Making Information Technology a Team Player in Safety: The Case of Infusion Devices

Christopher Nemeth, Mark Nunnally, Michael O'Connor, P. Allan Klock, Richard Cook

Abstract

Objective: To fulfill the promise of information technology in health care, automation must be made into a "team player." **Methods:** Observational research in both the laboratory and field focused on how subjects program infusion devices. These programming activities were examined in detail for a set of tasks, using experienced clinicians as subjects. Observations were validated using U.S. Food and Drug Administration (FDA) incident reports and field studies. **Results:** The laboratory study shows that difficulties arise from poor coordination between the operator and the device as a result of device complexity. These findings are consistent with reports of device programming problems in the FDA's Manufacturer and User Facility Device Experience (MAUDE) database. **Conclusions:** Problems with infusion devices are the result of difficulties that operators have with complex devices and their programming. Opportunities to improve device reliability lie primarily in improving the interface design, such as by making the device's state evident and improving menu structure navigation.

Introduction

Advanced information technology (IT) is at the core of many proposals to improve patient safety. While IT can benefit health care, it can also impose additional workload and cognitive demands on operators and become a source of new forms of failure. If the promise of IT is to be fulfilled, it is essential for automation to be made into a "team player" that demonstrates a "fluent, coordinated interaction between human and machine elements of the system."[1]

Intravenous medication

Potent, short-acting intravenous medications form an important part of critical care. Previously, the use of fixed long-acting drugs, such as intramuscular morphine injections, limited intravenous medication to comparatively benign fluids, such as saline. The drug's effect on the patient was gradual and minor variations in dosing rates could be tolerated. Current medical practice makes use of a larger number of pharmaceutical agents. Newer agents are more potent and faster in onset and offset of their action than were their predecessors. Some agents, such as chemotherapeutics for cancer, require complex, changing infusion schedules. Deliberate administration to body compartments other than the blood (such as the spinal fluid) further complicates infusion practice. The pharmacology of these agents, as well as different practice patterns, have driven the need for multiple, carefully controlled infusion schemes.

Infusion devices

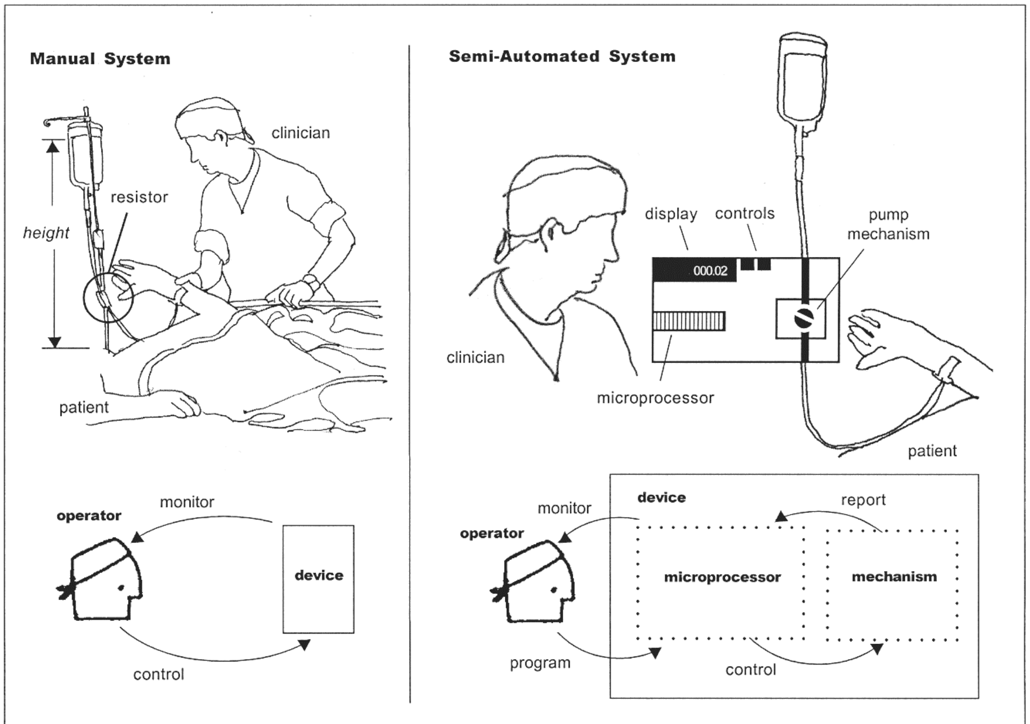
Historically, care providers used a bag of medication that relied on gravity to draw the fluid through flexible tubing to the patient. Droplets of the fluid could be seen to form and drip into the tubing. This “drip” was regulated by a simple manual fluid-flow resistor placed in series with the infusion tubing. Practitioners observed the droplet formation rate and adjusted the resistor to achieve the proper dosing scheme. The advent and proliferation of potent, short-acting intravenous agents for use in anesthesiology and critical care medicine required precision and accuracy, and the perception was that the simple control loop of a gravity-fed drip could not provide it.

The advent of small, cheap microprocessors led to the development of infusion pump systems that were capable of performing consistently and with high accuracy. These programmable devices were introduced in the interest of improving control over infusions. Most infusions in U.S. hospitals are now provided by such devices.[2] In effect, this adds another member to the acute care team that needs to cooperate with clinicians. Unfortunately, these devices do not always cooperate well. Practitioners have to perform additional work to coordinate and program the devices in order to make the pumps equal participants. Programming these devices presents unforeseen complications, with significant implications for patient safety.

Figure 1 compares manual and semi-automated approaches to infusion. In the manual arrangement, a clinician observes fluid drip directly and controls its rate using a mechanical resistor. In the semi-automated arrangement, the clinician observes the display that reports on microprocessor status and presses controls to change the microprocessor state. The microprocessor controls and monitors the pump mechanism, which in turn pumps fluid to the patient.

Electronic infusion devices make it possible to precisely deliver fixed volumes of fluids. The automation can be used in several ways: (1) to deliver medications (fluid bolus); (2) to continue medication delivery while unattended; (3) to administer infusions of short-acting drugs at a constant rate, or; (4) to titrate particular kinds of medications to desired effect such as vasopressors, which are used to control blood pressure. The current generation of infusion devices provides clinicians with aids to calculate medication doses, as well as a variety of alerts and alarms that are related to their functions. Some models include memory that contains a library of approved medications, in the interest of making programming more efficient. Pump models are available that can handle individual medications or “multi-channel” versions that can infuse more than one medication at a time. As many as 10 pumps have been seen in use at one time for a single intensive-care unit patient.

Figure 1. A comparison of manual and semi-automated infusion systems



Copyright © 2004 Cognitive Technologies Laboratory. All rights reserved.

Manufacturers typically represent their infusion pump products as precise, accurate, reliable, and flexible. Pumps are described as being able to automate multistage processes, such as piggybacks and step infusions, to compensate for the staff's limited abilities to pay attention, calculate doses, and adjust medications. Seeking economies of scale, manufacturers tend to develop a single design that can be used to serve many uses throughout the market. This tends to produce pumps that are capable of many functions, are light in weight, and that consume a minimal amount of power.

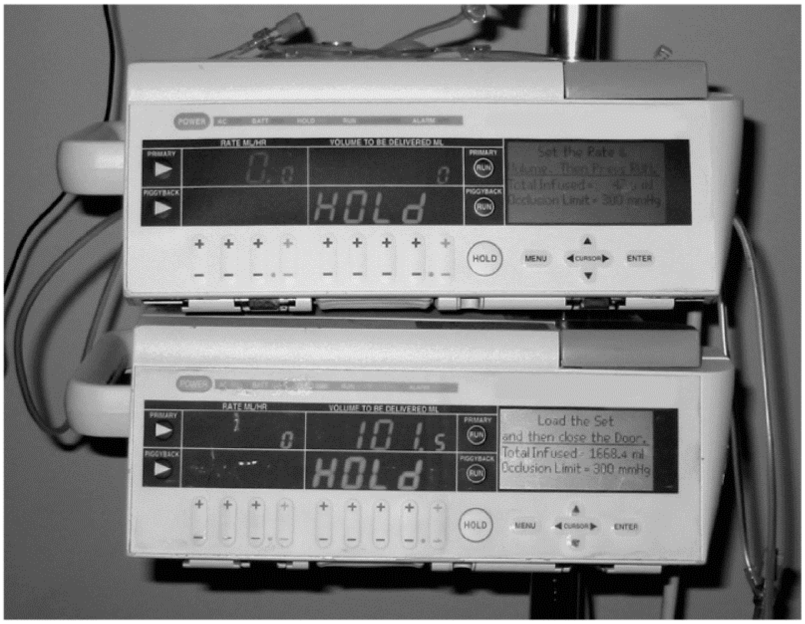
As an FDA Class II device, infusion pumps are subject to special controls, such as mandatory performance standards. Adverse events including device-related deaths and serious injuries must be reported to the FDA and the manufacturer. The FDA Center for Devices and Radiological Health (CDRH) maintains a Manufacturer and User Device Experience (MAUDE) database that serves as a clearinghouse for adverse event reports involving medical devices, including infusion pumps.

Previous studies in our lab have demonstrated that modern infusion devices feature multiple modes of operation; require substantial operator programming; and contain layered, nested menus with complex branching. Interface designs provide little useful feedback about the state of program entry, the history of operation, or the past or present state of the infusion device. These are not characteristics of "team players." These traits cause experienced device operators to frequently become lost while programming, have difficulty tracking device states, and misinterpret device function. Representations of the device's current state and the paths that are available to reach goal states are often ambiguous. This forces the practitioner to develop coping strategies that are effective, yet are vulnerable to failure in actual use. In a study of the effect of information technology on health care practice,[3] we sought to answer three questions: How is the programming of a specific infusion pump structured? How do the users accomplish programming tasks within this structure? How do existing incident reports help to describe adverse events in terms of infusion device programming characteristics?

Methods

The first phase in the study sought to understand all of the possible routes that subjects could take while programming a pump. The pump model used in the study (Figure 2) is commercially available, representative of current technology, and hundreds of them are used daily at the research site (University of Chicago Hospitals).

Figure 2. The experiment apparatus: two infusion devices



As a first step, members of the lab staff systematically programmed the infusion pump to explore all possible programming permutations. This pump allows multiple pathways to enter the data that is needed to begin an infusion and provides multiple modes for infusion. Pump programming structure, or "menu space," was then represented as a state diagram that depicted all possible programming routes. The state diagram was used during later observation and analysis, making it possible to trace each subject's route through the interface architecture.[4] The diagram also enabled the team to determine the minimum number of keystrokes that were required to reach the desired state for each task. The study's second phase was a laboratory experiment in which subjects operated the pump. In the third phase, field observations were performed to validate lab results. A fourth phase reviewed the MAUDE database for all adverse event reports related to the infusion pumps under study.

A sample of 14 anesthesiologists and 26 intensive care unit (ICU) nurses from the same research site were invited to participate in individual programming sessions. All sample members had significant experience with this pump, ranging from 11 months to 5 years. Using Verbal Protocol Analysis (VPA), the subjects were asked to describe their thoughts while they performed four pump programming tasks, listed in Table 1, on the apparatus shown in Figure 2. This device has two options for programming: a dose mode at the left side of the interface allows direct entry of information, such as fluid volume and drug concentration; and a library mode on the right side provides a roster of drugs and related information that can be used to build a program.

Tasks ranged from simple to slightly more complex, and were similar to subject experiences in their daily work. Unlike dopamine, nesiritide and its stock, concentrations are not in the pump drug library. This required the user to abandon library-based programming and enter all of the data manually using dose mode.

Table 1. Pump programming tasks used in the study

a)	Check a running dose of the drug dopamine (a premix concentration of 400 mg in 250 ml) that is set to run at 3 mcg/kg/min for a 75 kg patient.
b)	Change the same dopamine infusion to a rate of 2 mcg/kg/min.
c)	Set up and run a second pump (powered down) to deliver 1 liter of intravenous fluid over 8 hours.
d)	Change the pump from scenario 3 to now deliver dopamine (400 mg/250 ml) at 3 mcg/kg/min in a 65 kg patient.
e)	Change the same pump to deliver a premix of the drug nesiritide at a rate of 1 mcg/kg/min (a higher than normal dose).*

* This fifth question was added to the nurse sample test and later used for qualitative analysis.

Sessions were recorded on audio- and videotape, and the videotapes were analyzed to the level of individual keystrokes. These analyses were compared to the state diagrams of the device menu structure. User programming was analyzed for efficiency, choice of mode, and sequence selection.

Results

The state diagram made it possible to see the most direct route that could be taken to program a desirable goal state. Lab staff compared user programming keystrokes with the state diagram. Any keystrokes the practitioner entered while programming the device that moved the program sequence in the direction of the goal state were coded as "goal directed keystrokes (GDK)."

We compared the minimum number of possible keystrokes for the task (from the state diagram) with total keystrokes the subject actually entered to show programming efficiency. This assumed that subjects who knew "where to go" while programming would have a greater percentage of keystrokes that were goal-directed.

Successful programming performance has clinical consequences. Comparing members of the sample with averages for the sample gives a fair assessment of subject performance. Finding central tendencies of subject performance is less revealing than understanding how subjects varied in the way they programmed pumps. In particular, we examined the correlation between experience and performance.[5]

The limited number of keys gives the controls for this particular pump a simple appearance. This simplicity, though, requires that keys have several different functions, depending on device state. These functions are not necessarily obvious during programming and it is not apparent which keys are active at any given point in time. Screen and cursor manipulations linked to functions create the possibility for unintended changes to the device program. The pump operator moves a highlight cursor over various options on an LCD screen to perform programming steps. Options could be chosen or changed when highlighted, although other options that were available were often not displayed. Extra key presses offer an increased opportunity for inadvertent program operation. Although no subject in our study performed an unintended operation that altered the final results of the infusion scheme, there were several instances of operations that occurred without the knowledge of the user. Additionally, screens can be changed by certain commands. There are only 10 keys available for power up/power down, run, program, and input selections. An additional 11 keys are available to enter numerical data. As a result, each key may have several possible functions depending on the device state. Not all keys are active in all states and only some of the functions are be available at any particular moment. It is sometimes apparent which keys are active by reading the text on the screen. The caption "Press HOLD: to Start Over" is one example. At other times there is no indication as to which keys are active and which are not. Several manipulations of the screens or highlight cursor are linked to functions. For example, cursor movement in a dose calculation screen changes the relationship between dependent and independent variables. In some cases, the user controls the progression between steps in the programming process. In other cases, data must be accepted into the device's memory in order to proceed.

Table 2. Pump sample and programming performance results

Anesthesiologists							
Subj	**Exper (pract)**	**Exper (pump)**	**%GDK Task I**	**%GDK Task 2**	**%GDK Task 3**	**%GDK Task 4**	**Mean %GDK**
2	3	3	25	71.4	41.5	81.8	54.92
3	2	2	33.3	100	69.2	93.3	73.95
4	3	3	53.6	71.4	91.2	88.9	76.27
5	4	4	40	100	86.5	93.3	79.95
6	3	3	46.5	73.3	90.6	97	76.85
7	1	1	39.1	66.7	94.7	41.3	60.45
8	3	3	55.6	93.8	100	83.1	83.12
9	3	3	4i	100	100	96.3	84.07
10	11	5	35.3	72.7	92.9	74.4	68.82
11	3	3	72.7	90	32.6	90.3	71.4
12	4	4	25	100	90	88.9	75.97
13	5	5	100	87.5	90	100	94.37
14	12	5	61.5	83.3	81.8	88.6	78.8
15	14	7.6	100	100	69.1	69.17	
mean 5.07 3.5							
Intensive care unit nurses							
Subj	**Exper (pract)**	**Exper (pump)**	**%GDK Task I**	**%GDK Task 2**	**%GDK Task 3**	**%GDK Task 4**	**Mean %GDK**
16	5	5	95.45	85.71	85.71	91.11	89.49
17	13	5	100	85.71	75	100	90.17
18	13	4	100	66.66	100	68.42	83.77
19	2.5	0.92	100	85.71	75	100	90.17
20	4	1.5	100	88.88	57.14	95.83	85.46
21	5	2	100	85.71	100	83.33	92.26
22	6.5	3	90.9	100	92.3	97.05	95.06
23	10	3	73.68	83.33	61.53	94.11	78.16
24	20	5	80	85.71	95.23	100	90.23
25	7	5	30.76	83.33	97.36	90.9	87.11
26	14	5	71.42	100	100	81.48	88.22
27	2.5	1	100	85.71	52.94	80.55	79.8
28	18	5	60	87.5	100	100	86.87
29	22	5	100	100	35	100	83.75
30	8	2	88.88	100	31.53	93.33	78.43
31	1.5	1.5	71.42	85.71	86.66	95.65	84.86

Table 2. Pump sample and programming performance results, cont.

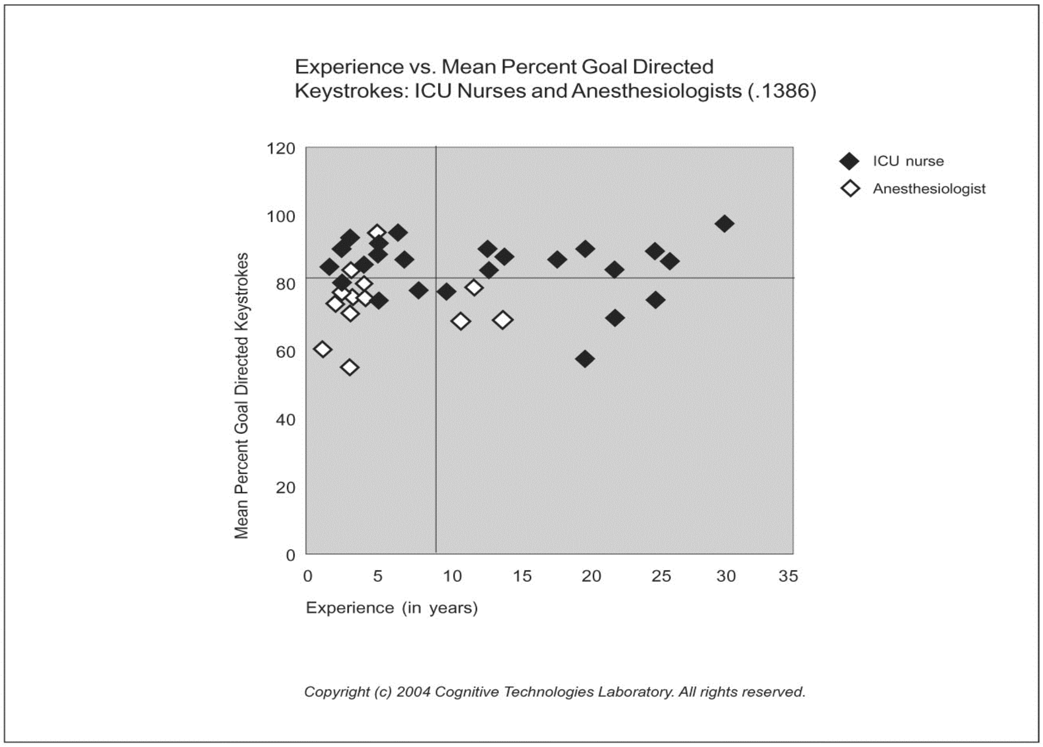
Intensive care unit nurses							
Subj	**Exper (pract)**	**Exper (pump)**	**%GDK Task I**	**%GDK Task 2**	**%GDK Task 3**	**%GDK Task 4**	**Mean %GDK**
32	3	1.5	100	87.5	100	84.84	93.08
33	26	2	83.33	75	85.71	103.12	86.79
34	30	5	100	100	90	100	97.5
35	2.5	2	100	85.71	36.76	87.5	77.49
36	5	1	61.53	75	76.47	86.11	74.77
37	20	5	0	80	50	100	57.5
38	22	5	20	83.33	81.25	96.55	70.28
39	25	5	93.33	90.9	17.5	100	75.43
40	25	5	90.9	66.66	100	100	89.39
41	12	2	84.61	90.9	45.2	96.55	79.31
mean 12.40, 3.36 Group mean 9.78, 3.41, 80.87							

Table 2 shows each subject's years of experience as a health care practitioner (pract), and with operating the pump in this study (pump). It shows the percent of goal-directed keystrokes (%GDK) for each subject and the mean %GDK for all four tasks. Correlation (Pearson *r*) between years of experience as a practitioner and mean percent goal-directed keystrokes were: anesthesiologists, -0.001; nurses, -0.08; and entire sample, 0.13 (Figure 3). Correlations between years of experience using this pump and mean percent goal-directed keystrokes were: anesthesiologists, 0.37; nurses, 0.07; and entire sample, 0.07. None of the correlations reached a level of significance.

Figure 3 illustrates the correlation (Pearson *r*: 0.13) of sample member experience in years with the mean percentage of goal-directed keystrokes that each subject used in order to accomplish all of the four tasks. The very low correlation indicates that even as practitioner experience increases, their ability to use the pump does not improve.

Our review of state diagrams for the pump showed that a subject could accomplish all four tasks using a minimum of 41 keystrokes, resulting in a minimum of 1,626 keystrokes for the sample. However, lab study results showed that sample members used a mean of 3,796 total keystrokes. Of those keystrokes, 2,640 (69.5 percent) were goal-directed. Comparison between minimum keystrokes and total keystrokes shows that subjects entered 57.1 percent more keystrokes than were necessary to accomplish the tasks.

Figure 3. Experience as a practitioner versus mean percent goal directed keystrokes



Copyright (c) 2004 Cognitive Technologies Laboratory. All rights reserved.

Discussion

The complexity of menu space and the structure of the infusion device interface makes operating the device difficult. The inefficiencies that subjects experienced represent the difficulties that are involved in navigating the many programming pathways that are available. Behaviors such as power-downs (an occasion in which individual turns electrical power off to clear the interface buffer and restart programming) suggest that subjects frequently encounter fruitless pathways and must exit from them in order to complete a successful programming sequence. Our subjects were usually able to successfully complete the tasks given to them, but frequently took circuitous routes to do it.

In actual operation, the programming of an infusion may begin at different device states. Subjects were presented with slightly different default system states and therefore had different starting points for a task depending on the system state at the end of the previous programming task. This makes it difficult to identify a keystroke as an “error.” In addition, the design allows users to navigate the menu in different ways. A subject could arrive at the correct system state using any number of approaches. Understanding performance in programming an infusion pump has more to do with whether the subject reached the correct state and the route they took to get there, rather than how many right or wrong keystrokes were entered. It would be unfair to compare subjects solely on the basis of time to complete programming task or total keystrokes because some were offered pumps in default state that varied. Some subjects began their task with a pump state that was very close to the desired state while others were quite far away. As a result, variation in *total* keystrokes had little to do with user performance. That is why coding percent goal-directed keystrokes mattered.

Conclusions

The conclusions that can be drawn from this study cover the methods that were used to understand these complex devices; what has been learned about pumps and their programming; what implications this may have for adverse event reporting; and what further work might shed light on the topic.

Infusion devices

Scoring individual keystrokes required a substantial working knowledge of the capabilities and restrictions in the various programming pathways. As a method that has been proven to assist analysis of other complex systems,[6] finite state diagrams permitted us to fully describe the inner workings of the infusion device before lab and field studies. Creating diagrams of the pump menu structure revealed modes and states, as well as device features that were enabled at each step in a programming pathway. Without such maps, it would have been difficult to know how subjects got lost or found their way back to a desired goal state.

We had previously observed that infusion device complexity lies hidden beneath layered, nested menus with irregular branching.[7] Findings among the 40-member sample confirmed our hypothesis that this would have an effect on programming performance by clinicians. The complexity of the menu structure, the *menu space* of the device, appears to defy any attempts at mastery. Even the most skilled users appeared to have a working knowledge of just a small portion of the pathways that made becoming "lost" very likely. This suggests that significant problems with navigation through the device menu space exist and are related to the interface design. The problem arises, in part, from the many fluid delivery modes the device offers, the small screen size, the limited number of controls that are used, and the variety of uses for the device.

No operator failed to reach the correct goal state in each of this experiment's four assigned tasks. However, none of the findings tells us there is one group or individual that can use the device well. How *do* operators know how to make the device work? Subject comments from this first sample indicate practitioner cognitive work practices that deserve further attention. For example, pumps are designed to administer drugs using parameters that do not match the way the clinicians think about infusions. Operators first figured infusions in terms of flow rate (ml/min), then needed to convert to other parameters (mg/kg/min, mcg/kg/hr) in order to program the infusion. In another example, operators appear to develop personal strategies to reach desired goal states. These strategies may work in the lab but are vulnerable to other influences when operating pumps in actual conditions.

Relationship between studies of devices and adverse event reports

Adverse incidents with infusion devices may have severe consequences. In spite of technological progress in the practice of infusions, failures and adverse events are plentiful. Cook, et al.[8] and Cook and Woods[9] provide examples of adverse drug infusion incidents involving these devices. Attempts have been made to create adverse event reporting systems to capture and analyze incidents and accidents. The FDA's MAUDE database entries show that problems with infusion devices are common, although submissions to the MAUDE database are often incomplete. Entries frequently lack deeper explanations of the circumstances in which adverse events occur and fail to identify specific causes. Instead of causes, events that involve infusion devices often attribute the outcome to "user error."[10] By contrast, our study observed instances of mode error and digit transposition on the apparatus.

Reports in the MAUDE medical device incident-tracking system indicate that practitioner difficulties with pump programming and operation are common, which indicates poor quality device design rather than operator failure. Our study's initial findings give us the opportunity to better understand actual adverse events that are associated with infusion devices as reported to the MAUDE database. We have previously described how these databases lack sufficient detail from which to draw reliable conclusions.[9] Even so, it is possible to produce plausible scenarios to describe how such events occur. In actual use, operators do not have the convenience of unlimited time to probe various paths to find the correct route as they did in the lab study reported here. It is possible that practitioners who are involved in adverse events that are reported in the MAUDE database have employed personal programming strategies, thought that they had reached the goal state, but were prevented from verifying it by intervening events.

Adverse event reports suggest that human–computer interactions are the "root cause" of the majority of events with these devices. Features of human–computer interaction mismatches seen in other industries, particularly mode errors,[9] are the source of many events.

Further work

Our study represents a new approach to looking at infusion pump safety in terms of usability and human-device interaction. The methods described provide a useful basis for study the evolution of adverse events. The results do not suggest an Achilles heel of infusion devices. There is no suggestion that a single design flaw produced the user difficulties that incident reporting and our own studies demonstrate. Instead, the problems with this and other commercially available devices arise from the complexity of work and the need to handle this complexity through a restricted interface. Improving the compatibility between infusion pumps and anesthesia work requirements is not a matter of fixing a particular aspect of a particular design. Instead, it is more a matter of developing a new approach to representing and aiding the interactions of workers with infusion devices.

The case of infusion devices clearly shows the challenges that IT designers face as they craft new automation that is intended to promote patient safety.

Acknowledgments

This research is supported by a grant (R18 HS11816) from the Agency for Healthcare Research and Quality.

Author Affiliations

All authors are at the Cognitive Technologies Laboratory, University of Chicago, Chicago, IL.

Address correspondence to: Christopher Nemeth, Ph.D., CHFP, Cognitive Technologies Laboratory, University of Chicago, 5841 S. Maryland MC4028; Chicago, IL 60637; phone: 773-702-4892, e-mail: cnemeth@uchicago.edu.

References

1. Christoffersen K, Woods D. How to make automated systems team players. advances in human performance and cognitive engineering research*.* Vol. 2. New York: Elsevier Science; 2002. pp. 1-12.
2. Hunt-Smith J, Donaghy A, Leslie K, et al. Safety and efficacy of target controlled infusion (diprifusor) vs. manually controlled infusion of propofol for anesthesia. Anaesth Intensive Care 1999;27:260–4.
3. Nemeth C. Report on infusion pump operation by healthcare professionals. Cognitive Technologies Laboratory. The University of Chicago; Chicago: 2003 July.
4. Nunnally M, Brunetti V, Woods D, Cook R. Infusion device characteristics related to user error during programming and operation determined by finite state modeling. Anesthesiology 2002;96:A520.
5. Bordens K, Abbott B. Research design and methods. Mountain View, CA: Mayfield Publishing Company; 1998.
6. Romera ME. Using finite automata to represent mental models. Unpublished Masters Thesis. San Jose, CA: San Jose State University; 2000.
7. Nunnally M, Brunetti V, O'Connor M, Render M et al. Lost in menuspace: variability among users programming infusion devices under controlled conditions. Anesthesiology 2002;96:A521.
8. Cook RI, Woods DD, Howie MB, et al. Case 2-1992: unintentional delivery of vasoactive drugs with an electromechanical infusion device. Journal of Cardiothoracic Vascular Anesthesia 1992;6:238–44.
9. Cook RI, Woods DD. Implications of automation surprises in aviation for the future of total intravenous anesthesia (TIVA). J Clin Anesth 1996;8:29S–37S.
10. Nunnally M, Brunetti V, Gosbee J, et al. Features of infusion device related incidents revealed by systematic analysis of an incident reporting database. Anesthesiology 202; 96:A1073.