The Human Factors of Algorithms and Sensor Design

Paul Picciano
Cognitive and Neural Science
University of Utah

I. Introduction

Human-operated systems often suffer from designers' limited consideration of user capacities and how they impact system performance. In many instances, technological capabilities and constraints drive design decisions. There is often a sense that if it *can* be done technologically, it *will* be done. However, this approach can prove to be an unwise appropriation of funds, time, and other resources. Justification should not simply rest on satiating a technology craving.

A greater return can be expected if sufficient resources are invested in optimizing the human-automation interaction. Algorithm design and sensor selection should be such that they support the needs of the human operator in accomplishing the task. Attention to the human element in complex system design is tantamount to technological considerations. To the detriment of many projects, human capabilities and limitations are often neglected or underemphasized (Sheridan, 2002).

We have adopted a human factors approach in the design of a just-in-time support (JITS) system to aid novice responders deliver cardiopulmonary resuscitation (CPR). Many design challenges were encountered in specifically addressing an untrained population. As a result, technological system development - from task drafting, to algorithm construction and sensor selection- required the integration of numerous human factors principles. Our system development process will serve as an illustration of these points later in the paper. Figure 1 provides a look at the system in use. A video screen and speakers provide step-by-step instructions.

Figure 1. JITS system for CPR.

II. Concerns with Technology Driven Design

Allowing technology to drive design almost by default diminishes the priority placed on the human interaction with the technology. It is vital to design for the interaction of task, user, system, and context. Fostering this interaction and optimizing "fit" is necessary to engender system-operator compatibility. Adding more technology to mitigate shortcomings usually only provides an ephemeral solution and can result in more drastic problems later (Reason, 1990).

Clearly, technical limitations in system design do pose formidable constraints. The inability to obtain certain data, or perform various tasks due to insufficient technology often requires alternative methods and procedures or a complete restructuring of the task.

However, possessing the means to perform a technical feat should not mandate its implementation. Again, the impact on the human operator should first be considered. Take for example extreme temporal updates of information. It may be possible to update a given data block at 5Hz; but assume empirical studies have shown the operator will only use that information at a rate of 0.2 Hz. This may represent not only a waste in developmental resources to achieve the 5 Hz rate, but could negatively impact performance by distracting and confusing the operator.

Conversely, it may be technically possible to obtain a piece of data but at such a slow rate, it is worthless to the operator. Should that sensor be added to the system and the data utilized by an algorithm? Waiting for information that doesn't arrive (in practical terms) could have deleterious effects on operator performance.

The crux of human factors design is eschewing the notion of working forward from what is *technically possible*, and embracing the idea of working backward from what the *user needs*.

III. Knowing the User and the Task

Before designing any system, a fundamental understanding of the goals and methods pertinent to the task should be garnered. When a human operator is part of a system, the human, as well as the human-automation interaction, require significant exploration in design decisions. Effective management of these issues requires insight pertaining to human cognition and their interaction with automation (Rasmussen, 1986).

User Expertise

Designers must determine the proficiency and knowledge of the user population in order to satisfy their information needs. Novices will likely have little success using a system designed for experts. Similarly, expert performance can be degraded when faced with a system intended for novices (either mismatch may lead to system abandonment). Therefore, it is vital to identify the user base and design for an apposite proficiency level.

Notable disparities exist between experts and novices (Chi & Glaser, 1988). These are important for both assessing the user population and making appropriate design choices. For example, experts tend to think more abstractly, perceive large, meaningful patterns, and organize tasks based on their domain expertise. In contrast, novices are unable to reason or organize abstractly in the domain, fail to recognize patterns, and rely on concrete and superficial representations.

Cues and feedback should be designed to prompt the user with actionable information. Aptly designed perceptual cues can engage and direct the novice to orchestrate the completion of each subtask. Sensors and algorithms can track user progress. These data are captured and processed by the system in an effort to optimize information delivery and ultimately improve task performance.

Addressing operator information needs provides a sound origin from which decisions about sensors and algorithms should be made. Collecting data that are neither important to the system nor the operator is not an efficient deployment of resources.

Task Analysis

A critical step in developing support tools is a thorough examination and description of the tasks, methods, and goals. Task analysis (TA) yields a deeper understanding of the cardinal elements of the task and exposes the structure and organization of the sub-tasks. A wide range of specific task analysis techniques exists including: cognitive task analysis, hierarchical task analysis, critical path analysis, timeline analysis, failure modes and effects analysis, and goals-means task analysis.

Method selection should be driven pursuant to the focus of the analysis. Areas of emphasis may include: actions performed, cognitive requirements, performance evaluation, temporal and sequencing issues, functional descriptions, or goal accomplishment. See Kirwan and Ainsworth (1992) for coverage of various techniques.

The motivation for TA is to reduce the global task into tractable modules. This provides a sensible template from which to construct the necessary algorithms. The subtasks and their interdependencies affecting the algorithms will be revealed allowing designers to accommodate the human and technological needs of the project.

Through task decomposition, requirements assessment, and error prediction, task analysis can lead to a robust, fault-tolerant system by elucidating critical performance issues thus elucidating information needs of the users.

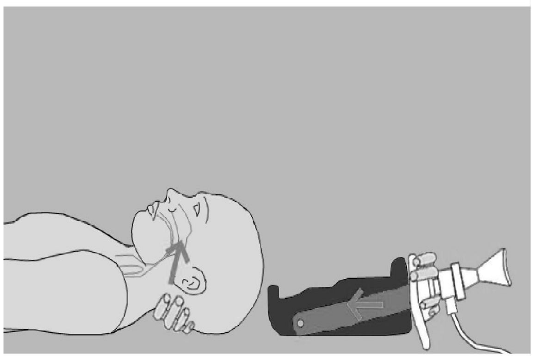
IV. Just-in-Time Support (JITS) for Novice Responders

In the system we designed in support of novice CPR responders, we discovered several issues in pilot testing that heavily impacted algorithm and sensor design. Algorithm construction and sensor selection evolved in an iterative process. This resulted in essentially concurrent development and afforded an efficient design process.

One of the first things we learned through the literature and pilot studies was that novice responders had an extremely difficult time delivering breaths. Many responders had shown an inability to maintain the airway while providing rescue breaths. Therefore the team set about to discover a means to provide additional support in this endeavor.

A headrest was developed that would provide a mechanical method for tilting the head relieving the operator of that dexterous task. Figure 2 shows a still from the animated instructions demonstrating headrest placement. Cues of shape, color, and affordances for head placement all contribute to the operator finding and correctly placing the headrest.

Figure 2. Animated instruction for placing headrest.



The synergy in algorithm and sensor development is exemplified here. The algorithm was altered (as was the task – removing manual headtilt) with the addition of the headrest, requiring integration of sensors in order to provide the proper feedback support the goal-driven algorithm. Pressure sensors partly recessed in the headrest detect head placement. Upon surpassing threshold (indicating the head was properly placed), the algorithm proceeds to the next step which involves placing the mask over the nose and mouth.

The system's suitability for a wider range of users can also be highlighted here. While a novice may require additional time identifying the headrest and placing it correctly, a more sophisticated user would not be inappropriately delayed. The trigger of the pressure sensors drives the algorithm when the head is properly placed and advances the operator to the next step resulting in a self-paced task.

Simply collecting data to drive the algorithm ballistically was not sufficient. This project also relied heavily on feedback to the operator to improve performance. Thus it was vital to identify user information on multiple levels of the task. This in turn would drive the selection of sensors and algorithms as well.

Since we were dealing with a novice population, advanced concepts of physiology and emergency procedures would not be helpful. We didn't need to provide feedback concerning oxygen perfusion or intrathoracic pressure (this also illustrates a point where we could scale back. We found we could eliminate a sensor that we initially thought would be a part of the system – a pulseoximeter. Even though we had the technology to obtain this information, we determined novice responders would have no use for such information). We instead provided feedback in more concrete terms such as "push harder" and "give 2 large breaths" accompanying animated instructions.

V. Conclusions and Future Research

The development of our system has benefited from the input of a diverse group of designers representing Anesthesiology, multiple engineering disciplines, Nursing, and cognitive psychology. Fortunately all parties have embraced the importance of the user in system development. The system is currently proving its merit in controlled experimental tests with participants "saving" a CPR manikin.

This effort has resulted in a robust system that is able to provide instruction to naïve subjects and enable them to administer efficacious treatment. Comparable participants not receiving JITS are demonstrating an inability to provide any life-saving measures.

Future experiments will explore how trained experts (EMTs) interact with such a system. We plan to have two "expertise levels" available. One will be a low expertise condition, where the system operates as it does now (designed for novices). The second condition will be tailored for the expert user. The information provided as well as the presentation will be more suited to their level of expertise We anticipate better performance from EMTs when using the expert system as that should be the best fit for their knowledge and experience.

Acknowledgments
This work has been supported through a grant sponsored by NASA - Johnson Space Center and has been conducted under the leadership of Dr. Frank Drews and Dr. Dwayne Westenskow.

VI. References

Chi, M.T.M., & Glaser, R., & Farr, M.J (Eds.). (1988). The Nature of Expertise. Hillsdale, NJ: Erlbaum.

Kirwan, B., and Ainsworth, L.K., (1992). A Guide To Task Analysis. London: Taylor & Francis,

Rasmussen, J. (1986). Information Processing and Human-Machine interaction: An Approach to Cognitive Engineering. New York: North-Holland.

Reason, J.(1990). Human Error. London: Cambridge University Press.

Sheridan, T. (2002). Humans and Automation: System Design and Research Issues. Santa Monica, CA: Wiley & Sons, Inc. in cooperation with the Human Factors & Ergonomics Society