console	White with red cross		Yes	Yes (top right side of screen)	1-step immediate activation with confirmation message of successful programming. ** Not max output of device, only to 5 V.

Unless otherwise noted, emergency pacing is always at maximum output of the device, typically 7.5 or 8 V.

designs or performance of the programmer. Research in other types of computer-based medical equipment like drug infusion devices has shown that users cope with bad designs by tailoring their activities and modifying their procedures.[12] These coping strategies are not ideal because they do not truly rectify the system problem and may fail at times of high demand or stress or introduce new problems to the system.[12] The ideal user interface should "speak" the user's language, not dictate what the user should "speak" or have so much variations that a user has to remember. The interface should help the user by interaction through help menus, dialogue boxes, and constructive messages that identify potential problems and provide possible solutions. Users may have learned to cope with the deficiencies of the programmer interface and hence the underreporting of some the important issues identified by the usability experts.

Despite the use of computer-based programmers for device follow-up, many aspects of their user interface designs do not conform to human to computer usability principles. For example, poor feedback to users on what the system is doing, differing terminologies between programmers, critical controls that looked and operated differently, acronyms that appeared the same but mean different things, and heavy reliance on the user to recall facts, features, steps, and sequence of operation. This burdens the users and leads to inefficiency. It is error prone as the user attempts to navigate through various screens or press different controls by trial-and-error to find what they need or to conduct a test.

Implantable device technology is becoming increasingly complex with many advanced programmable features, diagnostics, and expanding applications to different patient populations.[13,14] Each manufacturer in the industry is driven by the need to patent their own proprietary technology, features, capabilities, and terminology to control their use by competitors.[15] These issues increase the challenge in the user interface design of programmers. A "systems approach" as depicted in Figure 1 addressing the programmer and the clinician operator becomes more essential to achieve a useable interface. The complexity of pacemaker programmers greatly exceeds that of other medical devices like patient-controlled analgesia pumps or drug infusion devices, thereby making the application of human factors engineering principles even more important.

Despite this study's focus on problems with user interface, the authors recognized that each programmer has its merits and strengths. It was not their intention to say one programmer is superior to another. This is the reason why they chose to code manufacturers rather than to list their names. The goal of the study was to point out specific usability issues so that manufacturers can consider them in improving their platforms and interfaces. Programmer platform or software upgrades by the manufacturer are important to continuous improvements. In contrast to other programmers, programmers D and E have not undergone any platform upgrades since the mid-1990s when their manufacturers were acquired by new owners who chose not to upgrade nor incorporate D and E's devices into their programmers.[15] The lag in technology of programmer D is evident from the problems reported by users and usability experts.

The authors believe it is important to establish industry guidelines in the labeling, location, and operation of critical controls and standardized terminology. Activation of the emergency settings of a pacemaker served as a prime example of the inconsistencies that existed between the different programmers. There were attempts to address this issue through discussions to develop a "universal" programmer since 1983 but none of the manufacturers were keen to take this route.[16] The authors recommend that conventions be established for critical features in all programmers. Without conventions it will be technology driving the users rather than the other way around!

Study Limitations

This study is a preliminary assessment into the usability of programmers and relied partly on the subjective opinions or attitude of the users. Sufficient data was not available to further explore the perception and cognitive themes of the model. It is hard to separate out the human performance or the impact of learning or experience on usability. The study attempted to balance this by heuristic evaluation through usability experts who focused on the interface aspect and have little training on pacemaker follow-up or programmer. In a study on a simple telephone billing inquiry task, it was concluded that heuristic evaluation produced too many "false positive" findings compared to usability testing. That is, usability experts identified issues that were irrelevant to user preference or performance levels on the risk.[17] The authors believed that in this study, findings from heuristic evaluations tended to corroborate those from users. In addition, usability experts identified important safety, efficiency, and effectiveness issues that will help to make programmers more "usable." As evident in Table II, the programmer platforms in this study were launched at various times. This study excluded programmers from quantitative analysis based on the number of evaluations received rather than based on the launch year. The authors felt this is more reflective of what users encountered in clinical practice.

Conclusions

Programmer user interface designs do not consistently meet user needs or general usability principles. This may impact on the safe and effective use of the programmer in following patients with devices. Guidelines in programmer design should be established particularly with respect to labeling, location, and operation of critical controls.

Acknowledgments: The authors thank Ms Gillian Hillel, BASc, MASc candidate and Ms Andrea Cassano, BASc, for their insightful usability evaluation of the programmers, the volunteers who completed the questionnaire and provided many useful comments, and Ms Tracy Biggs and Ms Vanessa Hung for kindly loaning some of the programmers for this study.

Appendix I. Description of the 15-Point Heuristic Rules

Heuristic	Description
Visibility of system status	System keeps users informed about what's going on.
Match between system and world	System speaks user's language and follows real-world conventions. Information should appear in a natural and logical order.
Consistency and standards	Things that act the same should be visually consistent, things that act differently should be visually inconsistent. Style standards should be followed.
System should anticipate user action	System should anticipate user's wants and needs. Do not expect user to search for or gather information or evoke necessary tools. Bring to the user all the information and tools needed for each step of the process.
User control and freedom	There should be clearly marked exits, system should support "undo" "redo."
User efficiency	Aspects of the interface that allows user to quickly and with minimal effort perform their tasks e.g., clear and well-organized menus.
Error prevention	Careful designs to prevent errors. Make use of good error messages.
Help users diagnose, and recover from errors	Error messages should be expressed in plain language, precisely indicate the problem and constructively suggest a solution.
Help and documentation	Although it is better if the system can be used without documentation, it may be necessary to provide help and documentation. This information should be easy to search, focused on the user's task, list concrete steps to carry out and not be too large.
Recognition rather than recall	Make objects, actions, options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable.
Flexibility and efficiency of use	Accelerators (unseen by novice users) may often speed up the interaction of the expert user such that the system can cater to inexperienced and experienced users. Allow users to tailor frequent actions.
Aesthetic and minimalist design	Dialogues should not contain information that is irrelevant or rarely needed. Every extra bit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.
Fitt's Law principles	Use large buttons for the most critical functions. Use the corners and edges of the screen to "pin" active areas of the critical buttons to allow for quick easier activation.
Readability	Text that must be read should have high contrast. Favor black text on white or pale yellow backgrounds. Avoid gray backgrounds. Use font sizes that are large enough to be readable. Favor particularly large characters for the actual data you intend to display rather than for the labels and instructions. Eg. The label "Last Name" can afford to be somewhat small but the actual last name entered or displayed should be clearly readable. This is particularly important for numbers.
Color blindness	Any time color is used to convey information in the interface, clear secondary cues should also be used to convey information.

Adapted from Nielsen 1994[10].

Appendix II. Additional Examples of User Comments, Dislikes, Problems, and Wishes for Programmers

Theme	User Comments
Operation	• Unable to use analyzer and program device(A), paper jams a lot(B), • Very long delay when threshold test stopped(C), • On screen help is problematic(B), EGMs not switching intuitively when doing A's or V's(B), • During threshold testing cannot see on printout where loss of capture occurs (not identified V or ms) also have to watch ECG and down below..the decrement occurring(C), • Poor telemetry-communication errors+++; poor EGM(D), • Telemetry errors(A,G), • Printer very temperamental, doesn't work after being carried outside in winter(B), • Cannot use due to electrical interferences in ICU or ER (A,B,C), • Diagnostic data..long interrogation time(G).

Continued

Appendix II. *Continued*

Theme	User Comments
Physical attributes	• Programmer heavy, awkward, cumbersome (A,B,D,G), • Is keyboard necessary; it takes up a lot of space(A), • Programming head awkward to position(C), • Since I stand to do a f/u..new larger screen almost too big... I'm short!(A), • LCD display doesn't stay in position selected(A), • Cords carried separately(B,C), • We have a specific cart to move programmer between departments(A,B,C,D,E,G), • The screen does not stay up..it slowly goes down while programming(A), • Need power..no charger(A,C), • Outlet cord is too short(A,B), • Hard to pack up(A,E).
Perception	• I have less experience on this programmer but I find the information slightly confusing(B), • I'm still learning so much about each pacemaker parameters..need standard language among companies for same thing(A), • Due to different wording sometimes difficulty finding program functions(C), • I like the programmer, service, and support(G), • I am becoming familiar with this programmer and am finding it easier to use each time(B), • Not user friendly..terminology different to other programmers..have to think more when using it..screen too busy(B,C), • Confusing use of programmer(E).
Cognition	• Difficult to change parameters if you violate intervals i.e., LRL, URL, AVDelay(B), • Difficult to manage different tabs and screens; not clearly intuitive(C).
Action	• Replacing paper at the end of the roll(D), • Can't adjust rate thresholds—are done at (temporary) (E), • Limitations in settings for threshold testing(G), • Changing printer paper(A).
User's wish list	• Finger touch screen would be nice (A), • I wish for less cords, wireless transmission(A,B,C), • That I won't have to use it much longer as pacemakers are replaced!(D), • Make this programmer lighter(A), • Wish that these pacers could be followed with new company's programmer(D,E), • Load up the software faster, especially in an emergency(A), • Wish the information in the screen could be easy to decipher(B), • Get rid of it!(D), • Continuous ECG printout/not just "freeze" rhythm strip(B), • Wish other manufacturers would use same terms for same algorithms(A), • Wish for bigger screen, better printouts(C), • Needs Internet access for remote center management.(B,C), • The pen that activates the screen is very fragile please modify(A).

See text for further discussion. Letters within brackets indicate programmer to which comment applies to.

References

1. Shannon K. Use of implantable cardoverter-defibrillators in pediatric patients. Curr Opin Cardiol 2002; 17:280–282.
2. Bryce M, Spielman S, Greenspan A, et al. Evolving indications for permanent pacemakers. Ann Intern Med 2001; 134:1130–1141.
3. Saksena S, Krol R, Kaushik R. Innovations in pulse generators and lead systems: Balancing complexity with clinical benefit and long term results. Am Heart J 1994; 127:1010–1021.
4. Proctor R, Van Zandt T. Human factors in simple and complex systems. Needham Heights, MA, Allyn and Bacon, 1994, pp. 1–14.
5. Faulkner X. Usability engineering. London, Macmillian Press Ltd., 2000, pp. 1–20.
6. Sawyer D. Do it by design: An introduction to human factors in medical devices. US Department of Health and Human Services, Office of Health and Industry Programs, 1996.
7. Lin L, Isla R, Doniz K, et al. Applying human factors to the design of medical equipment: Patient-controlled analgesia. J Clin Monit 1998; 14:253–263.
8. Association for the Advancement of Medical Instrumentation. Human factors design process for medical devices. ANSI/AAMI 2001. Arlington, VA, AAMI, 2001.
9. Garmer K, Liljegren E, Osvalder A, et al. Arguing for the need of triangulation and iteration when designing medical equipment. J Clin Monit 2002; 17:105–114.
10. Nielsen J. Enhancing the explanatory power of usability heuristics. Proceedings of the ACM CHI '94 Conference, Boston, MA, April 24–28, 1994, pp. 152–158.
11. Levine P. Guidelines to the routine evaluation and follow-up of the implanted pacing system. Sylmar, CA, Pacesetter Systems Inc., 1993.
12. Obradovich J, Woods D. Users as designers: How people cope with poor HCI design in computer-based medical devices. Human Factors 1996; 38:574–592.
13. Obias-Manno D. Unconventional applications in pacemaker therapy. AACN Clin Issues 2001; 12:127–139.
14. Hayes D, Lloyd M, Friedman R. Cardiac pacing and defibrillation: A clinical approach. Mount Kisco, NY, Futura Publishers Inc., 2000.
15. Jeffrey K. Machines in our hearts: The cardiac pacemaker, the implantable defibrillator and American health care. Baltimore, MD, The John Hopkins University Press, 2001, pp. 271–273.
16. Boal B. Emergency reprogramming of cardiac pacemakers. Pace 2000; 23:1323.
17. Bailey R, Allan R, Raiello P. Usability testing vs. heuristic evaluation: A head-to-head comparison. Proceedings of the Human Factors Society, Atlanta, GA, Oct. 12–16, 1992, pp. 409–413.